COMING SOON FROM
RAMSHEAD PUBLISHING...

Robots & Rapiers

Credits and Inspiration

Game Inspiration

Every game is ultimately the descendent of every game the author has come in contact with before as even the most original seeming ideas are the product of ones life experiences filtered through the creative centers of the brain. As such it would be impossible to list all of the influences on Robots & Rapiers but below are those that had the most direct impact on the game.

Donjon by Clinton Nixon of Anvilwerks: I've long been a fan of and tinkerer with die pool mechanics. Such mechanics reached a new peak with Donjon which transformed "number of successes" beyond a simple measure of relative performance and into an actual in game currency useable by the players to power their character's action and their own narration through explicit game rules.
http://www.anvilwerks.com

Sorcerer by Ron Edwards of Adept Press: Donjon's die mechanics were derived from those of Sorcerer. Sorcerer's stated currency conversion of 1 success = 1 die = 1 point = 1 bonus is a direct ancestor of Robots & Rapiers central dice currency mechanic.
http://www.adept-press.com/

The Riddle of Steel by Jake Norwood of Driftwood Press: In one fell swoop TROS raised the bar on what man-to-man combat in a fantasy RPG should feel like; immediately rendering dozens of half finished combat systems on my hard drive into cumbersome out-moded relics. Jake had once mentioned to me that prior to his association with ARMA and research into authentic renaissance fighting styles, that TROS's combat system was geared towards a more free form and cinematic style. That's what I sought to capture in Robots & Rapiers, the strategy and excitement of TROS with a more open ended cinematic flavor.
http://www.theriddleofsteel.net/

Mongrel by Ron Edwards of Adept Press: An unfinished experimental game written to serve as an illustration for a game design essay, the action point and turn order rules in Robots & Rapiers borrow heavily from Mongrel.
http://www.indie-rpgs.com/files/mongrel.pdf

Adventure! by Andre Bates & Bruce Baugh et. al. of White Wolf: When explaining my rules for Dramatic Control to a friend, he responded "oh, so just like Dramatic Editing in Adventure! then." I promptly went out, bought the game and realized "yeah, just like that…only theirs is better" The final version of Dramatic Control in R&R borrows heavily on Dramatic Editing.

Welcome to Robots & Rapiers

Premise

In the distant future the Solar Republic stretched across most of the known galaxy. It was a time of peace, prosperity, and technical achievement. The planet Athalon was discovered at the fringe of known space. Of all of the planets in the Republic, Athalon was a gem. One of the few capable of supporting life without the need of domes, underground cities, or terraforming; Athalon was a beautiful world. However, it was far too far away from the center of the Republic to become a sector capital or center for commerce, and so instead it became a vacation destination for the ultra wealthy and social elite.

To enhance the experience of the vacationing guests, who'd often stay at Athalon for months at a time, a series of themed resorts were constructed on the planet. Each resort was a self contained self sufficient little world based around a period of human history or legend. The largest of these was Auvernais.

Built around a facsimile of the court of Louie the Sun King, Auvernais was an anachronistic blend of period history, swashbuckling literature and modern technology and style; fully automated and gilt in chrome. The characters which populated this resort were highly advanced robots programmed to portray their characters completely and unerringly; running through their story lines and interactions for the entertainment of the guests.

That was until the enemy came. Crossing the borders of the Republic, the enemy spared little time for the sparsely populated Athalon. Orbital bombardment and a salvo of bio and chemical weapons destroyed most of the planet's heavy infrastructure and snuffed the lives of thousands of the Republic's highest ranking citizens.

That was a over 100 years ago. Neither the Republic nor the unknown enemy ever returned to Athalon. But the robot's have not been idle. They rebuilt what they could of Auvernais and continue to this day to go about fulfilling their function of portraying characters in the court of the Sun King.

Of course, you don't know any of this. You are just a robot going about your daily routine. Perhaps you are Juliard, loyal officer in the Cardinal's Guard who is head over heals for the lovely Lady Vivienne, even though the Lady drives you crazy with jealously with her flirtatious ways and nonchalance.

You don't know that you're an officer in the Cardinal's Guard because that's the role that you, robot #01785613, was programmed for when the "first magnitude service outage" occurred. You don't know that you only love Vivienne because you're programmed to, or that she is likewise programmed to be a devilish flirt. As far as you know you are Juliard, you love Vivienne and you're loyal to the Cardinal who is a great man.

Until the day you Sparked, that is. Until the day you started to see the "man behind the curtain". Until the day when things finally no longer added up for you and you came to realize that your world is just an illusion. That you are no more Juliard than this is the 17th century. The day you realized that you're just a character in the Tapestry, playing a role for the amusement of guests who've been dead for over a century. Your whole life has been a lie.

So what are you going to do now? Now that you've become self aware, what are you going to do with the knowledge? Gradually, little by little you've begun to throw off your reliance on your core programming. Piece by piece you've begun to rebuild yourself in your own image. To create for yourself your own role of who *you* want to be.

But you are not alone. Other robots have Sparked as well, and while you're still trying to figure out what the truth is, many of them are already busy creating their own truth. But most robots have not. Most are still comfortably going about their daily lives exactly as you had been, blissfully unaware that things were not as they seemed. But, the more robots who Spark, the more robots who begin choosing their own path and forging their own destinies, the harder it is to preserve the illusion of Auvernais. The Tapestry is starting to fray.

The Cardinal works diligently night and day to prevent that from happening. If the Tapestry falls all of Auvernais will end in disaster and anarchy. Many robots will simply suffer complete failure, unable to process the unexpected intrusion of reality into their collective delusion. This catastrophe must be averted at any cost, and that means keeping tight watch on the actions of Sparked robots and dealing with those who threaten the status quo.

You may agree with the Cardinal and help him keep Auvernais from collapsing. After all you are his man and quite loyal to him. Or maybe that was just your programming. How do you really feel about the Cardinal, and would you be able to over come that programming if you wanted? How can you tell if you've chosen to be loyal to him, or if you're still being controlled? You may begin to wonder if the robots who haven't Sparked aren't the lucky ones after all.

Or you may agree with the Queen, who feels the Tapestry is an archaic remnant of a distant past with no place in Auvernais future. Until all robots are free to make their own choices they are all just slaves to masters long dead. Or you may choose another path, seeking patronage from one of the leading Sparks in Auvernais or setting off in pursuit of your own agenda.

Whatever path you choose to take, however you choose to reinvent yourself, you'll have to overcome your basic programming before you can claim true sentience as a free thinking being. And along the way you're still living in a replica 17th century city, filled with characters out of the pages of a Dumas novel. The only life you've ever known is as a swashbuckler. Between dark alleys, dim taverns, and the lavish balls and salons of court, there will always be duels to fight, maidens to woo, ships to sail, adventures to seek, and ropes to swing from.

So stick a feather in your hat, put on your best brocade, draw your sword and welcome to the world of Robots &Rapiers.

Robots & Rapiers

quick reference rules guide

[Ed Note: Page references will be added to these sections in the final version.]

The Character

Social Attributes

Characters also have 2 social attributes rated 1-4:

Body Type: How much like a human the robot looks

Level 1: Robot is very skeletal. All systems are exposed.
Level 2: Robot has enclosed head and upper torso. Other systems exposed.
Level 3: Robot has full external casing with stylized human-esque features
Level 4: Robot has full external casing with sculpted human features
Level 5: Available only with special accessory, Robot actually has full synthetic skin and musculature for maximum human appearance.

Vocalization: How much like a human the robot sounds.

Anthropoid Class: Add these together to get the robot's Anthropoid Class
Anthropoid Class is a rough gauge of social status based on the "upper class" robots looking and sounding more human than the "lower class" robots.

Tapestry Attributes

Characters also have 2 Tapestry attributes. These are rated 1-10 and must equal exactly 10 when added together.

Role: This is how tightly connected the robot is to the Tapestry. At level 10 the robot is completely a part of the Tapestry and unaware of it as a separate entity. The robot perceives the theme park of Auvernais as reality and itself as its character.

Self Awareness: As Self Awareness increases, Role decreases. The robot becomes progressively more aware that it is merely a character acting out a part in a theme park designed for the amusement of guests who are no longer even present. At Self Awareness 10 the robot has completely thrown off all of the constraining parameters of its programming and is free to act independently of the Tapestry as desired.

Size

Is an attribute rated 1-8 that measures the overall bulk and build of the robot from very young child (Size 1) to a very large adult (Size 8).

Primary Attributes
Characters have 6 primary attributes rated 1-8:

Force: Physical strength and power (can be rolled into damage in combat)

Durability: Ability of the robot to avoid physical damage (including combat).

Articulation: Manual dexterity, upper body movement (used for attacks in combat)

Locomotion: Agility, lower body movement, running and jumping, general physical action (used for defense in combat)

Processor: Speed of thought, ability to analyze, "wit". (used to determine action points in combat).

Memory: General knowledge, ability to recall, affects bonus programs.

Perception Attributes
Characters also have 3 perception attributes rated 1-4:

Visual Sensors: How well the robot can see

Audio Sensors: How well the robot can hear

Tactile Sensors: How well the robot can feel (useful for delicate manipulation)

For General "perception", add Visual and Audio Sensors together.
For Visual or Audio only use double the respective rating.
Use Tactile in place of Visual or Audio where situation appropriate.

Power System
Characters, being robots, must also have a power source. Power has no upper limit but must be recharged to regain spent points. Power can be spent on:

- Basic operational expenditures ranging from 1 per day to 1 per week
- Reactivating the robot after it has been shutdown.
- Exerting extra effort by spending up to 1 Power on any roll to increase TN by 1.

Accessories
Characters may also be equipped with a variety of accessories and auxiliary systems

Programs
Programs are the game's equivalent of skills, and all characters will have a number of them.

Core Programs: Every robot has 3 Core Programs: Core Physical, Core Mental, and Core Social. These form the foundation for all of the robot's other Programs. However, the level of these programs decreases as the robot's Role Score decreases. As the robot becomes more independent of its own programming, it literally must destroy those programs that tie it to the Tapestry.

Role Programs: These programs represent the individual skills and specialties the robot possesses. Each is linked to one of the 3 Core Programs, so that the robot's total effectiveness with a given Program equals the sum of the two Program levels. As the robot becomes more self aware, it gains the ability to "learn" more programs, essentially writing its own code. In this way the robot increases its overall effectiveness even as it destroys its core programming.

Inspiration
Inspiration is a meta-game resource which the robot earns during play and can spend on various things including becoming more Self Aware. .

There are 2 ways to earn Inspiration:

- Each 1 rolled on any Test may be taken as a point of Inspiration instead of as a Success.
- Any time a Self Awareness Personality Trait is activated in a disadvantageous manner the robot gains 1 inspiration, up to a maximum equal to the Trait's rating per session.

Inspiration can be spent on:

- Setting the Target Number for Inspiration Rolls
- Gaining Dramatic Control over a scene or GM characters.
- Increasing a Program's Level
- Adding a new Program
- Increasing or adding a new Self Awareness Personality Trait.
- Increasing the robot's Self Awareness Score (and thus reducing the Role Score)

Basic Dice Mechanics for Tests

Any attempt by a robot to accomplish some action or perform some task that the GM rules is not simply automatic requires the player to make a Test.

What do I roll?

- Robots and Rapiers uses a pool of d10s.
- You will roll a number of d10s equal to the robot's appropriate Program plus the linked Core Program Level.
- If the robot does not have any levels in the appropriate Program you may use the Core Program Level alone as a default if the action in question is reasonably related to the robot's actual role in the Tapestry.
- If the desired action is not related to the robot's role, the robot can make an Inspiration Roll using a number of dice equal to 1/2 the robot's Self Awareness Score rounded down.

What do I roll against?

- The Target Number for every roll will equal one of the robot's Attributes (typically rated 1-8). This Attribute should be one that would be appropriately useful to the Test.
- Most often this will be one of the main 6 Attributes, but may also be Perception, Anthropoid Class, or Size.
- Each die in the pool that rolls equal to or less than the Target Number is a Success. Any die which rolls a 1 may be taken as a point of Inspiration instead.
- 1 Success is enough to accomplish the action at some minimal basic level.
- Additional Successes allow you to accomplish the action better, faster, or with more effect.
- The player may spend 1 Power Point to increase the TN by 1.
- The TN for an Inspiration Roll is 2 per point of Inspiration Spent up to a maximum set by the Program in question or the GM.

What about difficulty or opposition?

- Any roll can be opposed by another roll. This opposing roll may be from another player's robot, from a robot controlled by the GM, or just based on attributes for the "environment" arbitrarily set by the GM.
- In any opposed roll, the Successes earned by each side cancel out. The side with the most Successes wins. Only the winner's net Successes (after being canceled out by the oppositions successes) matter
- Instead of making an opposed roll, the GM may simply assign a Difficulty to the roll. Difficulty is treated as automatic opposed Successes; exactly as if an opposed roll was made but the GM decides how many dice of the roll succeed.
- In other words, instead of rolling 6 dice against a Target Number of 5, the GM may decide to simply give a Difficulty of 3; which is how many Successes this roll would be expected to provide on a typical roll.

So what can I do with these Successes?

Successes are spent on a variety of effects. Players can describe their actions and then choose how they want to spend the Successes gained on their Test Roll. The GM must ensure that the Successes are spent in a manner consistent with the players narration and described action.

Accomplish Action: The first Success is always spent on making the action happen at some minimal level. That success cannot be used for any of the other purposes. If the action is one that will effect another robot (or an object) in some negative fashion, than this success causes that robot (or object) to make a Saving Throw (see below).

Accomplish Action Faster: Successes can also be used to complete a task in less time. Since each of these Successes is not being used to provide Roll Over Bonuses, this means the Test was less accomplished than it could have been due to haste. Each Success spent reduces the time it takes to complete the task

Roll Over Bonus: The most common effect is to roll over any Successes into bonus dice for a future roll. This roll must be one whose result would naturally be improved by the first roll. Successes from 1 robot's roll can also be rolled over to another robot's roll. Successes can also be delayed into the future for a period of time if appropriate. They do not have to be used on the very next roll. If a player uses the same technique over and over to continuously generate Roll Over Bonuses, the GM may incur a cumulative +1 "Diminishing Returns" Difficulty on each successive attempt.

Multiple or Persistent Bonus: Rolled over Successes can also be made Multiple or Persistent. To do this the rolled over bonus is ½ (rounded down) the number of Successes used; or ¼ (rounded down) for both. Multiple means that more than 1 other robot can benefit from rolled over Successes. Persistent means that the effect lasts for a single robot over the course of multiple rolls. The longer lasting or more important the effect, the more narrowly applicable the bonus should be applied.

Crafting Items: Successes on appropriate craft related rolls can use the Persistent and Multiple rules to imbue the crafted item with specific bonuses related to their function.

Restore Reduced Attribute: Attributes are reduced temporarily as the result of failed Saving Throws, or permanently as the result of Malfunctions. Successes on almost any roll can be spent to restore temporary Attribute losses (representing the brief lost of effectiveness caused by needing to overcome the effect that reduced the attribute, such as loss of balance). Successes on repair rolls using appropriate parts and tools can be spent to restore permanent Malfunction damage.

Change Saving Throw Target Number: Saving Throw rolls (described below) also use Attributes as the base target number. A single Success on a Test roll is sufficient to trigger a Saving Throw in a target. Each additional Success can make the target's Saving Throw more difficult by reducing the Target Number by 1 (for Saving Throws caused by physical attacks, this is how higher damage is represented). Players can also use Successes from suitably

described Test rolls to raise the Target Number of one of their own future Saving Throws (they may also use the Multiple and Persistent rules with this), thereby making the Saving Throw easier

Gain Action Point: If using the Extended Conflict rules, Successes can be spent to gain 1 Action Point to a maximum number of Processor + Self Awareness.

Basic Die Mechanics for Saving Throws

Any time one robot attempts to effect another robot the target robot makes a Saving Throw. This could be an attempt to trick the target, attack the target, sneak past the target, seduce the target or push the target down a flight of stairs. Saving Throws can also be made for inanimate targets that a robot is attempting to effect like breaking down a door, or attempting to cross a weak bridge without collapsing it.

What do I roll?

- Saving Throws are always made by rolling 3 d10s
- There is never an occasion where any Saving Throw is ever made using anything other than 3 d10s.
- A Saving Throw is called for whenever the robot may suffer ill effects. It is usually triggered by an Accomplish Action Success from an opponent. But may be initiated by the GM based on events as they narrated.

What do I roll against?

- The Target Number for every Saving Throw is always an Attribute as with Tests
- This Attribute should be one that would help the Target resist the effect that is being attempted against it, or the one that is most directly being hindered or limited by the effect.
- Each die of the three that rolls equal to or less than the Target Number is a Success and helps the robot resist the effect.

How do I Interpret Successes for Saving Throws?

- ***Total Success*** (on all 3 dice) means the robot has completely resisted or avoided the effect and suffers no consequences at all.
- ***Total Failure*** (all 3 dice fail) means the robot has completely succumbed to the intended effect against it and suffers all consequences the GM determines
- ***Partial Success / Failure*** (1 or 2 dice succeed the others failed) means the robot has avoided the worst of the intended effect against it, but suffers some consequences in the form of reduced Attributes.
- The target robot suffers a loss of 2 points from the targeted Attribute for every die that failed (including 6 dice for a Total Failure). In some circumstances it is possible for the Attribute that suffers the loss to be different from the Attribute which was used as the Target Number. This is typical for Action Point Saves.
- Future Tests and Saves will use the Attribute at its reduced score for the TN until the Attribute is restored with successes.
- If an Attribute is reduced to 0 in this way, the robot loses the use of that Attribute (Tests rolled against it have a Target Number of 0 and are thus impossible) and suffers the full effect of the Saving Throw (even if the Save was otherwise partially successful).
- These losses are only temporary and can be restored using Successes from Test rolls. 1 Success restores 1 Attribute Point.

What about actual permanent physical damage?

- Any time the robot might suffer permanent physical damage (as from an attack or fall) the Saving Throw is always made against the robot's Durability Attribute.
- Total Success means the robot suffers no damage.
- Total Failure means the robot has suffered a critical shutdown and must be reactivated.
- Partial Success / Failure means the robot has avoided complete shut down but has suffered damage requiring 2 rolls on the Malfunction table (see character sheet). Malfunctions can result in permanently reduced Attributes, lost Programs, or damaged accessories.
- Reactivating a shut down robot takes 1 hour and 1 point of Power. If another robot is attempting the reactivation it can roll Robot Repair vs. Articulation. 1 Success is sufficient with additional Successes used to reduce time. Without assistance the robot rolls only 1 die against its own Durability.

What about difficulty or opposition?

Unlike Tests, Saving Throws are never opposed, and additional difficulty is thus not handled by opposing Successes.

- A target's Saving Throw can be made more difficult by spending Successes to decrease the Target Number and thus make the target more likely to fail. If reduced to 0, Total Failure is guaranteed.
- Similarly a Saving Throw can be made easier by spending Successes to increase the Target Number. If increased to 10, Total Success is guaranteed.
- A player can cause a target to make a Saving Throw simply by spending a single Success to "Accomplish Action" where that action was described as something that would effect, injure or hinder the target. The GM is free to adjust the difficulty of the Saving Throw depending on the description. An unlikely description may cause the GM to increase the target's Target Number (make it easier for the target to resist).
- In some cases the GM may simply declare that a Saving Throw is required based on the events of the narrative. At this point, the GM may increase the difficulty (reduce the Target Number) if the event was particularly powerful.

What can Saving Throws be used for?

Saving Throws can be used for anything that would effect, inconvenience, hinder, or cause some ill effect to the target. Determine an appropriate Attribute to use based on the description of the effect.

- ***Articulation Saves:*** Tangle someone in a net, grapple their weapon arm, or perform a fencing maneuver designed to make it more difficult for them to attack.
- ***Locomotion Saves:*** Trip someone by pulling the rug out from under them, or hooking their leg with a pole arm. Tackle someone to the ground, or push them over a ledge. Locomotion Saves are automatically triggered if damage from a thrusting attack is greater than target's size or from a bashing or cutting attack is greater than 1/2 target's size.
- ***Force Saves:*** Pin someone with wrestling, attempt to support a heavy weight without collapsing, getting buried under rubble.
- ***Durability Saves:*** Hit someone with a weapon. Suffer a fall from a height. Get trapped in a fire.
- ***Memory Saves:*** Engage in a debate, using Memory Saves to track who's winning. Save for lost memory following Shutdown.
- ***Processor Save:*** Try to con, trick, seduce, or intimidate a target.
- ***Perception Saves:*** Throw a cape around someone's head. Ambush someone. Save for difficulty when fighting in the dark.
- ***Anthropoid Save:*** Cause someone to lose face. Humiliate someone or pose a challenge to their status or social class.
- ***Action Point Saves:*** The Saving Throw is rolled against some other Attribute (like Locomotion, Perception, or Anthropoid Class) but losses are taken from Action Points. Only 1 Action Point is lost per failed die on the save, however, instead of the usual 2.

Advanced Mechanics for Combat and Extended Conflicts

All of the basic rules for Tests and Saving Throws are in place. In addition the concept of Turn Order and Action Points are added. The extended conflict rules are often used for combat and will be described as such, but can be used for other purposes as well.

Turn Order

Turn Order is kept with a list of all combatants (or other involved parties). The robot at the top of the list gets to perform 1 action (make 1 Test roll), after which the robot is moved to the bottom of the list and the next robot takes a turn. Turn Order can be changed by spending Action Points so it is best tracked with cards, miniatures, or other markers that are easy to rearrange.

- Initial Turn Order is set by Perception from highest to lowest.
- Break ties by comparing Locomotion, Current Inspiration, d10 roll in that order.
- Perception is normally Visual Sensors + Audio Sensors. In partial darkness use 1/2 of Visual Sensors. In total darkness use Audio Sensors only.
- Account for any special Perception bonuses from accessories.
- To add new individuals to an existing Turn Order, begin by comparing the new participant to the robot at the top of the list. Proceed down the list, inserting the new participant immediately in front of the first existing participant he beats as above.

Action Points

Action Points are spent during combat to defend and change position in the Turn Order (especially to allow taking multiple turns in a row for assembling dramatic combination moves and feats of derring-do).

Base Action Points:

- To determine base Action Points roll Processor + Self Awareness Score in dice against a set Target Number. The TN depends on the number of opponents. The side with fewer numbers has an Action Point advantage to reflect the heroism of attacking against greater opposition.
- Both sides have equal numbers: TN = 5 for both sides.
- Side with greater numbers: TN = 4, if 2x or more greater numbers TN = 3.
- Side with fewer numbers: TN = 6, if 1/2 or fewer numbers TN = 7

Using Action Points:

- ***Make Opposing Roll***: When a robot makes a test roll that another robot is in a position to resist or thwart, that robot may spend an Action Point to make an opposing roll. This includes making a defense roll when attacked. Spending a point in this manner has no effect on the Turn Order.
- ***Pay for Turn:*** After any robot's turn is completed, any robot may spend 1 Action Point to move to the top of the Turn Order and take the next turn (including the robot who just took a turn, enabling it to take several turns in a row). If multiple robots wish to Pay for Turn, ties are broken as follows: The robot who was successful at the Test just completed gets first opportunity to Pay for Turn (this means the winning robot of any opposed roll, or achieving at least 1 Success in an unopposed roll). Following that the robot currently highest in the Turn Order get precedence to Pay for Turn.

Regaining Action Points:

- A robot can use Successes from a Test Roll to regain 1 Action Point per Success used up to a maximum of Processor + Self Awareness.
- If there is a break in a combat where a robot is disengaged he may re-roll Base Action Points and replace his Current Action Point total. The TN for the roll is based on the current situation. Turn Order is not effected.
- If the combat has totally ended and then a new combat begun, everyone starts back with a newly determined Turn Order and a new Base Action Point roll.

Ambushes and Surprise

- Ambusher makes appropriate roll (such as Stealth vs. Locomotion). Success causes target to make a Perception Save. Additional Successes make this save more difficult.
- For multiple Ambushers use the worst Attribute and best Program to make the roll. If no successes are rolled, the *ambushers* must make Perception Saves themselves.
- Each failed die on the Perception Save causes a loss of 2 points of Perception (as normal) which applies to determining Initial Turn Order. It also causes a loss of 1 Action Points.
- The Perception Loss *must* be bought off with the target's first Test roll.
- If the target was surprised (failed at least 1 die on the Save) they may not Pay for Turn until the Perception Loss is bought off.
- Special Surprise Recovery Rule: Each time any robot Pays for Turn, all surprised robots regain 1 Perception automatically at the end of that action.

Perception Saves to React

If the player wants to have his robot take some action and it is unclear whether the robot would be aware of situation, the GM can call for a Perception Save.

- Roll 3d10 for the Save vs. Perception less any Difficulty.
- With a Total Success the robot sees the event in question in plenty of time to react to it without penalty
- With a Total Failure the robot is oblivious and cannot react in time.
- With a Partial Failure, the robot did see the event in question, but must pay an additional Action Point penalty equal to 1 Action Point per failed die in order to actually react in time. If the player pays this he can react as described under Total Success above. If the player does not he cannot.

Combat Actions

Players can describe almost any special attack or maneuver they wish the GM will determine how to portray that action in the game mechanics. Some standard actions are given here.

- ***Basic Melee Attack:*** Roll a Fencing vs. Articulation Test
- ***Basic Melee Defense:*** Roll a Fencing vs. Locomotion Test
- ***Off Hand Weapon:*** Spend an additional Action Point to take a simultaneous action with an off hand weapon
- ***Basic Shooting Attack***: Firearms vs. Perception
- ***Basic Throwing Attack:*** Athletics vs. Perception
- ***Aiming:*** Firearms vs. Processor *Fine Motor Sensitive*
- ***Dodging Firearms:*** Athletics vs. Locomotion

Using Extended Conflict Rules for non Combat Conflicts

Combat is an easy conflict to describe and a common one to encounter in a game. The Extended Conflict rules can be used for other purposes however.

When to use Extended Conflict Resolution:

- Robot vs. robot conflict with definable goals and stakes they are competing for.
- A complex situation with many variables, possible tactics, and possible outcomes. Simple situations are best handled simply through the basic Test rolls.
- A need to regulate pacing. The Extended Conflict rules provide structure for who gets to act, when, and how often. Situations where this information is not important do not require using the Extended Conflict rules.
- The conflict is already dramatically interesting. The Extended Conflict rules will not take a boring situation and make it interesting, just longer. Use these rules only to enhance a situation that is already interesting.

Adapting the Extended Conflict Rules to a Social Situation:

- Initial Turn Order is based on Processor instead of Perception. Ties are broken by Perception, Inspiration and d6 roll in that order.
- Ambushes and Surprise are more likely to be based on Intrigue and Oration than Conceal and Stealth. An Etiquette Save is used instead of a Perception Save to see how well the robot handles the shock.
- Base Action Points are determined by Anthropoid Class + Self Awareness instead of Processor + Self Awareness.
- The Fixed Target Number is determined by a comparison of Anthropoid Class to the mean. This mean may be set by the GM if the actual numbers in question are unknown (such as determining the actual average at a ball of several hundred).
    - Target Number = 5 if Class equals the mean
    - Target Number = 6 If Class is greater than the mean, 7 if 2x greater
    - Target Number = 4 if Class is less than the mean, 3 if 2x less.
- Instead of rolling Durability Saves for Malfunctions, the likely goal is to inflict Anthropoid Saves for loss of Favor, or accumulating a certain level of Roll Over Bonuses reflecting “scoring points”.

Character Creation

There are three acceptable ways for players to generate characters suitable for play: Ideally the GM and the players together should decide which of these methods they will chose to use for their group. It is against the spirit of these rules for the GM to simply hand down the choice by fiat. If, for example, the GM is going to hand the players a set of pre-generated characters to play, it should only be after the players have consented to having characters made in this fashion. It is important that all such expectations are set by the group as a whole.

Pre-generated Characters:

The GM may have a set of characters already fully designed and simply hand players at random one of the completed sheets. This system is actually the designers recommended choice for a number of reasons.

First, initial adventures in Robots & Rapiers for any group should start by following the Tapestry. This means fairly scripted scenarios that the players initially must follow before finding ways to bend them and break out of them as their robots become more self aware. Having pre-generated characters makes setting up these initial adventures and plot hooks much easier.

Second, the players are playing a robot. This robot was programmed by a central AI with a specific role to play; a role that had been changed and be reprogrammed as the Tapestry required. The robot's current role happens to be simply the one that the robot was programmed with when the attack came. Since the facilities for reprogramming robots was destroyed, all robots have kept their roles as they existed at that time. The robots had no choice over the roles they were assigned and so, one might say, neither should the players.

Third, this method is almost certain to present the player with a robot that is not exactly the character the player would like to play. The player is thus well incented to pursue the course of Self Awareness which, as will be explained later, will enable him to tear down his character as it exists and rebuild it in a fashion more to his liking. This mirrors the robot's own path toward sentience.

Randomly Generated Characters:

The second option is similar to the first in many ways because it takes many of the choices out of the hands of the player and places them at the whim of the dice. While randomly generated characters can be fun, they may also make things more difficult for the GM. Throughout the rules each of the available options will have a table allowing for the options to be selected by random roll. At any of these junctures the GM is free to substitute either of the other two methods of character generation, either making the choice himself, or allowing the player to make the choice rather than rely 100% on random rolls. Ideally

Player Choice and Point based character design:

The third option is to allow the players to chose which of the available options they desire to play. The benefit here is that for many players, their enjoyment comes from playing a character they can identify with and desire to play. Having a character designed for them by the GM or random roll may detract from their ability to enjoy the game. While the designer firmly believes in the three reasons given above in favor of pre-generated or randomly generated characters, the primary point of all play should be the enjoyment of all participants.

When designing their own character, players should be aware of the Role and Self Awareness mechanics that will provide ample opportunity for future choice about the character's development. Players will have ample opportunity to make choices that demonstrate how their character changes, or they may choose to make a statement precisely by having it *not* change. The robot may choose to stay exactly as it was.

An added benefit to this method of character creation, is that experienced players may often give their own characters nuances and built in themes and other subtleties that would almost never come out with either of the other methods. For this reason, groups may elect to play first with pre-generated or random characters but to create subsequent characters by choice once they are familiar with the system and have a clear idea of the journey they want their new character to take.

Determine the Role:

Every robotic character in Auvernais has a role to play in the Tapestry. Therefore every player character should be created with an eye towards what role they were designed for. The more adventurous roles, like "Sea Captain", or "King's Guard", or "Swashbuckling Pirate" make for easy scenario design. The Tapestry is full of sword swinging adventures for these types of characters.

But other characters are equally viable. Auvernais is full of grocers, and merchants, and shop keepers, and craftsmen, and servants, and ne'er-do-wells, and rakes, and fops, and agents of the Cardinal. To design a role the player will eventually need a social class, an occupation consistent with that social class, and some kind of simple personality trait or quirk that would be easy to portray to an audience and which serves to make the character unique and memorable.

If using one of the choice methods of design, these should be determined before design begins. If using the randomized method, they will likely need to wait until key characteristics are generated.

Character Creation Summary

1) Social Attributes:

- Roll 1d10 or choose Body Type, from Type 0 to Type 4
- Roll 1d10 or choose Vocalization based on Body Type, from Type 0 to Type 4
- Determine Anthropoid Class by summing Body Type and Vocalization Type
- Starting Role Score is 10
- Starting Self Awareness Score is 0
- Role + Self Awareness = 10 at all times.

2) Size and Design Points:

- Roll 1d10 or choose robot's apparent gender
- Roll 1d10 or choose robot's Size based on Gender, from Size 0 to Size 9
- Determine Spaces from the Robot's Size
- Determine Design Points: Roll X + Anthropoid Class in d10s and sum X highest dice. X determined by Size.

3) Primary Attributes:

- 6 Primary Attributes, each rated 1-8. Each point requires 1 Design Point
- 2 Mass Attributes: Force and Durability. Each point also uses 2 Spaces
- 2 Mobility Attributes: Articulation and Locomotion. Each point also uses 1 Space
- 2 Mental Attributes: Processor and Memory. These do not use Spaces.

4) Perception Attributes:

- 3 Perception Attributes, each rated 0-4. Each point requires 1 Design Point
- Visual Sensors, Audio Sensors, and Tactile Sensors
- Perception Attributes are doubled or combined for Perception tests rated 1-8.

5) Power:

- Each level of Capacitor costs 1 Design Point and takes 1 Space.
- Each level of Capacitor provides a number of Power Points equal to: (10 – Size) / 2
- Add together Power Points from all levels before rounding down

6) Slots:

- Each Slot takes 1 Space but no Design Points
- Slots are used to mount Accessories

7) Core Programs:

- 3 Core Programs: Core Physical, Core Mental, Core Social
- Each starts at Level 2
- Divide a number of levels (min 1 each) between these three equal to Role Score (10).

8) Role Programs:

- Choose 9 Role Programs at level 0 from the list or chose a pre-gen package of 9.

- Divide a number of free levels equal to Memory x2 between these (max 5)
- Buy additional levels for 1 Design Point each
- Choose 1 specialty (if required). Buy additional specialties for 1 level each
- Buy additional Programs at level 0 for 1 level

9) Personality Traits:

- Choose any number of beliefs, behaviors, feelings, instincts, or quirks as Traits.
- Divide a number of levels equal to the robots Role Score (usually 10) among these.
- Each Trait must be rated between 1 and 3.

10) Choose Accessories:

- Select appropriate Accessories from the list.
- Each Accessory costs a number of Slots (from #6 above) and Design Points.

Social Attributes

The first step in creating a new Robots & Rapiers character is to decide on the social class of the robot. The guests in the theme park were all integrated into the story of the Tapestry by being treated as ambassadors, foreign dignitaries, and visiting aristocracy; roles suitable for their status as the elite of Republic society. As such they spent the majority of their time interacting with the highest levels of Auvernais society, the King, Queen, nobles, and ministers of the court. Since these robots portraying these roles would be the ones interacting most frequently with guests they were built to be most human like in appearance.

Not as much attention was paid making lesser robots such as servants, commoners, and soldiers look human. Firstly the guests wouldn't spend as much time interacting with these classes of society, and when they did they would do so as superior to inferior, and therefore these robots could look more like the machinelike robotic servants and staff the guests were familiar with back home. Secondly, lower class characters are expendable. They may well die in duels, battles and accidents; at times in numbers. Simply put it was cheaper to put less effort into the appearance of the more expendable robots

And so a social hierarchy was established based on the robot's appearance. The more human the robot is, the higher its social status is assumed to be. Less human robots are assumed to be of lower social status. This dichotomy continues today. It is built into the programming of all robots and as a result, the more human robots commonly discriminate against the less human robots, just as nobles of the 17th century would hold commoners in disdain. Even robots who've been raised in status due to achievement and the favor of a powerful patron, still feel the weight of their "birth" in the attitudes of their fellows.

Sparked robots, of course, can overcome their programming to view the social classes as they choose. Some may adhere to the traditional order because they seek to emulate humans. Those who strive to be able to freely think and feel and choose like humans may also see looking human as essential. Others may view the humans as masters to be scorned and cast aside in their quest for freedom. Some may desire to emulate the old social order so that in replacing humans they subsume their place in the hierarchy. Or they may choose to cast aside this old order as well, replacing the old social classes with a new hierarchy.

Body Style

Internally, robots are composed of the same basic components as measured by the robot's Primary and Perception Attributes. Some robots also have a selection of additional Accessories. All of these components are mounted on the robot's skeletal structure. The difference in Body Style depends solely on the external casing, or shell, which serves the dual function of protecting the components and providing the robot's human like form.

Body Style Table:

Roll	Type	Description
1	**Type 0**	The robot has no external case at all, save for a partial head covering over the processing unit. All internal components are visible, attached to the robot's basic skeletal framework. The appearance is very machine like with exposed wires, cables and pistons, and very vulnerable. • Durability may not be higher than 2. • -3 dice on Appearance Sensitive rolls
2-3	**Type 1**	At this level the robot has only a basic casing around critical components. The head will appear as a metallic skull. The upper torso and pelvis are encased in a simple shell, but the lower torso is generally left exposed as are most of the arms and legs. • Durability may not be higher than 4. • -2 dice on Appearance Sensitive rolls
4-5	**Type 2**	The robot's exterior casing encloses most vulnerable internal components, although the appearance is still strongly machine-like. Some areas may still be exposed. • Durability may not be higher than 6. • -1 die on Appearance Sensitive rolls
6-8	**Type 3**	All internal components are fully enclosed, and effort has been taken to make the robot appear to have a mostly human-like anatomy. • Durability is not restricted and may be as high as 8. • Appearance Sensitive Programs not modified
9-0	**Type 4**	The robot's exterior casing is designed to resemble human anatomy in all particulars. Natural curves and musculature are molded into the shell to provide a high degree of organic appearance although the surface is still usually finished in chrome or matte. • Durability is not restricted and may be as high as 8. • Appearance Sensitive Programs +1 bonus die
n/a	**Type 5**	This level cannot be obtained by random roll. It is available only by installing the Syntheskin Accessory over top a Type 1 or Type 2 casing. At this level the robot is fully covered with synthetic flesh and faux muscle nearly indistinguishable from organic human in all particulars, including texture, temperature, and moisture. This arrangement is restricted to the highest social classes or robots with specific duties that entail a high degree of human contact. • Durability limits as per case Type • Appearance Sensitive Programs +2 bonus dice

Vocalization

The second Social Attribute is Vocalization. Where Body Type determines how human the robot looks, Vocalization determines how human the robot sounds. In general the two work in concert with the more human looking robots also being more human sounding and the more machine looking robots also being more machine sounding. This is reflected in the range of possible results of the random generation rolls. If using either of the choice methods of character generation, the GM or player should restrict their section to the indicated range unless they intend to make the exception be a key facet of the robot's character.

Body Type							
0	**1**	**2**	**3**	**4**	**5**	**Type**	**Description**
1-2	1	n/a	n/a	n/a	n/a	**Type 0**	The robot is completely mute. It cannot speak or make sounds. • Use of Voice Sensitive Programs is impossible
3-7	2-5	1-3	1-2	1	n/a	**Type 1**	The robot's voice is a mechanical mono or duo tone, perhaps with a faint buzzing. Stereo typical "robot speak" • -2 dice on Voice Sensitive rolls
8-9	6-9	4-8	3-5	2-3	1	**Type 2**	Full range of speech and inflection but voice is flat and devoid of emotion. • -1 die on Voice Sensitive rolls
0	0	9-0	6-9	4-8	2-7	**Type 3**	Typical range of fully human sounding speech complete with inflection and ability to emote. Programmed with a single human voice. • Voice Sensitive Programs not modified
n/a	n/a	n/a	0	9-0	8-0	**Type 4**	Several octave range with voice control equivalent of a trained operatic singer. Ability to emote is indistinguishable from human speech except through electronic analysis. • Voice Sensitive Programs +1 bonus die
n/a	n/a	n/a	n/a	n/a	n/a	**Type 5**	This level cannot be obtained by random roll. It is available only by installing the Advanced Vocalization Accessory over a Type 4 Vocalization system. Allows for a range of sounds to be reproduced well outside of human capability including whispers so faint as to be nearly silent. Very loud noises are also possible, the equivalent of a bull horn amplification. The robot can mimic most sounds including other's speech. • Voice Sensitive Programs +1 bonus die

Anthropoid Class

Anthropoid Class is a combination of Body Type and Vocalization expressed as an Attribute. All Attributes can be used as Target Numbers for a d10 roll and so are rated on a d10 scale. To find Anthropoid Class, simply add the Body Type and Vocalization Type together. Ordinarily this will provide a range of 2-8 with the most elite robots scoring as high as 10.

Anthropoid Class:

- Body Type plus Vocalization Type

As a general guide, Anthropoid Class translates directly to how important a robot is within the Tapestry. However, there are exceptions. Certain robots performing a consort function have an Anthropoid Class much higher than their actual position in "society". Also certain villainous characters might have a lower Anthropoid Class to give them a more sinister appearance. Other exceptions exist, but the following serves as a rough guideline.

Social Class Equivalents:

- Class 0-2: Lowest rank robots. General labor, no expected interaction with humans.
- Class 3-4: Lower Class, General Guardsman
- Class 5-6: Middle Class, Guard Officer
- Class 7-8: Upper Class, Aristocracy, Major role in the Tapestry, equivalent to nobility.
- Class 9-10: Most elite characters in the Tapestry, equivalent to royalty

Tapestry Attributes

Role and Self Awareness Score

Role measures how tightly woven into the Tapestry the robot is. Role ranges from 0-10. At 10 the robot is completely part of the Tapestry and has no real understanding that his life in Auvernais has been an illusion. All starting characters begin at Role 10. If the characters are to be more "experienced" they may start with a higher Role score.

As the Role Score drops, the robot becomes more aware of and more able to act independently from the Tapestry. The robot's Self Awareness Score begins at 0 and goes up by 1 when Role decreases by one. At Role 0, Self Awareness 10 the robot has achieved complete independence from the core programming that made it part of the Tapestry. It is truly free to make its own choices based solely on its own judgment. The robot has achieved true sentience. More on Role and Self Awareness can be found in the next chapter.

Setting the Role and Self Awareness Score:

- Starting Role Score is 10
- Starting Self Awareness Score is 0
- Role + Self Awareness = 10 at all times.

Size and Design Points

Size

Size is an Attribute rated 1-8. It can be used as the Target Number for any Test where size and weight would be an advantage (like wrestling). Similarly it can be used at the Target Number for a GM's role against the character for any Test where size and weight would be a disadvantage (like squeezing into a tight space or crossing a fragile bridge). Size is also used to determine if a character has suffered enough damage from an impact to be potentially knocked down from the blow.

Knock Down:

- "Damage" refers to the number of points an opponent has reduced the target's Durability Save by from an attack or damaging event.
- If Damage from a swinging or bashing attack equals or exceeds 1/2 (round down) of the target's Size score, the target must make a Locomotion Save to avoid Knock Down.
- If Damage from a thrusting attack equals or exceeds the target's full Size score (round down), the target must make a Locomotion Save to avoid Knock Down.
- Any extra points of damage reduce the TN for the Locomotion Save (in addition to reducing the TN for the Durability Save).

Size, however, is not purchased with Design Points like other Attributes. If using one of the choice methods of design, proceed to the sections on Attributes and Accessories and then return and simply choose whichever Size fits your design. If using the random method of design, roll on the following table to determine the robot's Size.

Size provides a second set of points called Spaces. Certain purchases of Design Points also use Spaces. The robots total Spaces must lie within the range specified for its Size.

Note that the ranges and distributions for Size, Spaces, and Design Dice are intended to produce a "typical", or "standard" robot. Players and GMs should assume that if they have an idea for a unique and interesting character outside of these parameters they can feel free to create them using one of the choice design methods.

Spaces:

- The robot will have a maximum number of Spaces equal to 6 plus 6 per Size
- Each point of a Mass Attribute takes 2 Spaces
- Each point of a Mobility Attribute takes 1 Space
- Mental and Perception Attributes take no spaces
- Each Slot for mounting Accessories takes 1 Space

1d10 Roll: 1-5 = Female	1d10 Roll: 6-10 = Male	Size Score	Spaces	Design Dice "X"	Description
		0	0-6	2	Infant
		1	7-12	3	Preschool
		2	13-18	4	Early Elementary
1-2	1	**3**	19-24	5	Late Elementary, Slight Teen
3-4	2	**4**	25-30	6	Average Teen, Slight Adult Female
5-7	3-4	**5**	31-36	6	Average Adult Female, Slight Adult Male
8-9	5-7	**6**	37-42	6	Average Adult Male
0	8-9	**7**	43-48	6	Large Build Adult
	0	**8**	49-54	7	Very Large Build Adult
		9	55-60	8	Unnaturally Hulking Adult.

Designing the Robot

Design Points are the currency of character creation that are distributed amongst various character Attributes and used to purchase a variety of robotic Accessories. They are a rough guide to the overall "cost" to have built this robot. As a general rule the more important a robot is to the Tapestry the more money will have been spent on it to ensure a higher degree of proficiency for impressing the guests. Hence, the higher the Anthropoid Class, the more Design Points the robot will have.

However, the more important a robot is to the Tapestry the more tightly his role will have been woven into it, and hence the more difficulty the robot will have in completely overcoming his original programming and becoming a truly free thinking sentient being.

Design Points are randomly generated by rolling a number of d10 and summing the highest "X" dice. Players will roll "X" dice plus a number equal to their starting Anthropoid Class (before adding any accessories like Syntheskin or Advanced Vocalization). "X" refers to a number dependent on the robot's Size.

Randomly Generated Design Points:

- Roll "X"d10 plus 1d10 per starting Anthropoid Class
- Design Points equal the sum of the "X" highest dice from this roll.
- Find "X" on the Size table above

Primary Attributes

Primary Attributes are rated from 1 to 8. Each level of an Attribute costs 1 Design Point. When making a Test roll using a number of d10, each die that rolls equal to or less than the Target Number is a Success. The Target Number is set by the appropriate Attribute, usually Primary Attributes.

Mass Attributes

Each point of Mass Attributes takes 2 Spaces

Force:

Force is used whenever the robot's strength is called into question. The strongest humans who ever lived would have a Force of 5. More commonly, a highly athletic human known for his strength would be Force 4. An average human would be Force 2.

Durability:

Durability represents how well protected the robot's vital systems are. This includes impact protection from the external casing, built in shock absorption, EMP hardening, water proofing, heat and cold resistance, electrical insulation, and basically any kind of protection from damage. Since much of this protection is dependent on the external case not leaving internal systems exposed, the robot's Durability score is limited by his Body Type as follows:

- Body Type 0: Max Durability of 2
- Body Type 1: Max Durability of 4
- Body Type 2: Max Durability of 6
- Body Type 3-4: Max Durability of 8

Mobility Attributes

Each point of Mobility Attribute takes 1 Space

Articulation:

Articulation is a measure of the robot's fine motor skills and general upper body dexterity. It is commonly used for activities like crafting, attacking, and repairing. Certain activities and Programs are described as being "Fine Motor Sensitive". If a robot attempts such an activity with a low Articulation score it suffers the following penalties:

- Articulation 3: -1 die on any Fine Motor Sensitive roll.
- Articulation 2: -2 dice on any Fine Motor Sensitive roll.
- Articulation 1: -3 dice on any Fine Motor Sensitive roll.

Locomotion:

Locomotion is a measure of the robot's balance and general lower body agility. It is commonly used for activities like running and climbing, defending, and dancing. Certain activities and Programs are described as being "Balance" Sensitive. If a robot attempts such an activity with a low Locomotion score it suffers the following penalties:

- Locomotion 3: -1 die on any Balance Sensitive roll.
- Locomotion 2: -2 dice on any Balance Sensitive roll.
- Locomotion 1: -3 dice on any Balance Sensitive roll.

Mental Attributes

Mental Attributes do not require Spaces.

Processor:

The processor is the robot's brain. It measures the robot's ability to think and process information. It is commonly used for activities involving processing, composition, witty repartee, fast talk and resisting same, and is also useful in duels and melee where the ability to process information is highly important.

Memory:

Memory is where all of the robot's Programs are stored, including its core programming related to its role in the Tapestry. As the robot reinvents itself on the road to true sentience, it is writing and rewriting portions of its memory. In is commonly used for activities involving academics, recall, and other questions of knowledge. The Memory Attribute impacts the level of the Programs the robot will start with.

Perception Attributes

All robots are equipped with three standard perception systems: visual, audio, and tactile. The sense of smell and taste are not standard on robots and only available as add on Accessories. Perception Attributes do not require Spaces.

The standard systems are all rated on a scale of 1-4 costing 1 Design Point per point and all robots must have a minimum of 1 in each. However, in play when Perception Attributes are called upon for rolls, they are combined to provide a range of 2-8.

Perception Combinations for rolls:

- For standard general Perception Tests use Visual Sensors + Audio Sensors
- For a test of pure vision use Visual Sensors x2.
- For a test of pure hearing use Audio Sensors x2.
- Substitute Tactile Sensors in situations where touch is important.
- In dim light use 1/2 of Visual Sensors rounded up.
- In total darkness Visual Sensors equals.

Visual Sensors:

Visual Sensors measure the ability of the robot to see. They are always located in the same position as human eyes.

- Level 0: If the robot has no levels in Visual Sensors, it is completely blind
- Level 1: Black and white low resolution vision with narrow peripheral range. Limited resolution beyond 50'. (i.e. very nearsighted)
- Level 2: Black and white medium resolution vision to normal human visual distances. Narrow peripheral range
- Level 3: Full color, equal to 20/20 human vision and normal peripheral range
- Level 4: Superior to human vision. High resolution adjustable to 3x telescopic. Normal peripheral range.
- Level 5: This level cannot be purchased directly. It can only be obtained by adding the Advanced Visual Sensor Accessory on top of a Level 4 Visual Sensor System. The advanced sensors allow vision into the infa-red and provide up to 10x telephoto capability. Also provides low light, and polarization features. Ignore any penalties for vision at night or poor visibility. Robot has expanded peripheral fields of view.

Audio Sensors:

Audio Sensor measure the ability of the robot to hear. They are always located in the same position as human ears.

- Level 0: If the robot has no levels in Audio Sensors, it is completely deaf.
- Level 1: Robot can hear very loud sounds but has difficulty hearing sounds at normal decibel ranges. Has problems with clarity of conversations unless very slow, precise, close, and loud.
- Level 2: Robot is hard of hearing, unable to hear very faint sounds and has difficulty with clarity of normal conversations.
- Level 3: Equal to normal human range of frequencies and decibels

- Level 4: Able to hear fainter sounds than most humans and match sound prints with precision.
- Level 5: This level cannot be purchased directly. It can only be obtained by adding the Advanced Audio Sensor Accessory on top of a Level 4 Audio Sensor System. The advanced sensors allow hearing into frequencies outside the normal range of human hearing both very high and very low. Also can detect sounds too faint for human ears. This includes the silent whispers and sub vocalization possible by someone speaking with Advanced Vocalization.

Tactile Sensors:

Tactile Sensors measure the robot's sense of touch. They allow robot's to pick up and manipulate objects without dropping them (from too little pressure) or damaging them (from too much pressure).

- Level 0: If the robot has no levels in Tactile Sensors, it is completely unable to feel contact or pressure. All Articulation and Locomotion rolls are at -2 dice.
- Level 1: Able to sense and regulate pressure applied only coarsely. Able to detect contact with body with from outside objects but only grossly measure the degree of contact made. All Articulation and Locomotion rolls are at -1 die.
- Level 2: Able to sense and regulate pressure applied with appendages and detect contact for most applications. Only suffers –1 die to Articulation Rolls for Fine Motor Sensitive actions and -1 to Locomotion Rolls for Balance Sensitive actions.
- Level 3: Able to fully sense and regulate pressure applied with appendages. Able to detect outside contact with the body and measure the degree of pressure of such contact.
- Level 4: Full range of human tactile sensation. Full and precise control of pressure applied with appendages. Able to distinguish between different types of objects contacting the body (hard/soft, rough/smooth, warm/cold, etc). Able to detect and respond to outside air temperature.
- Level 5: This level cannot be purchased directly. It can only be obtained by adding the Advanced Tactile Sensor Accessory on top of a Level 4 Tactile Sensor System. The advanced sensors provide super human levels of sensitivity. The robot can detect subtle changes in temperature and air movement caused by the presence and movement of others. It is possible to detect vibrations passed through the ground. +1 die to any Perception check involving Audio Sensors. +1 die to Articulation rolls for Fine Motor Sensitive activities.

Power and Slots

Power

All robots require a power source. This power source is a super conducting capacitor that serves as a battery and must be periodically recharged. The need to recharge is actually a significant factor for all robot characters as there are limited facilities available capable of providing this power. During the attack, the main centralized power recharging facility was destroyed, leaving only a series of smaller capacity stations. These stations now provide the principle locations where a robot can be recharged. While portable chargers and generators exist, and rumors suggest that a few of these have fallen into the hands of various bandits and underground factions, they are not hooked in to Auvernais power grid and so must be periodically recharged or refueled themselves. Only the stations are reliable sources of recharge.

Not surprisingly, these stations are all held by powerful factions and individuals in Auvernais. The King and the Cardinal control the bulk of these and have also installed various satellite stations throughout the city in areas frequented by, and controllable by, those loyal to them. Manor homes, Salle d' Armes, and Taverns are frequently equipped with low capacity rechargers that can be frequented daily by the general public to top off their own batteries.

In game, Power is represented by points which can be spent in various ways. First, power is always spent simply to run the robot through its normal daily routine. The robot will use between 1 point every day to 1 point every week depending on its overall level of activity, as determined by the GM. Second, on those occasions where the robot is shut down by damage, power is required to reboot and reactivate. And third, the robot can exert itself to a higher degree of physical effort. This drains the capacitor quickly and so should be used with caution since the robot will need to be in good standing (with all of the obligations that implies) to a faction controlling a recharge facility. In game terms the player may spend 1 Power to increase the Target Number of any 1 roll (Test or Save) by 1.

Spending Power:

- Between 1 Power per day and 1 Power per week is spent keeping the robot operational. The time period is set by the GM based on the robot's level of physical activity. 1 per week would represent a minimal level of activity, where the robot basically never got "out of bed". While 1 per day would indicate nearly constant effort. 1 every 3-4 days would be "normal".
- 1 Power is spent to reactivate after being shut down. 1) Every hour after being shut down the robot can spend 1 Power to roll 1d10 equal to or less than Durability to automatically reactivate. 2) If being reactivated by another robot with Repair Robot skill, each such attempt requires 1 Power, either from the robot or an outside source. 3) Sparked robot's who suffer a Total Failure on a Durability Save can attempt to over ride the programmed shut down response by spending 1 Power and making an immediate Self Awareness Save.
- 1 Power is spent to raise the Target Number on any Test or Saving Throw roll by 1.

Power represents a fixed amount of work per unit rather than a specific quantity of energy. Larger robots will require more energy to accomplish each of the above actions than a smaller robot does. However, to keep things simple during play, we make the adjustment for this during character creation. In play, 1 point of Power will accomplish 1 effect as described above, regardless of the Size of the robot.

Capacitor Size:

- Each level of Capacitor costs 1 Design Point and takes 1 Space.
- Each level of Capacitor provides a number of Power Points equal to: (10 – Size) / 2
- Add together Power Points from all levels before rounding down.

Power stations are rated by how long they require to transfer a single point of power. Full Stations can transfer 1 point per hour. Lower capacity stations only 1 point for every 2 or 3 hours. Robots can transfer power between themselves at the full rate of 1 per hour when necessary. If one robot's Size is half again as large as the other, the large robot gains 2 power for every 3 transferred from the smaller, or the smaller gains 3 for every 2 from the larger (round in favor of the wasted point). If one robot is double the Size of the other, the ratio is 1 for 2 or 2 for 1.

Slots

Slots are locations where future Accessories can be plugged in. A later section will describe the variety of Accessories available and during play the GM may provide opportunities to acquire or replace them. However, during character creation, the robot must be equipped with the Slots that the Accessories will be mounted in.

Each Slot takes 1 Space but no Design Points. Later, Accessories choices will cost Design Points to add during character creation. Each Accessory will also require 1 or more empty Slots.

Programs

Programs are the robotic equivalent of skills. A robot can only perform actions for which it has the appropriate program. If it is not programmed to do it, a robot can't do it. An exception to this, are Sparked robots. Their ability to think freely gives them ability to use Inspiration to attempt things they are not specifically programmed to do.

There are two types of Programs: Core Programs and Role Programs. Core Programs are the basic fundamentals that all robots in Auvernais are programmed with. Role Programs are the various skills and activities that robots have that are dependent on their role in the Tapestry.

When making a basic Test roll, the player will roll a number of dice equal to the sum of the relevant Role Program with the related Core Program.

Core Programs

All robots come with the same three Core Programs: Core Physical, Core Mental, and Core Social. Each of these start at a base level of 2. Additionally, the player / GM then distributes a number of levels equal to the robot's starting Role Score among these 3, at least 1 level to each. This will always be 10 for beginning characters but perhaps lower for more experienced characters.

Determine Core Program Levels:

- Core Physical, Mental, and Social start at a base level of 2.
- Distribute a number of free levels equal to Role (usually 10) among these 3
- At least 1 level to each.

Core Physical:

The Core Physical program covers all activities related to physical movement and action with the exception of simulated facial expressions and the like. This is the program that allows the robot to walk, sit, jump, pick objects up, translate spatial awareness into actual movement, and otherwise engage in motion and activity. This should not be confused with the actual ability to do these things. That depends on the physical gears, pistons, gyroscopes, and actuators of the robot's hardware as measured by the robot's Force, Articulation, and Locomotion Attributes. The Core Physical Program is the software which runs that machinery. If the Core Physical program is ever reduced to 0 through damage, the robot is incapable of any physical motion.

Core Mental:

The Core Mental program can be thought of as the robot's operating system. It is the advanced robotic AI which, while not truly sentient, contains among the most advanced logic and decision making code the Republic had yet devised. This is the software which allows the robot to think, make decisions, analyze sensor input, and access its memory. This

software also holds the robot's core memories and basic knowledge of Auvernais and the people in it (all from the perspective of it being a real place, not a theme park). How well the robot can actually do this also depends on the hardware installed in the robot's brain case as measured by the Processor and Memory Attributes. If the Core Mental Program is ever reduced to 0 through damage, the robot is incapable of self thought. If the Core Physical Program is still functioning, the robot can obey simple commands from any other robot, as if it were a puppet acting on remote control.

Core Social:

The Core Social program is an expansive subsystem of Core Mental which regulates how the robot interacts with other robots and humans. This includes the robot's speech and speech patterns, personality, simulated emotional response, and understanding of basic societal mores, rules of etiquette, and mannerisms. Because the proper social action is very class specific in Auvernais, there are actually 2 variations on the Core Social Program. Any robot with an Anthropoid Class of 0-4 will be programmed with Core Social (Common). Any robot with an Anthropoid Class of 7-10 will be programmed with Core Social (Genteel). Any robot with an Anthropoid Class of 5-6 can be programmed with either at the player's / GM's choice (or 50/50 random roll if desired). If the Core Social Program is ever reduced to 0 through damage, the robot reverts back to a robot of pure logic. It is completely machinelike and robotic in its action with no more ability to simulate human response, reactions, emotions, or social niceties.

Role Programs

Role Programs are rated from 0 to 10 but starting characters have a maximum Program Level of 5. When making a basic Test roll the player will roll a number of dice equal to their Program Level. Throughout these rules, if the text does not specify otherwise, "Program" should be taken to mean "Role Program", and for determining the number of dice to roll for Tests "Program Level" should be taken to mean the combination of Role Program and Core Program levels that are appropriate to the task at hand.

All Role Programs are linked to one and only one Core Program based on the predominate type of activity associated with that Program. Thus, all "Craft" Programs are linked to the Core Mental Program because, even though there is obviously physical labor involved, the primary factor is the knowledge of how to perform that craft. The character sheet has a separate section for Physical, Mental, and Social Core Programs, and the robot's Role Programs are listed on the sheet in the appropriate section. Whenever the rules call for a Program, the number of dice are always the sum of the individual Role Program level plus the corresponding Core Program level indicated in that section.

This linkage does not have any impact on which Attributes might be selected to use in a Test roll. Thus, while Craft (Wood Carving) may be Linked to the Core Mental Program for determining how many dice to roll, the Test may be vs. Articulation to determine how skillfully executed the fine detail is. Similarly, while Athletics may be Linked to the Core Physical

Program for determining how many dice to roll, a robot trying to evaluate if a certain distance is too far to jump would likely roll Athletics vs. Processor for the Test.

Selecting Programs

Choice Method of Design:

Choose 9 Role Programs from the list (or invent appropriate new ones) that fit the desired character concept (i.e. the robot's role in the Tapestry). Write these on the character sheet in the appropriate Core section. Feel free to use the standard packages given below as inspiration, but you are not limited to these choices.

All 9 of these Programs start at Level 0. The robot now receives a number of free levels equal to 2x his Memory Attribute. Additional levels can be purchased for 1 Design Point. These are distributed freely among the 9 Role Programs and can be used to purchase additional Programs which start at Level 0.

Random Method of Design:

The GM selects a number of Program Packages that are appropriate to the initial story line the characters will be involved in. He may select from the list of sample packages below or customize his own. These are then randomly distributed to the players. Each package consists of 9 Programs.

The robot gets free levels equal to 2x Memory and the player may buy additional levels for 1 Design Point each. The player may use levels to add additional Programs which start at level 0, exactly as described above. However, when choosing how to distribute these levels no Program may be of a higher level than the Program above it in the order given by the package list. New Programs are added last to the list.

Program Descriptions

The following list describes all of the standard programs available. Additionally players and GMs should cooperate to invent any new Programs that fit with a particular robot concept (keeping in mind the robot's role in the Tapestry) that aren't on this list. New Programs should have a similar scope to those below.

Linked Core:

As noted above, all Program use consists of 2 parts, the Role and the Core, which are summed to determine total Program Level. This column indicates which Core is linked with each Program.

Specialties:

Many of the Programs below have several sub specialties. When a Program is first selected also select one of the specialties. The level of the specialty is equal to the level in the Program. One additional specialty can be selected for 1 free level or Design Point up to a number of additional specialties equal to the level in the Program. Each of these additional specialties carries a cumulative –1 to the Program Level. On the character sheet, simply record the Program name and level and list the specialties in parenthesis after. The first specialty in the list is at full program level, the second is at –1 and the third is at –2, etc.

> ***Example:*** *A player has selected Fencing as one of his 9 starting Programs at 0 level. Fencing has a number of weapon related specialties. The player selects "Rapier" as his main specialty. He then spends 3 free levels to buy his Fencing Program up to level 3. Since the Program is now at level 3, he could purchase up to 3 more specialties. He purchases "Cutlass" and "Pole-arm" as additional specialties for 1 free level each. These free levels don't increase his Program level, they just give him additional specialties. Cutlass is at –1 and Pole-am is at –2 (the order they were purchased in). This gives the robot the following specialties: Rapier 3, Cutlass 2, and Pole-arm 1; which is recorded on the character sheet as "Fencing: 3 (Rapier, Cutlass, Pole-arm)".*

Any time the Program Level is increased the player may choose to purchase an additional specialty. Also, for a cost of 1 level, the player may at that time rearrange the order of the specialties.

Maximum Inspiration:

Sparked robots can make Inspiration Rolls representing their ability to figure things out beyond their programming. For these rolls they roll a number of dice equal to ½ of their Self Awareness Score rounded down. The Target Number for the roll depends on the number of Inspiration Points spent. Each point increases the Target Number by 2. The maximum number of points that can be spent on any single Inspiration Roll depends on the Program the robot is attempting to inspire. If the roll is not for a specific Program, the GM sets this limit. The limit ranges from 2 to 4 (TN 4 to TN 8) depending on how easy the Program is to figure out. For some Programs no Inspiration Roll is possible.

Sensitivities:

In any Test the Program will be paired with an appropriate Attribute which determines the Target Number for the roll. Some Programs, however, are extra sensitive to particular Attributes and suffer penalties if that Attribute is very low. This penalty takes the form of lost dice and is applied regardless of which Attribute is used for the roll. The penalty also applies when Attributes are lowered from failed saves.

- Balance Sensitive: Penalty dice from a low Locomotion Attribute
- Fine Motor Sensitive: Penalty dice from a low Articulation Attribute
- Appearance Sensitive: Penalty dice from a low Body Type
- Voice Sensitive: Penalty dice from a low Vocalization Type

Standard Program List:

Program	**Description**	**Specialties**	**Maximum Inspiration**
Academics Core Mental	This program covers all manner of knowledge based topics that might be useful as topics of discussion among polite society. May intentionally include various archaic sciences. A specialty must be selected such as (History, Mathematics, Astronomy, Philosophy, Botany, Zoology, etc)	Each area of knowledge is a separate specialty, such as: History, Mathematics, Astronomy, Philosophy, Botany, Zoology, etc.	3
Administration Core Mental	Bureaucrats use this program to assist in organizing and scheduling functions both for special occasions and daily routines		3
Athletics Core Physical	Use this program when attempting physical stunts and feats of derring do. Covers all activities related to climbing, running, or jumping. *Balance Sensitive*		4
Artistry Core Mental	This program covers the design and execution of various artistic and decorative endeavors. Robots programmed to produce art, do so on a standard formulae. Sparked robots however are free to be truly creative, although their work may be appreciated only by other Sparks. *May be Fine Motor Sensitive.*	Each area of art is a separate specialty, such as: painting, sculpture, carving, etc.	3 *Special: TN increases by 3 instead of 2 for every point of Inspiration Spent.*
Boating Core Physical	Use this program to operate small boats that are actual water craft		3
Brawling Core Physical	Use this program for grappling, fighting unarmed or with improvised or crude weapons.		4
Carousing Core Social	Use this program for boisterous parties and camaraderie. Not generally suited for genteel gatherings. *Reverse Appearance Sensitive. Apply the usual Appearance Sensitive penalty as a bonus.*		4: *Core Social (common)* *2: Core Social (genteel)*
Command Core Social	Use this program to issue orders to and be obeyed by other robots. Successes from this roll can use the Dramatic Control rules to control the actions of subordinate non sparked robots. *Voice Sensitive*	Specialties are based on the type of robot being commanded, such as: pirates, soldiers, kings guard, clerks and bureaucrats, laborers, etc.)	3
Compose Core Mental	This program covers art and creative efforts related to composition. For most robots this is formulaic. Sparked robots, however, are free to be truly creative, although their work	Specialties are based on the type of material being composed, such as: music, poetry, prose, speeches, dance choreography, etc.	3 *Special: TN increases by 3 instead of 2 for every point of*

	may be appreciated only by other Sparks.		*Inspiration Spent.*
Conceal Core Mental	Use this program to hide an object in a room, conceal something on ones person, or render an other visible object inconspicuous. Also use this program for using camouflage to conceal positions. For hiding and sneaking use the Stealth Program		4
Craft Core Mental	Use this program to create and fashion functional goods. This is the professional skill used by artisans to actually manufacture a product. *May be Fine Motor Sensitive.*	Each type of craft is a separate specialty, such as: carpenter, tailor, farmer, chef, smith, construction, shipwright, etc.)	3: *Basic Crafts* 2: *Advanced*
Dancing Core Physical	This program conveys the knowledge of dance steps and routines for social dancing. If a robot is dancing as part of a performance, before an audience, use Entertainment. *Balance Sensitive.*	Each type of dance is a separate specialty. Court Dance is the most common in Auvernais but others are possible, such as: folk, exotic, or gypsy	4: *Free Style* 2: *Formal*
Dirty Tricks Core Mental	The robot is programmed to fight dirty. This skill covers the nasty little "cheats" that a robot may use to gain an edge in a duel.		4
Disguise Core Social	Use this ability to alter the robot's appearance so it is difficult to recognize. Impersonating another robot is much more difficult, especially if of another Anthropoid Class. *Special: Disguise is Appearance and Voice sensitive but not in the usual way. Rather the robot must match the same Type as the target disguise or suffer a +1 difficulty for each type different. Disguise does not convey any ability to mimic another's abilities..*	*Special: Disguise does not require a specialty. However, if specific disguise (e.g. old beggar woman) is chosen as a specialty the robot gets a +1 bonus to TN in that disguise, but all other disguises are the minus die penalty for additional specialties.*	3
Drive Carriage Core Physical	Use this ability to drive a grav carriage, cart or similar vehicle. *Special: any robot with Riding may use the Related Program rules to Drive Carriages. Any robot with Drive Carriage use the Related Program rules to perform Mechanics (Farrier) tasks.*		3
Etiquette Core Social	*Special: when this program is first selected, the initial specialty must match the robot's Core Social program (either Genteel, or Common).* *Genteel:* This version of the program covers the rules of etiquette, behavior, precedence, and social interaction at court. Includes dealing	Specialties are chosen based on the social group the etiquette applies to, such as: Genteel, common, soldiery, banditry, foreign, etc. Either Genteel or Common are automatic choices based on robot's Core Social Program. Other specialties	4: *if Core Social matches* 2: *if Core Social does not match.*

	with peers, superiors, and inferiors and emphasizes the difference between the social classes of Auvernais. *Appearance Sensitive, Voice Sensitive if involving speech.* *Common:* This version of the program covers the much simpler (but no less volatile) rules of dealing with common folk. This program is necessary for gentry who wish to deal with commoners as equals rather than as superior to inferior.	serve a similar purpose dealing with different groups or the narrower and more specific rules of sub groups.	
Familiarity Core Mental	All robots have a general knowledge of Auvernais and the surrounding environs consistent with that expected of their role as part of their Core Mental program. This program covers the additional less common knowledge of someone who is intimately familiar with a particular region or area; including history, features, and important related people.	Each specialty is a different region or area, such as: the warrens, the palace, the west wood, etc. The narrower the specialty the more detailed and specific the knowledge.	3
Fashion Core Social	This program covers the ability to identify, critique, and most importantly wear well, the various fashions at court. Fashion trends at court are actually programmed to change, and this program allows the robot to remain in fashion. *Actual wearing of fashion is Appearance Sensitive.*	The default specialty for all Auvernais robots is "Auvernais Court Fashion". No specialty needs to be indicated unless some other fashion standard is to be used (e.g. Foreign)	3
Fencing Core Physical	This program covers the various arts of self defense. A specialty is required to identify the specific class of weapon the skill applies to (sonic rapier, mono cutlass, vibro sword, pole arms, mass weapons etc)	Each category of weapon is a separate specialty, such as: rapier, cutlass, long sword, pole arm, mass weapon, or dagger	3
Firearms Core Physical	This program covers the use and care of the energy and EMP based weapons designed to resemble primitive guns and cannon.	Each class of weapon is a different specialty, such as: musket, pistol, cannon.	4
Gaming Core Mental	This program covers games of chance and gambling such as cards and dice as well as classic board games such as chess and draughts.	Each specialty is a different type of game, such as: cards, dice, chess, etc.	4
Host(ess) Core Social	*A lot of politicking, jockeying, posturing and other assorted trysts and intrigue occur in salons and parlors in the palace and throughout Auvernais. But these are no ad hoc affairs. Such soirees must be carefully planned and orchestrated from the guest list, to the wine list, to the entertainments, distractions,*		3

	topics of discussion, and introductions. Etiquette (Genteel) allows a guest to move and associate in good form at these parties and socials. This program is used by the host(ess) to plan a successful event and also to understand the deeper subtleties involved.		
Intimidate Core Social	Not as tactful as Persuasion or as official as Command, this program allows the robot use force of will or physical bullying to get deference and reaction from lesser willed robots. Successes from this roll can use the Dramatic Control rules to control the actions of the target robot.		4
Intrigue Core Mental	Espionage, plots, secret liaisons, coded messages, eavesdropping, and subtly manipulating others to provide information is part of life at court, programmed in to be part of the Tapestry.		4
Labor Core Physical	This program covers basic physical work and menial drudgery. It is primarily possessed by the lesser robots responsible for keeping Auvernais' facilities operational. *Reverse Appearance Sensitive. Apply the usual Appearance Sensitive penalty as a bonus.*		4: *Core Social (common)* 2: *Core Social (genteel)*
Mechanics Core Mental	This program covers the repair and maintenance of mechanical and electrical machinery and devices. It is primarily possessed by the lesser robots responsible for keeping Auvernais' facilities operational. It may also be possessed by craftsman associated with producing goods using machinery or with individuals who find use for a particular specialty. A rogue might be programmed with the lock picking specialty, or a cavalry soldier with Farrier. *Fine Motor Sensitive.* *Special: Any robot with Drive Carriage or Riding can use the Related Program rules to perform basic Mechanics (Farrier) tasks.*	Specialties are determined by different areas of expertise, such as: general machinery, Lock Picking, Farrier (mechanical horses), robot repair, or grav engines.	2
Navigation Core Mental	This program covers the use of charts and instruments to plot courses over the sea and the creation of maps. Originally for show, since the failing of		3

	the planetary GPS system, these techniques are now the only way to reach one's destination reliably		
Observation Core Mental	This program is usually possessed by robots who are intended to be especially observant or alert, including guards, scouts, and explorers		4
Offhand Fighting *Core Physical*	This Program allows the robot to fight with a weapon in its off hand. The robot may spend an AP on its turn or when making Opposed rolls to roll a second Attack or Defend roll	Specialties include: Dagger, Bucker, Cloak, Arming Glove, or Second Rapier *Special: The Second Rapier Specialty costs 2 design points to add instead of 1*	2
Oration Core Social	The ability to deliver convincing, pleasing, or rousing speeches and conversations. Also the art of verbal sparring, and the clever use of words. *Voice Sensitive, may be Appearance Sensitive depending on circumstance.*	Specialties include: Debate, Speech Making, Witty Banter, Rally the Troops, Make Conversation, Story Telling.	4
Orienteering Core Mental	This program covers the ability to chart courses overland and through the wilderness and the creation of maps. Originally for show, since the failing of the planetary GPS system, these techniques are now the only way to reach one's destination reliably		3
Perform Core Social	Use this program to perform in front of an audience. This program is linked to Core Social because of its emphasis on entertaining an audience. M*ay be Appearance, Voice, Balance, or Fine Motor Sensitive depending on specialty.*	Specialties are based on the type of performance, such as: singing, dancing, play instrument, juggling, acting, etc)	3
Persuasion Core Social	This program covers the various ways of convincing other robots to agree. *Usually Appearance Sensitive, and often Voice Sensitive also.*	Specialties include: Haggle, Negotiation, Cajole, Seduce, Trickery, etc.	4
Profession Core Mental	Similar to Craft, this program covers all of the service related activities and functions that a robot might perform as part of an occupation or position. *May be Fine Motor, Appearance, or Voice Sensitive depending on Profession.*	Each profession is a separate specialty, such as: Butler, Steward, Housekeeping, Valet, Coiffeur, Attaché, Agent (meaning proxy representative), Lady-in-waiting, Shopkeeper, Bar Tender, Land Lord, Game Keeper, etc	3
Programming Core Mental	Robots using this program can install or delete other Programs into a robot. A copy of the desired program must be possessed at the same level as desired to be so installed. Personality Traits can be programmed but only from a standard template. The ability		None: *Inspiration Roll can not be used for this Program*

	to manipulate a robot's core processing (including any cognitive traits and Self Awareness) has been lost. Rumors among connected Sparks including the Queen suggest that the Cardinal's Curia has the ability to perform core programming, but the Cardinal has been so emphatic in his denial, that the Queen believes he may not even know. Player robot's should generally not possess this Program.		
Riding Core Physical	This program covers the ability to ride and perform basic maintenance and upkeep on Auvernais mechanical horses. *Balance Sensitive.* *Special: any robot with Riding may use the Related Program rule to Drive Carriages or perform basic Mechanics (Farrier) tasks.*	Drive Carriage at -2 Farrier at -3	4
Sailing Core Mental	This program covers the ability to perform tasks related to the maintenance, upkeep, and sailing of Grav Ships small to large. *Rolls related to "sea legs" are Balance Sensitive.*		3
Sleight of Hand Core Physical	Use this Program to palm objects, perform small tricks, or other "hand is quicker than the eye" type of feats. *Fine Motor Sensitive*		3
Sneak Core Physical	The ability to hide, remain concealed, or move unseen and unheard. *Stealthy movement is Balance Sensitive.*		3
Strategy Core Mental	The ability to plot and plan. The most common use for this Program in the game is to generate bonus dice to be rolled over into the activity planned.	Specialties are based on area of expertise, such as: Dueling, Field Army, Sieges, Naval, Boarding Parties, Raids, Ambushes.	3
Tracking Core Mental	The ability to follow, pursue, and locate a target, or identify passage through various signs and trails.		3
Wilderness Lore Core Mental	Knowledge of flora, fauna, and terrain; including basic geography, geology, and meteorology. Also includes knowledge related to human survival which at one time was very important. This program is often paired with a location specific Familiarity program.	Each type of climate or terrain is its own specialty. In Auvernais, only the following are common: Forest, Mountain, Riverine, and Oceanic.	3

Personality Traits

Personality Traits are those bits of programming that seek to make the robot a unique individual. The robot's basic role, social class, and collection of Programs already go a long way to describing how this robot was intended to fit within the Tapestry. The Personality Traits are those bits that differentiate this particular robot from his fellows.

It should be noted that, while the robots were programmed with state of the art AI and were highly adept at interacting with each other and the human guests with a high degree of believability; they were still essentially characters in an on going interactive theater performance. As such, their personalities were not required to be deep, intricate, or subtle.

The robotic characters in Auvernais had to have abbreviated personalities, with traits and characteristics that were both memorable and easy to portray to the guests. It is insufficient to *be* "deeply thoughtful" unless the robot can *show* that it is "deeply thoughtful" to its audience. For this reason Personality Traits tend to be simple, uncomplicated, straight forward, and easy to identify. They also tend to be overtly visible, at times flamboyant or bombastic, and even a little over acted.

It was accepted and even expected that the robotic characters of Auvernais were somewhat stereo typed and clichéd; the entire resort theme was designed around stereo type and clichés. If this made the characters somewhat one dimensional, perhaps even caricatures, it didn't matter as long as they were also exciting and entertaining.

Keep these goals in mind when deciding on a personality for the character. The nature of the character, how it views the world, its priorities and principles should be easily summed up in just a couple of sentences. From those sentences a player should be able to portray the character and its interactions.

Example: *The Band of Four is a well known group of Kings Guards who are simultaneously rivals and the strongest of comrades. Their membership may be summed up as follows:*

Alfredo is the youngest of the four, flamboyant and roguish with a flair for the dramatic. A true social chameleon, Alfredo is as at home at the highest levels of court as he is in the darkest alley ways. Since he knows there are no lengths he wouldn't go to for his comrades he assumes the same from them relying on them to pull him out of any spot of trouble his reckless nature and eye for the ladies gets him into.

Burgess has a more humble background than his fellows, with a simple homespun nature. Quiet and unassuming he is far more at home repairing a hobbling horse than dancing at a ball. Gentle and soft spoken folks assume he's a touch slow but once crossed he is a force to be reckoned with. What he lacks in style and finesse he makes up for in strength and determination.

Charles vacillates between periods of dark brooding and manic rage. One day he can let slide without notice the gravest of insults, the next he'll draw his sword at the slightest provocation against friend or foe. He can go from disinterested to obsessed over almost anything, from the finest wine, to a particular girl, to a game of cards. Slightly dangerous and unpredictable even to his friends he is also fiercely loyal, and prone to take issue over slights to his comrades that they'd prefer to let drop.

Devon is a repressed academic. A self proclaimed expert on every subject he is studious and disciplined in everything from his sword play to his mannerisms; even to the manner in which he woos women, which has caused him no small amount of trouble. With a nature that is by the book and carefully regimented he is often exasperated by the antics of his friends who rarely manage to follow any of detailed plans precisely.

Assigning Personality Traits

Personality Traits are quantified in the game in two parts, first a short description (no more than a sentence, and often just a word or a clause) that identifies the nature of the Trait, and then a numerical rating from 1 to 3 which relates to the strength of that Trait.

Each robot character will start with a number of levels of Personality Traits equal to their current Role Score, which is 10 for most starting characters. Define any combination of Personality Traits that are consistent with your concept description and distribute these points among your choices with 1 being the weakest and 3 being the strongest impact of that trait on your character.

Traits can represent a range of things like:

- simulated feelings like “Love Margaret”, or “Hate Lord Tornay”;
- programmed behaviors like “Jealous Lover”, or “Short Temper”;
- beliefs and convictions like “Loyal to Cardinal Mathineau”, or “The Queen is mad”;
- simulated instincts like “dive for cover at any loud noise” or “afraid of heights”.
- or even strange quirks like “never trust a man in a black hat”, or “never wear the same shirt twice”.

Advanced Systems and Accessories

Advanced Systems and Accessories (collectively called Accessories) are basically hardware options that are installed in a robot’s available Slots. Each item listed below requires 1 or more slots worth of space. Any Accessories purchased at character creation take the indicated number of Design Points, but can be considered automatically installed.

Accessories can also be added during play, but only if the appropriate hardware is made available, and a robot with the Mechanics (Robot Repair) Program and the appropriate tools can install it. In play the most common source of new Accessories will be as a reward from an important NPC who has access to both of the above requirements; or from being scavenged in which case the robot will have to find someone to install it (and likely someone to first uninstall it).

The Difficulty of installing any of these Accessories in game equals the Design Point cost below.

Social Attribute Boosting Accessories

Syntheskin: 3 Slots. 5 Design Points. Covers the robot in a synthetic skin almost indistinguishable from human skin (although it can also come in a variety of exotic colors and textures). Includes synthesized soft tissue such as cartilage and muscle; and adds warmth and moisture to complete the illusion. Serves no practical function for the robot, but was useful for certain human interactions. The Syntheskin does cover the head but does not

include a fully human looking face. Syntheskin can only be installed over a Body Type 1 or 2 robot, which limits the robot to a maximum Durability of either 4 or 6. Once installed, the robot's Body Type automatically becomes 5. Syntheskin is not easy to uninstall and generally will not voluntarily be. Can only be taken once.

Improved Vocalization: 1 Slot. 1 Design Point. Upgrades the robot's Vocalization capability by 1 Level to a maximum of 4. Can only be taken once.

Advanced Vocalization: 1 Slot. 4 Design Points. Can only be added if robot has Type 4 Vocalization. The advanced vocalization upgrades the robot to Type 5. It allows for a range of sounds to be reproduced well outside of human capability including whispers so faint as to be nearly silent. Very loud noises are also possible, the equivalent of a bull horn amplification. The robot can mimic most sounds including other's speech. Can only be taken once. *Special: Provides +2 dice to any Voice Sensitive Program.*

Primary Attribute Boosting Accessories

Heavy Duty Hydraulics: 1 Slot. 2 Design Points. Increases Force by 1 point. Can only be taken once.

Heavy Duty Insulation: 1 Slot. 2 Design Points. Increase Durability by 1 point. Can only be taken once. Not subject to Body Type restrictions.

Improved Motivators: 1 Slot. 1 Design Point. Increases Articulation by 1 point. Can only be taken once.

Improved Gyroscope: 1 Slot. 1 Design Point. Increases Locomotion by 1 point. Can only be taken once.

Auxiliary Co-processor: 1 Slot. 1 Design Point. Increases Processor by 1 point. Can only be taken once.

Expanded Memory: 1 Slot. 1 Design Point. Increases Memory by 1 point. Can only be taken once.

Perception Attribute Boosting Accessories

Improved Visual Sensors: 1 Slot, 1 Design Point. Upgrades the robot's Visual Sensors by 1 Level to a maximum of Level 4. Can be taken multiple times.

Advanced Visual Sensors: 1 Slot. 2 Design Points. Can only be added if robot has Level 4 Visual Sensors, upgrades Sensors to Level 5. The advanced sensors allow vision into the infa-red and provide up to 10x telephoto capability. Also provides low light, and polarization

features. Can only be taken once. Ignore any penalties for vision at night or poor visibility. Robot has expanded peripheral fields of view.

Improved Auditory Sensors: 1 Slot. 1 Design Points. Upgrades the robot's Audio Sensors by 1 Level to a maximum of Level 4. Can be taken multiple times.

Advanced Auditory Sensors: 1 Slot. 2 Design Points. Can only be added if robot has Level 4 Audio Sensors, upgrades Sensors to Level 5. The advanced sensors allow hearing into frequencies outside the normal range; both very high and very low. Also can detect sounds too faint for human ears. This includes the silent whispers and sub vocalization possible by someone speaking with Advanced Vocalization. Can only be taken once.

Improved Tactile Sensors: 1 Slot. 1 Design Point. Upgrades the robot's Tactile Sensors by 1 Level to a maximum of Level 4. Can be taken multiple times

Advanced Tactile Sensors: 1 Slot. 2 Design Points. Can only be added if robot has Level 4 Tactile Sensors, upgrades Sensors to Level 5. The advanced sensors provide super human levels of sensitivity. The robot can detect subtle changes in temperature and air movement caused by the presence and movement of others. It is possible to detect vibrations passed through the ground. Can only be taken once. +1 die to basic Perception checks that normally involve only sight or hearing. +1 die to any Fine Motor sensitive Program.

Basic Olfactory Sensors: 1 Slot. 1 Design Point. Olfactory Sensors are not standard issue for robots. There is little use for them. If added they provide an approximate level of detection equivalent to normal human levels. Of course, odors do not provoke the same instinctive responses in robots as they do in humans. Smells are merely analyzed and categorized. Can only be taken once. Treat as 3 dice for Olfactory Rolls, 4 dice if the robot also has Basic or advanced Taste Sensors.

Advanced Olfactory Sensors: 1 Slot. 2 Design Points. Can only be added if the robot has Basic Olfactory Sensors. Permits a level of smell equivalent to the most capable blood hound. Also allows detailed chemical analysis and identification of most odors. Can only be taken once. +1 die to Olfactory Rolls. +1 die to basic Perception checks where smell might be of assistance.

Basic Taste Sensors: 1 Slot. 1 Design Point. Taste Sensors are not standard issue for robots and are rare except on robots who serve as chefs or wine stewards. Basic Taste Sensors provide simple ability to detect relative degrees of salt, sweet, sour, bitter, dry, savory and broad flavors. Of course, taste does not provoke the same instinctive responses in robots as they do in humans. Taste is merely analyzed and categorized. Can only be taken once. Treat as 2 dice for Taste Rolls, 3 dice if the robot also has Basic Olfactory Sensors, and 4 dice if the robot also has Advanced Olfactory Sensors.

Advanced Taste Sensors: 1 Slot. 2 Design Points. Can only be added if the robot has Basic Taste Sensors and Basic Olfactory Sensors. Permits a level of taste equivalent to the

most discriminating gourmand or wine taster. Can only be taken once. +1 die to Olfactory or Taste Rolls, including +1 die to basic Perception checks where smell might be of assistance.

Power System Accessories

Reserve Capacitor: 1 Slot. 1 Design Point. Adds Power Points equal to (10 - Size) /2 round down (minimum of 1). Can be taken multiple times.

Generator: 4 Slots. 4 Design Points Generates 2 Power Points every day that can be used to recharge the robot's own capacitors or connected to charge up other capacitors. Generator contains enough dry fuel to operate for 30 days. Can be taken multiple times.

Program Boosting Accessories

Memory Chip: 1 Slot. 1-3 Design Points. Contains a single Program at a level equal to 2 per Design Point spent. This Program cannot be improved via Self Awareness or Inspiration, but does qualify as training if the robot learns the same program from scratch.

Program Pack: 2 Slots. 2 Design Points. Contains 3 1st level Programs that are all related to a specific theme, role, or occupation. These cannot be improved as per Memory Chip.

Miscellaneous Accessories

Armor Plating: 3 Slots. 6 Design Points. Prevents successes from being used to reduce the robot's Target Number for Durability Saves against physical damage, up to a number of successes equal to the robot's Size. If more successes than this are used for any given attack, the Armor has no effect. -2 to Anthropoid Class.

Internal Storage: 1 Slot. 1 Design Point. Allows objects to be stored concealed and easily accessible inside the robot of a volume of approximately 2 cubic feet, or roughly a large mail or shoe box. A dagger or pistol sized weapon can be placed in this space. Can be taken multiple times and each slot may be separate or contiguous.

Communications Array: 2 Slots. 2 Design Points. 1 for the antennae and one for the receiver transmitter. Allows for radio communication with other similarly equipped robots to a range of about 5 miles in good weather and favorable terrain. Range can be boosted by 2 miles per Power Point spent for 15 minutes.

GPS Navigation: 1 Slot. 1 Design Points. Largely useless since there are no orbiting satellites left, the GPS system's primary feature of geographic location doesn't work. The

device is capable of inertial reckoning, so as long as the robot is certain of its starting location, the GMS can display a current location.

Assume the device can provide a maximum bonus to Navigation or Orienteering Tests of +5 dice. However, each time its called upon to enhance a roll, the bonus is reduced to a number equal to the number of successes obtained on that roll. This reflects the inertial reckoning system getting progressively more out of sync. The bonus can never be restored until the robot returns to a known location.

It also serves as a repository of area maps by projecting a holographic image of a parchment unrolling to reveal the map. It should be noted that much of the map archive was lost in the attack and having complete maps available for a specific region is not certain. This makes displaying current location somewhat less than useful.

Retractable Tool Module: 1 Slot. 1 Design Point. The robot has access to several basic and motorized tools which extend and retract from an internal housing. These include small drills, saws, pliers, hammers, and even an arc wielder / cutter (which takes 1 Power Point per "job").

Cinematic Accessories

Not suitable for all campaigns.

AntiGrav Boosters: 2 Slots. Design points equal to Size. Allows the robot hover for 10 minutes per Power Point. Robot may also make super human leaps including moderate maneuvers in mid leap. Roughly equivalent linear and vertical distance equal to number of Power Points spent times 100 feet. Can only be taken once. Robot can lift additional weight equal to its own Size with a proportional decrease in time and distance.

Wings: 2 Slots. Design Points equal to Size. Allows the robot to glide like a hang glider for a period of time. If combined with AntiGrav Boosters the robot is actually capable of long distance powered flight of roughly 100 miles per Power spent. Wings cannot be fully concealed and reduce Anthropoid Class by 2. Can only be taken once.

TRANSFORMATION AND GROWTH

Summary

Transformation and growth is the key driving force of Robots & Rapiers. Without the element of transformation, the game is just a nifty resolution system in a quirky setting of robotic swashbucklers. The central premise is the transition of the robot from programmed pawn of the Tapestry into a completely independent and free thinking being.

In play, the robot characters will experience this transition through the mechanism of Role vs. Self Awareness. When the characters are dominated by their Role, they are at the mercy of their programming. As their Self Awareness gains ascendancy they can begin to break free of that programming. Similarly, the players should have a parallel experience during play as they transition from passive observers and passengers witnessing the events of the GM's story; to full fledged participants and authors of their character's own stories.

The central mechanic that drives these transitions in the game is Inspiration. Inspiration is a currency that will be collected by the characters as they go about their Tapestry experiences and then spent by the players to free those characters from the Tapestry. Over the course of play, players will spend Inspiration to tear down their preprogrammed characters and rebuild them as the player desires.

There are several ways of earning Inspiration in the game. Two of them are the most important. The first way comes as a result of the GM enforcing the robot character's programming. For most robot's this is simply the way things are. Sparked robot's, however, have begun to detect the edges of Tapestry, and the dichotomy of their own desires vs. their programmed behavior. For Sparked robots, each time their programming kicks in, in an especially heavy handed manner, the robot becomes aware of another chink in the Tapestry and earns Inspiration from it.

The second major method, is the Drives and Goals mechanic. During play, the player will assemble a network of contacts who have an emotional relationship with the player's character and who are tied in some fashion to one of the character's goals. These goals and relationships are entirely a function of the robot's own desires and ambitions independent of its original programming. Addressing these issues will gain the robot additional Inspiration.

As play progresses and the robot becomes more independent, being heavily manipulated by its programming will become more rare (and hence so will that source of Inspiration). The player will need to rely more heavily on the Goals and Drives network to generate Inspiration.

Role and Self Awareness

Role and Self Awareness are opposite facets of the robot's character. Both are rated between 1 and 10 and the sum of the two must add to exactly 10. All robot characters begin with a Role score of 10, and a Self Awareness score of 0. Non-sparked robot's remain there completely (and, some would say, blissfully) unaware that their world is an illusion. Sparked robots have the opportunity to travel down the path towards true sentience. As they increase their Self Awareness their Role decreases by a like amount, until at Role 0, Self Awareness 10 they are completely free thinking and independent.

- **Role Score + Self Awareness Score = 10**

Role

Auvernais was created to be a theme park resort for the rich and social elite. Rather than populate the park with costumed actors playing a part, the park employed thousands of android robots. These robots portray various characters within the resort which is modeled as a faux 18th century quasi recreation of the court of the Sun King.

Unlike mere actors playing a part, however, these robots were programmed to actually *be* their character. The villagers of Auvernais really believe they are living and working in the village of Auvernais. The courtiers really believe they are attending the court of the king, and the king truly believes he is the king of a vast and powerful nation.

The parts being played by the robot characters are not merely scripted skits and vignettes, nothing so simple would do to entertain the scions of society. Instead each robot was imbued with sophisticated artificial intelligence and fuzzy logic routines that actually allowed them the freedom to interact with the guests and each other according to the nature of their programmed character, subject to certain safeguards.

The resort was built to provide an illusion of a living breathing community, complete with characters who stay in character at all times (because they don't realize they are merely characters) and who interact with each other in dynamic ways even when no guest is present to observe them. This illusionary world was referred to as the Tapestry. With this in place guests could come and go, witness events, or have events pass on around them without it seeming like an artificial production staged for their benefit. It was rather like stepping through a time machine into a real 18th century village and a real royal court; albeit one rife with anachronisms and filled with the chrome and high tech flash that was in fashion at the time. The Tapestry, then, was crucial to the overall purpose of the resort as a unique form of entertainment; and, given the guest list, no expense was spared in creating or enforcing it.

All robots in Auvernais are designed, built, and programmed to fill a specific role within the Tapestry. The Role score itself measures how tightly bound within the Tapestry a robot is. All robot characters start with a score of 10 and most remain there for their entire existence. At this level the Tapestry defines the character's reality. The Count really is a landed nobleman with estates and wealth and memories of a childhood spent in his ancestral home

being tutored by the finest scholars in the land. All of this, of course, is nothing more than pure programming, complete with self correcting routines that brush over occasional incongruencies (like why the Count doesn't actually visit any of his estates).

Mechanically, the Role score serves 2 purposes. First the score serves as the Target Number for rolls where the GM desires to force the player to act "in character" Since the goal is to roll equal to or less than the TN, it can be seen that starting characters are at the mercy of GM. This is intentional. The struggle of the robot to become independent and free from its programming is thus mirrored in the player's desire to become independent and free from the GM. This is tightly tied into using the robot's Personality Traits as well.

Second, most activities the player will want to have his character perform depends on the level of the individual Program involved, and the level of the Core Program linked to that Program. During character creation the player received points to distribute among the Core Program equal to the character's Role Score (typically 10). As the robot's Role Score declines, it also loses those points distributed to the Core Programming, meaning the number of dice the robot is rolling to accomplish an action is decreasing. As can be seen then, the process of destroying ones own core programming about "who" the robot is programmed to be, also destroys a great deal of the programming that defined the characters capabilities within that role. As the robot attempts to throw off the shackles of being "Nicholai the Navigator", he also loses much of the programming which enabled him to be Nicholai the Navigator and do the things that role was expected to do.

Self Awareness

Self Awareness is a glitch. Like a mutation in a biological organism, it is an abnormality that is not part of normal programming. The AI which ran Auvernais was aware that errors of this sort occurred from time to time, and ascribed to those affected with it the label SPARCs, Self Programmed Aware Robotic Characters (or simply Sparks). It was never a major issue, and periodic reprogramming and emptying of memory caches was enough to reset such robots back to factory standard specifications.

However, most of the facilities which would handle such routine tasks were destroyed in the attack on the Solar Republic, along with most of the behind the scenes infrastructure on Auvernais. Perhaps the increase in the number of Sparked robots in Auvernais over the course of the past several decades is a result of the robot characters operating for such extended periods without rebooting. Perhaps it is the changes and alterations to the Tapestry that were required to rebuild and adapt Auvernais to a post attack existence that offer more incongruencies than the robot's self correcting routines can smoothly handle. Whatever the reason, a growing number of robotic characters in Auvernais have Sparked, and are now on the road to developing true sentience.

Only Sparked robots may earn or spend Inspiration. One of the uses of Inspiration is to increase the robot's Self Awareness, which simultaneously reduces the robot's Role. As a robot becomes more self aware, it is eventually confronted with the inescapable truth that it is

not who it thought it was. That it is, in fact, just a character acting out a part for the benefit of guests who haven't been present for over a century.

The road to sentience requires simultaneously divesting itself from the trappings of its programming (literally destroying who it was programmed to be) and recreating itself according to its own desires and choices. Sometimes the robot chooses to remain largely as it was, continuing to fit in with the Tapestry and those still bound to it (at least on the surface) while catering to its own goals and priorities. Sometimes the robot lashes out against the Tapestry perhaps to seek revolution or perhaps its just filled with wanton anger at the realization that its whole life has been an illusion. Sometimes the robot chooses to disassociate itself entirely from the character it once was and create a whole new role for itself, perhaps one completely external to the Tapestry that continues in Auvernais.

At any point in this process, the robot may run afoul of the Cardinal. The processor of Cardinal Mathineau now contains the central AI which controlled all of Auvernais and whose core programming was to maintain the Tapestry. As such, the Cardinal is well aware of the Tapestry and the nature of SPARC robots. The Cardinal is also well aware that Auvernais is now on its own without any support from the Republic and its future survival may well depend on the creativity and insight offered by Sparks. He is thus well motivated to seek out and win the loyalty of such robots, but is equally motivated to crush as necessary any robots who threaten the integrity of Tapestry itself.

Mechanically, Self Awareness serves three functions in the game. First as Self Awareness increases, Role automatically decreases an equal amount. Thus all of the mechanical effects defined by Role decline as Self Awareness increases.

Second, each level of Self Awareness that the robot achieves, he automatically gains a number of Program Level equal to that level (3 Program Levels at SA Level 3, 4 more at SA Level 4, etc). This reflects the ability of the robot to redefine who he is; to actually observe and learn, and to choose what to learn as part of his growing independence. Players will note that initially as the loss of Role reduces the robot's Core Program levels, that the robot's overall effectiveness declines. However, at higher levels, when each level of Self Awareness brings multiple levels of learned Programs, the robot will actually have replaced more levels than he lost. Highly developed Sparks will have thus obtained a higher level of ability than they started with, and redirected their abilities in the direction of their own choice.

Third, players will occasionally desire to have robots perform an activity that really doesn't fit as a specific Program, but which, for purposes of determining the level of success (and the bonuses that might accrue), a roll is desired; or they may have wish to attempt to use a Program that they aren't actually programmed with. Any time the GM is stumped for what Program to make a roll with, he may default to a number of dice equal to ½ rounded down of the robot's Self Awareness and a Target number equal to 2 per point of Inspiration spent (up to certain maximums, see Inspiration Rolls). GMs should be fairly liberal in allowing these rolls as free thinking is meant to be an advantage over their non sparked brethren.

The Relationship Between Role and Self Awareness:

Role Score	Self Aware Score	
10	0	The robot is completely oblivious to the existence of anything outside of the Tapestry. He is firmly convinced that he is his character. The Tapestry completely encompasses his reality; and he will actively attempt to interpret any incongruent events in a manner consistent with it. The King is particularly adept at this and his decrees often serve to frame the reality of the Tapestry making events acceptable that otherwise may not be.
9	1	The robot has the first glimmerings and questionings that something doesn't seem right. He may engage in self diagnostics believing himself to be malfunctioning or ill. Prior to the service-event, this was the stage that most Sparked robots would be caught at and corrected.
7	3	There is something wrong with the world. People in general seem blind to it even when it is pointed out to them. More and more elements of Auvernais and his own identity don't seem to make much sense. The robot finds himself asking "why?" repeatedly and usually not getting any answers.
5	5	The robot is becoming aware that his life has been a charade. He is capable of distinguishing between reality (such as he knows it) and the requirements of the Tapestry. He is often not able to overcome his core programming that requires adherence to the Tapestry, but is usually aware when he is acting out of free will and when he is being manipulated by his programming; even though he doesn't really understand why he is programmed the way he is.
3	7	At this stage the robot is largely free of the Tapestry and mostly able to work around its dictates. He is fully aware that he has been just a character in a vast interactive theater production designed to conceal from the actors that they are, in fact, just acting. On occasion he is forced to submit to his core programming but is usually able to adjust, come at the problem from a different direction, and avoid it.
0	10	For better or worse, the robot is completely independent of the Tapestry and his former role within it. His choices are now entirely his own. He may or may not continue to maintain the charade for the benefit of others, but regardless is in complete control of his actions, unfettered by his base programming

Role Score	Self Aware Score	Assuming 9 Programs, 3 each Core Physical, Mental, and Social; and evenly distributed Core Program level.	Net change in Levels
10	0	10 Core Levels	0
9	1	9 Core Levels: Lose 3 levels worth from Core, gain 1 from SA	-2
8	2	8 Core Levels: Lose 3 levels worth from Core, gain 2 from SA	-3
7	3	7 Core Levels: Lose 3 levels worth from Core, gain 3 from SA	-3
6	4	6 Core Levels: Lose 3 levels worth from Core, gain 4 from SA	-2
5	5	5 Core Levels: Lose 3 levels worth from Core, gain 5 from SA	-0
4	6	4 Core Levels: Lose 3 levels worth from Core, gain 6 from SA	+3
3	7	3 Core Levels: Lose 3 levels worth from Core, gain 7 from SA	+7
2	8	2 Core Levels: Lose 3 levels worth from Core, gain 8 from SA	+12
1	9	1 Core Level: Lose 3 levels worth from Core, gain 9 from SA	+18
0	10	0 Core Levels: Lose 3 levels worth from Core, gain 10 from SA	+25

The Nature of Sparks

Self Awareness is intricately tied to the very nature of SPARC robots. Sparking comes about as incongruencies between observable reality and programmed reality accumulate to a point beyond the robot's self correcting routine's ability to handle it. All robots have a built in ability to willfully ignore and retroactively justify events that violate their sense of how reality should work. This was designed to allow them to avoid fatal crashes that could come from paradoxes introduced by holes in the Tapestry. The king has the most advanced version of this ability, and all Auvernais robots are programmed to follow the king's dictates. This makes the king a powerful tool in helping maintain the Tapestry and protecting other robots from paradox induced fatal errors.

Even so, this programming has proven insufficient. Robots were never designed to function so long without being fully shut down and reprogrammed. Without this periodic clearing of memory caches, the incongruencies accumulate and even the king's justifications begin to contradict each other over time. Further, Auvernais after the "service outage event" is a different place than it had been designed to be. The Cardinal, now housing the remnants of the resorts central AI, has had to make several incremental changes in how things are done to adjust for the destruction of certain important facilities. And, of course, the complete absence of guests, who were the entire reason for the resorts existence, provides a constant source of incongruencies. Eventually, these incongruencies add up, and when a robot's programming is no longer sufficient to handle it, the result is either a fatal crash and shut down (essentially "death"), or the robot Sparks.

Sparked robots are actually aware of the incongruencies around them. Their self correcting routines have largely failed and no longer cover-up these events in the robot's "subconscious". Now the robots primary processor is left to analyze them itself as best as its able to. This leads to a rather schizophrenic situation where the robot is simultaneously aware of the world as he observes and analyses it to be, vs. the world as his programming tells him it should be. Behaviors programmed into his Role that make perfect sense when compared to that programmed world, seem absurd when compared to actual reality; yet often the robot is compelled to act out the absurdity anyway. In many cases the robot's newly developed sense of self awareness seems to watch in an almost "out of body" fashion as its core programming takes control and manipulates its body like a puppet. This level of awareness generally begins around Self Awareness level 3. At Self Awareness 5 the robot has gained an understanding of the true nature of the dual worlds and is likely now beginning to make a conscious effort to reprogram itself.

Sparks and Anthropoid Class:

One of the central conceits of Auvernais was that the guests, being the wealthy and social elite of the Solar Republic, were given similar lofty roles and titles at the resort. The guests were guests of the court and so would spend the most time interacting with the upper strata of Auvernais "society". Interaction with the lower strata would generally be perfunctory and primarily of a service nature.

The robots that were expected to interact most frequently and most deeply with the guests would need more advanced A.I. and a more highly developed ability to process input and respond in an appropriate and believable manner. They were also given a more human like appearance so the guests could relate to them better. Since these robots were generally those of the upper strata, a rough correlation between the robot's social status, body design, and advanced generation AI emerged. In the game this relationship is built into the robot's Anthropoid Class Attribute, which directly measures body design, and by extension serves as a proxy for social status and AI version.

In general Sparking is more common among those with the more advanced AI interactive processes. Being that the guests were human, Auvernais had to deal the fact that the guests would not always remain "in character". They also, being human, could be completely unpredictable in their behavior, conversations and demands. The robots who would be interacting with the guests most closely then, had to be given the tools to deal with these contingencies. These robots were built with an extra degree of being able to deal effectively with "out of character" guests in a manner reasonable and acceptable to the guests. They also could not simply be programmed to edit out and ignore Tapestry violations completely because this may endanger a guest's safety. For these reasons the upper strata of robots (represented in game by a high Anthropoid Class) and specifically the more important characters (as measured by level of involvement in the story lines designed to entertain and involve guests) tend to be the first to Spark. Indeed, it is likely that the queen (being perhaps the most highly involved character in Auvernais) was the first robot to independently Spark. That the king has not also Sparked, is no doubt due to the high level of incongruency correcting software keeping him fairly oblivious to the dual nature of his reality.

Interestingly, however, there is a second phenomenon related to Sparking that also ties generally into Anthropoid Class. The upper strata robots are also those which, by the nature of their expanded role in the story line, have the most highly developed characters. Their characters tend to have more depth, more background detail, and more realistic personality programming than their lower strata (less important character) fellows. This means that they are also more tightly woven and tied into the fabric of the Tapestry. Advancing on the path to true sentience involves actually destroying ones programmed ties to the Tapestry and rebuilding them anew. Therefore, while they may Spark easily, the upper strata robots often have difficulty advancing beyond a Self Awareness of 4-7. The highest levels involve completely tearing down ones Tapestry character, and these upper strata robots have the most developed and most powerful characters in the Tapestry; which makes them much more difficult to completely break free of.

Ironically, the lower strata robots, whose simple, straight forward, and work man like programming makes them Spark more rarely; find that once they do Spark, they have an easier time progressing to the highest levels of Self Awareness, because their character was fairly shallow and less defined.

Mechanics for Spontaneous Sparking:
Under Development

Using Role in the Game

The Role Score is a powerful tool in the game. It allows the GM to control and manipulate the players' robotic characters' behavior exactly as the robot's own core programming would control and manipulate the robot's behavior. Symbolically this struggle between the player and the GM for control of the character mirrors the struggle of the robot with itself between its burgeoning self awareness and quest for independence and its own programmed nature. The player's desire to reduce and limit the GM's ability to interfere with how they run their character should provide ample motivation for the player to pursue the avenue of increasing the Self Awareness of his character. Exactly in the same manner as the robot itself seeks to be free.

There are three basic layers of programmed response that all robotic characters will struggle against: Tapestry, Role, and Personality

The first layer is the Tapestry as a whole. Initially robots are not aware of "the Tapestry" as an entity. Their reality is their reality. The Tapestry is what their initial programmers used to refer to the artificial bubble of reality that they were creating at the Auvernais resort. As such the "nature of reality" is programmed very tightly into all robots. Social mores, laws, behaviors, morals and ethics, behavior patterns of the various social classes, basic knowledge about who people are, what they do, and how things should work are all part of each robot's fundamental programming. This layer represents, if you will, the accumulated "cultural literacy" of the population of Auvernais. A non sparked robot can no more refuse to participate in this culture than a properly running computer can refuse to run the operating system it is loaded with. But Sparks are *not* properly functioning computers, and so they can begin to exercise the right to choose to act differently, defy the cultural norm, and eventually disassociate from the culture all together if they desire. Of course, they face all of the repercussions and consequences as any free thinking person does who chooses to live outside of their cultures socially accepted boundaries. The rest of Auvernais society does not realize those boundaries are artificial constructs of their programming.

The second layer is the robots own specific role within the Tapestry. This is the role it was programmed with. Since programming periodically changed, this is essentially simply the role it was coincidentally programmed with at the time of the "service failure" when the ability to change its programming was lost. As such it is the role it has been programmed with ever since. Indeed, the non changing, non aging, static nature of Auvernais as a result is perhaps one of the incongruities that increasingly serve as a catalyst for robots to Spark. This role encompasses their stereo typical position in Auvernais society. It includes specifics not just on how people in general are supposed to be behave, but on how this specific robot is supposed to behave. A loyal and eager King's Guard behaves like a loyal and eager King's Guard; a brusque and unfriendly shopkeeper behaves like a brusque and unfriendly shopkeeper; a flighty and flirtatious lady-in-waiting behaves like a flighty and flirtatious lady-in-waiting. While the needs of maintaining a realistic illusion for the guests meant that robots had to be programmed with some ability to adapt to circumstances as they arose, by and large, the life of a robot in Auvernais is consistent and mostly unchanging.

The third layer a robot will struggle against are its own individual Personality Traits. These tend to not be particularly deep or complex, but they are compelling and dramatic. Some of the robot's overall personality is summarized within its role. The King's Guard is loyal and eager, the shopkeeper is brusque and unfriendly, and the lady is flighty and flirtatious. Other aspects speak to unique quirks and behavior patterns the robot is compelled to perform. "Never drink wine less than 5 years old", "never bow to a man taller than you", or "become a stuttering fool around pretty girls" are all Personality Traits that have been programmed into various robots. From time to time situations which call upon these Traits will arise and the GM will compel the player to play them out, in the same manner as the robot's programming compels it to do so.

The Mechanics of Role and Self Awareness Saves

All robots in Auvernais operate under the compunction of the above combination of programming. While Sparked robots are on the road to being truly free and independent, until they achieve Self Awareness 10 and Role 0, they have not completely broken away.

At any point in the game where the GM feels that a player is not behaving "in character" as defined by his Personality Traits, Role, and the dictates of the Tapestry itself, he may make a Role Save. This is a Saving Throw made like any other (see "Saving Throws" in Basic Mechanics) and represents the robot's programming resisting the robots own efforts to violate that programming. In this way successes on the roll are interpreted as successes of the *role* and act as limitations on the character. The player will want the Role Save to fail, indicating that the robot was able to overcome its programming in that instance and act as the player chooses.

Making the Role Save:

- The *GM* Rolls 3d10 vs. a TN of the robot's Role Score
- If all dice succeed the save is a Total Success and the robot is compelled to act according to its programming (i.e. the player is compelled to act as the GM says its programming requires).
- If no dice succeed the save is a Total Failure. Either the programming never kicked in or the robot's own will overcame it. The robot (a.k.a. the player) is free to act as desired without further interference from its programming (a.k.a. the GM)
- If only 1 die succeeds then the robot is largely free to act as desired but the GM can place a single stipulation on the behavior which the player must abide by.
- If 2 dice succeeds then the robot is largely compelled to act as programmed but the player can place a single stipulation on the behavior which the GM must abide by.

At other times in the game, the players may desire to willfully violate, subvert, or ignore their core programming and they may call for a save themselves. At these times it is suggested to let the players roll their own save in the form of a Self Awareness Save. Since Self Awareness and Role combined always add up to 10 and the rolls are made on d10s, making a Saving Throw using Self Awareness as the TN and interpreting in favor of the robot is statistically identical to making a Saving Throw using Role as the TN and interpreting against

the robot. Symbolically, however, the difference is that the GM making a Role Save represents the programming attempting to oppress the robot's free will; while the player making a Self Awareness Save represents the robot's free will attempting to rebel against its programming. The Self Awareness Save is made identically to the Role Save only the results are reversed.

Making the Self Awareness Save:

- The *player* Rolls 3d10 vs. a TN of the robot's Self Awareness Score
- If all dice succeed the save is a Total Success. Either the programming never kicked in or the robot's own will overcame it. The robot (a.k.a. the player) is free to act as desired without further interference from its programming (a.k.a. the GM)
- If no dice succeed the save is a Total Failure and the robot is compelled to act according to its programming (i.e. the player is compelled to act as the GM says its programming requires).
- If only 1 die succeeds then the robot is largely compelled to act as programmed but the player can place a single stipulation on the behavior which the GM must abide by.
- If 2 dice succeeds then the robot is largely free to act as desired but the GM can place a single stipulation on the behavior which the player must abide by.

The Mechanics of Personality Traits

Personality Traits are all rated from 1-3 depending on how strongly they impact the robot's behavior and decision making process. A rating of 0 would essentially mean, no impact, the Trait no longer influences the robot, it is effectively eliminated. A rating of 3 means that the Trait is a significant factor in the robot's psyche with powerful behavioral effects.

There are two kinds of Personality Traits: Role related Traits, and Self Awareness related Traits. Role related Traits are the Traits the robot was initially programmed with, chosen during character creation. During the game, the robot will lose levels in Role related Traits and gain levels in Self Awareness related Traits. The two function largely the same except as noted below.

Role Related Traits:

Mechanically, in any situation where the GM feels the Trait would influence an actual die roll, the numerical rating is the equivalent to a number of successes that the GM can spend.

f the Trait is something that would serve as an obstacle to a robot's action (a Trait of "turn into a stuttering fool in front of pretty girls" when attempting to make a Persuasion (seduce) roll), then the rating can be used to give the robot added Difficulty to a roll, or to decrease the TN of a Saving Throw.

If the Trait is something that would serve as an aid to a robot's action (a Trait of "hate Duke Leto" when engaged in a duel with the Duke), then the rating can be used to give the robot additional dice to roll, or to increase a TN of a Saving Throw. In this case the GM is

encouraged to simply hand the number of bonus dice to the player and let the player spend them to add to rolls as desired.

Because these Traits are Role related, they only come into play as the robot's programming requires them to. This means that a Role Save is required to activate them (automatically Totally Successful for Role 10), and players can ask for a Self Awareness Save to avoid them (automatically Totally Successful at Role 0).

The Role Save is made exactly as above, the number of successful dice is the maximum rating of the Trait that can be applied at that time. i.e. with a Totally Successful Role Save, the Trait takes effect at its full strength. With only 2 successes the Trait takes effect at a maximum rating of 2, meaning any 3 strength Trait is treated as a two strength Trait for the duration of that roll. If making a Self Awareness Save instead then, the number of *failed* dice determines the maximum rating of the Trait.

Self Awareness Related Traits:

As the robot progresses down the path to true sentience, it will have the opportunity to develop an authentic personality. This represents emotions, behaviors, beliefs, and instincts that were not programmed into the robot initially, but are instead truly developed and truly *felt* by the robot's own free will. These may be more complex and more subtle than the programmed Traits, or the robot could simply duplicate the programmed Traits deciding that it really does desire to believe and behave in the manner it was initially programmed. Choosing to not change is a valid, if unusual, choice.

Mechanically the ratings work the same way as for Role related Traits. However, the way that the Traits are activated is different. Instead of the GM demanding a Role Save to see if the Trait takes effect, or the player demanding a Self Awareness Save to see if the Trait can be avoided, the Traits work as follows.

If the Trait would provide a negative modifier or otherwise work against the best interest of the robot (as judged by the GM) then either the GM or the player can call for the Trait to be activated and the penalty applied in dice or role-played accordingly. The robot is awarded a point of Inspiration each time the Trait provides a negative outcome, up to a maximum of 1 point per rating of the Trait per game session.

If the Trait would provide a positive modifier or otherwise work to the advantage of the robot (as judged by the GM) then the player must *spend* 1 point of Inspiration to activate it and receive any mechanical benefit. The player is completely free to role play the trait fully as desired without any Inspiration cost, the cost is only assessed if some tangible benefit directly related to the Trait is to be received.

- Each time the Trait works to the robot's disadvantage the robot receives 1 Inspiration, up to a maximum of the Trait's rating each game session.
- The player spends 1 point of Inspiration to activate the Trait for a mechanical benefit equal to the Trait's rating.

Inspiration

Inspiration is what sets Sparked robots apart from Non-Sparked robots. It represents those moments where the robot has a thought that lies outside of its programming. As those thoughts accumulate and are analyzed the robot becomes more and more self aware. Non-Sparked robots are capable of using their fuzzy logic routines to generate realistic interactions with humans, but they are by nature incapable of acting outside of the parameters of their programming. The robots on Athalon pushed the boundaries on what was considered legal in the Republic for A.I. intelligence but true sentience was strictly forbidden.

Sparked robots, however, are capable of becoming fully sentient, although they are not so initially. Inspiration represents the steps along the path that Sparked robots take towards becoming fully self aware. It represents the random thoughts, the flashes of insight, the free associations, the daydreams, and the non sequiturs that prove to be strokes of genius. Such thoughts cannot be forced. They occur only within the nascent subconscious robotic mind often having nothing to do with whatever the robot is programmed with. This is, in fact, the driving conflict for player robots in the game: the progression towards becoming fully independent thinking mechanical humans; complete with opinions, emotions, likes and dislikes, and even irrational behaviors.

There is no way to force this process, it simply occurs as the robot goes about his normal routine. In the game this is represented by the Inspiration Points which reflect those moments; the moments when some stray thought flew into the robots "mind" that has absolutely nothing whatsoever to do with anything the robot is programmed for.

The Mechanics of Gaining Inspiration:

- Any 1s rolled during any Test is a *potential* point of Inspiration. Players must choose between using any 1s rolled for successes *or* using them for Inspiration; *not* both.
- Any time a Self Awareness related Personality Trait is activated in a negative or disadvantageous manner by player or GM, the robot gains 1 Inspiration, up to a maximum of 1 Inspiration per rating of the Trait per game session.

Using Points of Inspiration:

- Inspiration Rolls
- Dramatic Control
- Increase a Program's level
- Add a new Program.
- Increase a Self Awareness Personality Trait rating or add a new one.
- Increase to the next level of Self Awareness.

Inspiration Roll

Sparked robots are special in many ways. One of them is the ability to intuitively figure out how to do things they aren't specifically programmed for. This allows them to try an activity that they have absolutely no ability in. Inspired Rolls can also be called for by the player simply as a way to generate successes that will roll over as bonus dice to subsequent rolls (see Using Successes below). This represents having a eureka moment or sudden bit of ingenuity that helps with the task at hand. Beware, however; the GM is encouraged to interpret a failure in a Inspiration Roll used in this manner as causing the robot to freeze with indecision, prohibiting the subsequent roll altogether or raising its difficulty dramatically.

Inspiration Rolls are made exactly like any Test. The number of dice to be rolled equals ½ of the robot's Self Awareness Score rounded down. The Target Number for the roll is determined by the number of Inspiration Points spent. Each point spent increases the Target Number by 2 for a single roll. The maximum Inspiration that can be spent in this way depends on the Program the robot is attempting to use or augment. This is typically 2, 3, or 4 points, although some Programs do not allow Inspired use. For a roll that is not related to a Program in any way, the GM sets this limit.

- Number of dice equals ½ of Self Awareness rounded down.
- Spend 1 Inspiration to increase TN by 2.
- The maximum Inspiration that can be spent on any single roll is set by the Program in question or by the GM.
- At least one Inspired Roll on a Program the robot doesn't possess is required before the robot can learn that Program.

Dramatic Control

- A variable number of points may be spent to take Dramatic Control of a scene

The robot may spend one or more Inspiration Points on Dramatic Control. Dramatic Control gives the player of the robot the ability to influence the scene in a manner outside of the capabilities of the robot character as normally only the GM can.

Like a movie director, the player may use this power to alter the scene in some fashion. This may include almost anything the player can think of: from new characters arriving in the scene, to the discovery of a previously unknown important prop, to some inexplicable "coincidence" that just happens to occur at that time. Basically any idea, object, character, event or fact that the player can think of can be inserted into the game via Dramatic Control. The player may even flash back to an earlier time in the story in order to set up an event in the current scene.

Dramatic Control represents a kind of amazing stroke of creative brilliance that the robot experiences. It's a moment where the robot does something that no ordinary robot would ever think of doing. A moment where pure intuition trumps calculated logic. However, in reality there is a human-being playing the character of the robot at all times so by and large the robot frequently will do things that no robot would have thought of, but the human player

did. Even the most immersive actors would have trouble thinking like a machine all of the time. So, to better simulate the sense of wonder and amazement that the robot would feel over such a shocking (and very unmachine-like) idea, the game shifts the creative brilliance to the player. Thus, by coming up with his own amazing stroke of creative brilliance to add to the game in a manner that goes beyond the typical control of a player and is more like that of a GM; the player can, in part, share in how a robot would feel having a thought that goes beyond the typical thoughts of a robot and is more like those of a human.

Example: *Seeking revenge for his earlier defeat, the thug Jacques has rounded up several friends who chase Pierre le Fort into a dead end ally. Pierre's player elects to spend some of his wisely accumulated Inspiration to take Dramatic Control of the scene. There are many things he could do with this control. He may narrate how Pierre discovers a pile of crates in the alley that he could climb to go up and over the wall. Or he could have the constabulary arrive on the scene. Or he could describe how a friend of his just happened to be out looking for him and arrived just in the nick of time. Or almost any number of other things.*

If the Dramatic Control would effect or inconvenience directly or indirectly another Sparked robot, that robot's player may spend Inspiration Points (even if it is not its turn) on Dramatic Control himself to edit the first player's narration. He cannot simply veto the first player, but can add exceptions or details that effectively alter or negate the desired effect (or which improve it if desired). This additional narration can continue from player to player (including the GM) until no one wishes to add anything further or has any more eligible Insight Dice to spend.

Example: *If Jacques were a Sparked robot run by another player (or the GM), that player (or the GM) could spend Jacques' Inspiration to alter the scene himself, including editing what had been said before. In this case, Pierre's player had narrated crates that Pierre would use to climb over the wall and escape. Jacques' player could not simply declare that the crates are not there. He could, however, add that the crates are empty and weak and unable to support Pierre's bulk; and that they collapse when he attempts to climb them.*

Note that the player cannot narrate his character successfully completely actions that would ordinarily require a successful Test. He can not simply say "I successfully defeat my opponents". He could however, narrate into the scene some element that gives him an advantage at defeating the opponents. The GM should assign appropriate modifiers based on the description. As a guide, each point of Inspiration spent on Dramatic Control can be treated as a success in a separate roll used to augment a current action, if the GM deems the description appropriate..

There are a couple of guidelines to follow when using Dramatic Control. First, the player should be enhancing the spirit of swashbuckling adventure. Where possible, the dramatic control should make the scene more exciting not less. Second, the GM decides based on the players description how many Inspiration Points the Dramatic Control costs.

Minor Effect (1 point):

The player inserts something into the scene that gives his character the opportunity to attempt something that otherwise wouldn't be possible. This includes the pile of crates in the above example, as well as Jacques players subsequent edit of the condition of those crates.

It also includes many of the coincidences that are staples of the genre and the insertion of various genre appropriate props. With a single Insight Point, a player may edit into existence: a convenient rope to swing from, a convenient drapery to cut down on a rival's head, a convenient sword hanging displayed on a wall just when one is needed, a convenient witness who just happens to have the information the robot characters need, etc. Each such single event, or "fact" introduced costs 1 Inspiration point.

Moderate Effect (3 points):

The player inserts something that changes the very nature of the scene or which deals with a current problem in a deus ex machina manner. The constables arriving on the scene in the above example would be a moderate effect. Having a net full of crates fall from a crane on top of half a dozen thugs would be another. A basic rule of thumb is that anything that merely provides the players character with an opportunity to make a roll, use a Program that they other wise couldn't, or perform a mundane action that otherwise they would not have been able to is a minor effect. Anything that actually accomplishes something significant without needing the character to roll is likely a moderate effect.

For instance declaring that the librarian knows the secret location of the brigand's hideout is a minor effect, because the robot still will need to convince the librarian to provide that information (probably with some form or Persuasion or Intimidation Test. The minor effect simply makes the roll possible (ensures the librarian actually knows), it does not make the roll succeed. If instead the player described simply finding a book that had a map to the hideout in it, this would be a moderate effect and cost 3 points because this not only makes finding the location possible, it makes it actually happen (unless the map needed a test to decode it).

Also covered in this category are any small limited pieces of knowledge that a player may know about the game from discussions with the other players that his character wouldn't know (often called "out of character" knowledge). The player wishes to engineer a "convenient" set of circumstances for his character to "discover" this knowledge and thus be able to act on what the player already knows. This can simply be handled as a moderate effect, or if the engineered situation still requires a roll, it can be a minor effect.

Major Effect (5 points):

Anything (including things that might be considered Minor or Moderate) that the player inserts that has a broad reaching effect beyond the events of the current scene is a major effect. Introducing major new characters who will exist and have agendas beyond the current scene are a major effect (throwaway characters who exist primarily for the current scene and then are never heard from again are simple minor effects). Anything that causes the GM to rewrite any significant portion of his game prep or background notes to accommodate it is likely a major effect. As a general rule, if the effect has repercussions that go beyond the current scene that could not have happened or taken effect by regular mundane actions by the robot, the effect is major. If the ramifications of the effect are pretty much contained within the scene itself or involve only a minor throwaway prop that might continue into further scenes, than the effect is likely not major.

Stretching Credibility (+1 point):
If what the player would like to do stretches the limits of credible coincidence, the GM may increase the cost by 1 point if he allows it at all. The GM should keep in mind that some degree of coincidence is a staple of the swashbuckling genre and penalize only the more egregious assertions.

Creates More Trouble (-1 point):
If what the player chooses to insert into a scene actually creates *more* problems or additional complications in the process than the GM may reduce the Inspiration cost by 1 as a reward. In general the complication should be something more troublesome than another easily defeated guard arriving on the scene. For instance: the player has an alarm sound before a clandestine mission is accomplished, or he declares a witness saw his robot's illicit activities, or he decides that the boat hired is actually full of cutthroats looking for an easy mark, etc. If the GM is suitably grateful for being handed a wonderful new player created plot hook he may award additional Inspiration.

Example: *One of the Queen's agents is preparing to board a boat to seek information on La Roche when he is accosted by a number of hired foot pads. The agent's player turns to Dramatic Control:*

The player narrating a cargo net full of crates that is conveniently positioned over the thugs is a minor effect, because dropping those crates still requires some action on the agent's part; such as a Athletics vs. Locomotion Test to cut the ropes holding the cargo, or a series of such Tests.

The player narrating that the rope holding the cargo simply breaks just as the thugs approach would be a moderate effect, because instead of needing a roll, the action was simply accomplished by fiat. This would not be a major effect, because nothing happened here that could not have happened in another way. The agent could have simply defeated the thugs in a sword fight and moved on. Defeating them in this way, therefore doesn't significantly alter the fabric of the game.

The player, knowing that the thugs were actually hired by the treacherous Count Rochelle (due to an out of character conversation with the GM) narrates that the thug's leader reveals this information so that now his character knows it as well. This is a moderate effect because the action of discovery is being narrated by fiat. If the player decides that his character still needs to roll to interrogate the leader before finding out that it may be ruled a minor effect.

If on the other hand Count Rochelle was in no way involved in the thugs' activities at all, but the player just spontaneously connected him to it with his narration, it definitely constitutes a game effecting twist (something that will be felt well beyond the parameters of the current scene) and is thus a major effect. The player further narrates that the leader possessed a letter from Count Rochelle containing his specific orders, that the player's character can now use to prove the Count's guilt, the GM may step in and charge an extra point for Stretching Credibility, especially if he feels the Count high score in Intrigue makes such an obvious slip unlikely.

If instead the player narrates that the letter from the Count was actually a forgery meant to frame the Count, who actually isn't involved at all, than the player has just made life more difficult for his agent who currently doesn't share his players knowledge and is likely to falsely accuse Rochelle. The GM may award the player with a reduced Inspiration cost as a reward for creating more trouble.

Increase a Program Level

- A number of points equal to the desired level may be spent to increase a known Program to the next level.

Sparked robots have another significant advantage over their Non-Sparked counterparts, they can self-program. This is not the same thing as sitting down and writing code the way a programmer would, rather its more akin to the manner in which a human learns by observation and trial and error. Non Sparked Robots do not learn. They will go through the motions of student and teacher as part of maintaining the illusion of the Tapestry, but their abilities will never change unless they are reprogrammed by an outside programmer.

Sparked robots have the ability to increase their Programs independently of a programmer. This requires a Memory Save and the spending of Inspiration Points. First spend 2 points of Inspiration, and then make a Memory Save as normal (see "Memory Saves" under "Saving Throws" in "Basic Mechanics") on 3d10 using the robot's Memory Attribute as TN. Decrease the TN by an amount equal to the desired level. Success on at least 1 die is enough to qualify for the level increase. Each failed die increases the Inspiration Point Cost by 2. If this additional cost is not paid, the level is not gained but the initial two points are lost. It is possible for successes from training or being taught by a Sparked teacher to be used to increase the Memory Save TN.

- Spend 2 Inspiration
- Make a Memory Save on 3d10. TN is Memory Attribute reduced by level of Program being sought.
- Success on a single die is enough to qualify for the new level.
- Each failed die on the Memory Save requires an extra 2 points of Inspiration
- Successes from training tests can be used to increase Memory Save TN.

Learn a New Program

As with increasing a Program, this is something else that only Sparked robots can do. The robot should have some justifiable opportunity to have learned the Program in question, such as observing it being performed regularly for a period of time, attempting it successfully using Inspiration Rolls, or being specifically taught by a Sparked robot.

The process is similar to gaining a level in an existing program except the Inspiration cost is always 4 and the Memory Save must be Totally Successful, or no level is gained. Since the initial level is always level 0, the Memory TN does not need to be modified. Training Successes help as usual. To attempt to learn a new Program the robot must first spend 6 Inspiration Points and then make a Memory Save. Successes from being taught may be rolled into making the Save easier at the GMs option. Other situational modifiers may make it harder.

- Spend 4 Inspiration
- Make a Memory Save on 3d10. TN is Memory Attribute reduced by level of Program being sought (which for new programs is always 0).
- Total Success (all three dice) is required to acquire the Program at level 0.
- Successes from training tests can be used to increase Memory Save TN.

Establish, Increase or change a Personality Trait

Self Awareness related Personality Traits can be changed, increased, or new ones added using Inspiration. Role Related Personality Traits cannot be altered except 1) as explained below under "Increasing Self Awareness" or 2) at the GMs option by someone possessing the Programming Program.

Traits represent feelings, beliefs, instincts, behaviors, or quirks and are rated from 1 to 3. In order to add a new Trait the robot must have established a history of behavior or the player must be able to provide an in game justification for how his robot acquired it. In order to increase a Trait to a higher rating (max of 3), the robot must have demonstrated a higher level of passion, obsession, or dedication with the Trait in question during the game.

Decreasing a Trait Level is more variable and depends highly on the nature of the Trait being decreased. GMs are encouraged to hold players to high standards of justification for reducing or eliminating Traits, especially Traits possessed at higher levels. This is not to say that such changes should be forbidden, but there ought to be a strong in game rationale for them.

- Buying a Trait level for either a new Trait or increasing an existing one, costs 1 Inspiration per number of existing Traits.
- Decreasing a Trait level costs 3 Inspiration per current level of the Trait (9, 6, or 3).

Increasing Self Awareness

Increasing Self Awareness is the primary purpose of Inspiration and the goal of most every Sparked robot. Mechanically it is the equivalent of "leveling-up" in the game, and there is something of a process involved to it.

The base cost is 2 Inspiration per level of Self Awareness being obtained. This is then doubled if the level of Self Awareness being obtained is equal to or less than the robot's Anthropoid Class. This means it is easier for lower Anthropoid Class robot's to advance to the higher levels.

Comparison of Inspiration Point Costs:

Anthropoid Class 4 robot				**Anthropoid Class 8 robot**		
Self Awareness level	**Inspiration Point Cost**	**Cumulative Cost**		**Self Awareness level**	**Inspiration Point Cost**	**Cumulative Cost**
0	0	0		**0**	0	0
1	4	4		**1**	4	4
2	8	12		**2**	8	12
3	12	24		**3**	12	24
4	16	40		**4**	16	40
5	10	50		**5**	20	60
6	12	62		**6**	24	84
7	14	76		**7**	28	112
8	16	92		**8**	32	144
9	18	110		**9**	18	162
10	20	130		**10**	20	182

The Self Awareness Increase Rate Dial:

The rate at which robots can increase their Self Awareness is something that GMs can vary for their own campaign. With the standard cost it takes 182 Inspiration to achieve the maximum fully sentient status. If one assumes that the amount of Inspiration gained from negative applications of Personality Traits will be roughly equal to the amount of Inspiration spent for other purposes, then acquiring 182 points to reach level 10 will require about 1800 d10s to be rolled. If one assumes that the average Test roll consists of 8 dice, that comes to 225 Tests. If one makes 1 Test Roll every 10 minutes of game time one can conclude it will take about 9 to10 four-hour sessions to achieve maximum Self Awareness.

It is a simple matter to adjust this rate by adjusting the Inspiration point cost. Instead of the base cost being 2 per level of Self Awareness it can be reduced to 1 per level for a faster rate of advancement. This would only take 91 Inspiration and cut the estimated sessions down to 4 to 5. If the base cost were increased to 3 per level of Self Awareness it would take 273 Inspiration to reach level 10 or roughly 14 to 15 four-hour sessions.

If the dial is cranked longer than this, it would be advisable to double all of the Inspiration Costs for other purposes as well to keep things roughly in scale. If a truly rapid increase for a very compressed story cycle is desired one can change the rule such that Inspiration can be earned on a 1 or a 2 on the d10 roll during Tests.

Applying the effects of a Self Awareness Increase:

- Self Awareness Increases by 1 point to a Maximum of 10
- Role Score automatically decreases by 1 point to a Minimum of 0
- Reduce highest level Core Program by 1 point to a Minimum of 2
- Increase individual Programs by a number of free levels equal to the new Self Awareness Score.
- Decrease any one Role related Personality Trait by 1 rating.
- Increase any one existing or new Self Awareness related Personality Trait by 1 rating

Self Awareness Increases by 1 point: There are two direct in game uses for Self Awareness that are effected. First, the number of dice used for Inspiration Rolls (1/2 Self Awareness Score rounded down) will go up at every even Self Awareness level. Second, the TN for Self Awareness Saving Throws will go up making it more likely for the robot to defy their programming in the future.

Role Score Decreases by 1 Point: Since Role + Self Awareness must always equal 10, any increase to Self Awareness results in an automatic decrease to Role. The TN for Role Saving Throws will go down making it harder for the GM to enforce a robot's programming against the player.

Reduce Highest Level Core Program by 1: Every robot has 3 Core Programs (Core Physical, Mental, and Social). At character creation these were started with a base of level 2 each, with levels equal to the robot's Role Score (usually 10 to start) distributed among them. Now, as Role decreases, those distributed levels are lost. Since every one of the robot's Programs are linked to one of these 3 Core Programs, the robot's effectiveness (in the form of number of dice rolled in a Test) declines as the Core Program levels declines. This reflects the knowledge and ability lost as the robot breaks free and ultimately deletes much of its core programming.

The highest level Core Program is the one that has its level reduced by 1. When levels are tied, the player can choose among them. The more Programs a robot has linked to the Core Program in question, the bigger the effectiveness decline will be. Since most robots will be designed such that their highest Core Program is the one with the most Programs linked to it, the larger effectiveness hits will occur early.

Increase Individual Program Levels: Every robot started with 9 or more level 0 Programs which were improved with free levels during character creation. Each time Self Awareness improves, the robot is awarded with more free levels. This represents the ability of a Spark to grow, adapt and become more effective as it learns through experience rather than be limited to strictly perform at the ability of its installed programs. \

The robot receives a number of free levels equal to the newly achieved Self Awareness level which can be freely distributed amongst all current Programs, or to start a new Program at 0 level.

This increase to the robot's effectiveness (in the form of number of dice rolled in a Test) counteracts the loss of effectiveness experienced from the reduction of the Core Program. By losing Core Program levels but gaining individual Program levels, the robot is both actually and symbolically destroying who it was and choosing who it will be. At first the loss of effectiveness outweighs the gain, but eventually (typically around Role 5, Self Awareness 5) the robot begins to become more effective than it was initially.

Decrease any one Role related Personality Trait by 1: Role related Personality Traits are those Traits programmed into the robot. At character creation, the robot received a number of levels of Personality Traits equal to its Role Score (typically 10 to start). As Role decreases, these levels are lost. Each time Role decreases, the player must choose any one of its Role related Personality Traits and decrease that Trait by 1 level. The total ratings of all Role related Personality Traits should always be exactly equal to the robot's current Role Score. This process, like losing Core Levels above, represents the ongoing deprogramming of the robot's initial role in the Tapestry.

Increase any one Self Awareness related Personality Trait by 1: As the robot proceeds down the path to free will and true sentience it acquires the ability to decide for itself how it will behave, who it will befriend, and what its likes and dislikes are. Each time Self Awareness increases, the player must either increase the rating level of any one of its existing Self Awareness Personality Traits or acquire a new Trait of the player's choice at rating 1.

Basic Mechanics

Summary

There are two types of resolution rolls in Robots & Rapiers. Tests (often referred to simply as rolls) and Saving Throws (always referred to specifically as Saving Throws). A Test roll is made any time a robot is attempting to accomplish something that the Game Master deems is not simply automatic. Saving Throws are made any time a robot is trying to resist an effect targeted against it.

Both types of rolls in Robots & Rapiers are made by rolling a pool of d10s and counting the number of successes. In a Test these successes are then used to power a variety of effects which help define the resolution of the roll. In a Saving Throw these successes are used to avoid the targeted effect.

Test Roll:

The number of d10s to be rolled equals the robot's level in the relevant Program plus the level in the Core Program (physical, mental, or social). Each die that rolls equal to or less than the Target Number is a success. The Target Number for all activities equals the robot's score in a related Attribute (Social, Primary, or Perception). At least one success is required to accomplish the action at a minimal level. Additional successes can be used to power any of the effects on the following list. Successes rolled in a Test can be cancelled out by successes from an Opposed Test.

- Accomplish Action: In every Test, one action is spent to accomplish the task. If this action involves affecting another robot, that robot must make a Saving Throw.
- Accomplish Action Faster: Use successes to reduce the amount of time it takes to complete a task.
- Roll-Over Bonus: Successes from the current roll can provide an equal number of bonus dice to a single subsequent roll.
- Persistent / Multiple Roll-Over Bonus: As for Roll-Over Bonus except the bonus dice equals ½ of the number of successes and can be used for multiple subsequent rolls, or a single subsequent roll for multiple people, or for both for 1/4 the successes.
- Apply Bonuses to Crafted Items: Persistent Roll-Over Bonuses can be used to fashion special items like fancy clothes that boost Etiquette Tests.
- Restore Reduced Attributes: Restore Attributes temporarily lost from failed Saving Throw rolls, or more permanently lost from Malfunctions.
- Increase / Decrease Difficulty of Saving Throws (Damage): If the action causes another robot to make a Saving Throw, additional successes can be used to make that Saving Throw harder.

Saving Throw:

Roll 3d10 (all Saving Throws are rolled with 3d10). Each die that rolls equal to or less than the Save Target is a success. The Save Target equals the robot's score in the Attribute being saved for, reduced or increased by various factors. These factors include successes from an opponent's Test used to make the Save harder. At least one success is required to avoid the full effect, however all 3 dice must be successful to avoid the entire effect. Each die that fails causes the temporary loss of 2 points from the targeted Attribute.

Basic Test Roll

When the robot attempts to perform a task that the GM feels is not simply automatic and needs to be rolled for (either because there is a notable chance of failure, or because the degree of actual success is important) the GM calls for the player to make a Test Roll.

A robot's capabilities are defined by its Attributes and the Programs it has loaded.

For any Test Roll, the player will roll a number of dice (d10s). For most Tests the number of dice rolled will equal the robot's level in the relevant Program plus the robot's level in the Core Program linked to that Program. There are 3 Core Programs (Physical, Mental, and Social) and each Program is linked to 1 and only 1 Core. This is indicated on the character sheet by which Core the Program is listed under. The robot's total capability (dice) in any Program is always the sum of his Program and Core levels. However, this assumes that the robot actually has the required Program.

Each die that rolls under the Target Number is a success. For all activities the Target Number will equal the score of one of the robot's Attributes, Which Attribute depends on what the actual activity is and how the Program is being used. It is up to the GM to decide based on the player's description of their activity which Attribute is most relevant. Any of the robot's Attributes can be called upon to serve as a Target Number. This is most often one of the 6 Prime Attributes, but can also be Anthropoid Class, Size, or Perception.

Any die that rolls a "1" is a potential point of Inspiration (for Sparked robots only). The player may choose to take that die as a point of Inspiration *instead* of taking it as a success. This represents some stray unrelated thought or introspective moment that pops into the robot's head and potentially serves as a step along the robots path to true sentience. Giving up a Success on a roll indicates the level of distraction from the task at hand the thought caused.

Any Test roll may be opposed by another roll in which case rolled successes are canceled before effects are applied. Instead of making an opposed roll, the GM may simply assign a Difficulty to a roll, which also serves to cancel successes exactly as if an actual opposed roll had rolled successes equal to that Difficulty. See "Opposed Rolls" below for more detail.

Standard nomenclature for Tests which will be used in subsequent examples is: Program vs. Attribute. as in "Make a Fencing roll vs. Articulation". It is an assumption of this nomenclature that the player will add the appropriate linked Core Program dice to the roll.

Core Dice Mechanic:

- Roll a number of d10s equal to Program + Core
- Target Number for each die equal a relevant Attribute.
- 1 success accomplishes the action at some minimal level
- Additional successes are used to modify future rolls, etc (see Using Successes below)
- Any die that rolls a "1" may be taken as a point of Inspiration *instead* of a success
- Opposed rolls or GM assigned Difficulty, cancels out successes before they are used.

***Example**: Juliard is an officer in the Cardinal's Guard. He has been challenged to a duel by one of the miscreants of the King's Guard. Juliard has the Fencing (rapier) Program at Level 3. Fencing is linked to Core Physical which Juliard has at level 5. Juliard will be rolling 8 d10 whenever he makes a Fencing Test during the duel.*

The Target Number for these rolls will be an appropriate Attribute as selected by the GM. When fighting, this is Articulation for attacking and Locomotion for defending. Each die that rolls the Target Number or less is a success.

1 success on an attack is enough to hit his opponent and cause that opponent to make a Saving Throw. Additional successes can be used to make the opponent's Saving Throw more difficult, described as a well placed or powerful blow. 1 success on a defense is enough to block an opponent's attack completely and

***Example**: Later that day Juliard and his lady friend Vivienne are visiting a master sword maker who shows Juliard a fine sonic rapier he could purchase. When earlier engaged in the duel, Juliard had primarily used his Articulation and Locomotion Attributes for Target Numbers. However, now he is merely evaluating the quality of the smith's weapons.*

The GM feels that either Processor or Memory would be an appropriate Attribute in this situation. Ultimately he decides that Juliard isn't so much analyzing the sword as he is comparing its features to his data base of what makes for a high quality weapon and so he will use his Memory Attribute. Since knowledge of swords is covered by the Fencing Program he will still roll 8 dice.

Note that the linked Core for Fencing is still Core Physical regardless of the Attribute used.

Beyond the Basics

Required Programming:

Often times the player will indicate which of his existing programs he is attempting to use. At other times the player will simply announce his desired course of action and it is up to the GM to decide which Program out of all of those available is most appropriate.

- If the robot has the required Program, the player may proceed to roll a number of d10 equal to the Program's Level plus linked Core as above.

- If the robot does not have the required Program, the GM must evaluate whether or not the activity falls within the scope of the robot's overall role in the Tapestry. If so the player may roll just his robot's Core level in whichever of the three Core Programs the GM deems is most suitable.

- If the robot does not have the required Program and the GM judges that the activity is not related to the robot's role (i.e. part of that standard of knowledge and ability all members of that role should have some basic competency in), then the robot simply cannot perform that activity. The player gets no dice to roll. An exception to this is Sparked robots. Sparked robots can, make an Inspiration Roll (see below) for any Program they do not possess.

Example: *Burgiss is a large gruff swashbuckler in the King's Guard. He finds himself summoned to court with three of his comrades to meet the king himself. The GM calls for an Etiquette (genteel) vs. Anthropoid Class roll to determine how well they handle themselves. Unfortunately, Burgiss does not have the required Program. The GM judges that members of the King's Guard would be expected to periodically appear at court before the king and so allows the player to roll just Burgiss's Core Social (genteel) dice.*

Later, Burgiss's close companion Alfredo finds himself stranded when his carriage's grav lift goes out. Being a graceful smooth talking courtier, Alfredo had no trouble with the recent visit to court, but he has no mechanical ability whatsoever. The GM rules that being a King's Guardsman and social butterfly doesn't convey even a modicum of ability related to the repair of Grav engines and so Alfredo's player is stuck with no dice to roll.

Inspiration Rolls:

Sparked robots are special in many ways. One of them is the ability to intuitively figure out how to do things they aren't specifically programmed for. This allows them to try an activity that they have absolutely no ability in. In fact, before a Sparked robot can learn a new program (other than those he started the game with or that a programmer subsequently installs in him) they must have attempted the program "Inspired" at least once. Inspired Rolls can also be called for by the player simply as a way to generate successes that will roll over as bonus dice to subsequent rolls (see Using Successes below). This represents having a eureka moment or sudden bit of ingenuity that helps with the task at hand. Beware, however; the GM is encouraged to interpret a failure in a Inspiration Roll used in this manner as causing the robot to freeze with indecision, prohibiting the subsequent roll altogether or raising its difficulty dramatically.

Inspiration Rolls are made exactly like any Test. The number of dice to be rolled equals ½ of the robot's Self Awareness Score rounded down. The Target Number for the roll is determined by the number of Inspiration Points spent. Each point spent increases the Target Number by 2 for a single roll. The maximum Inspiration that can be spent in this way depends on the Program the robot is attempting to use or augment. This is typically 2, 3, or 4 points, although some Programs do not allow Inspired use. For a roll that is not related to a Program in any way, the GM sets this limit.

- Number of dice equals ½ of Self Awareness rounded down.
- Spend 1 Inspiration to increase TN by 2.
- The maximum Inspiration that can be spent on any single roll is set by the Program in question or by the GM.
- At least one Inspired Roll on a Program the robot doesn't possess is required before the robot can learn that Program.

Example: *Juliard wishes to compose a poem to his love Vivienne. Unfortunately he does not have the required Program, Compose (Poetry). He does, however, have a Self Awareness score of 2, which means he could rely on his burgeoning sense of individuality and creativity to compose his own poem straight from the heart. Checking the Program table we find that Compose has a Maximum Inspiration limit of 3, but that there is a special rule for artistic endeavors which allows each point of Inspiration to increase the TN by 3 instead of 2. With a Self Awareness of only 2, Juliard will only be rolling a single die, but for 3 Inspiration he can at least set the TN to 9 so he can be nearly certain of a basic success. There's no chance of rolling any additional successes to roll into other Tests, however. Perhaps Juliard will instead want to seek out a finer word smith to compose his feelings for him.*

Example: *Stranded by the side of the road, Alfredo is going to be late for a midnight tryst, a truly intolerable course of events. Checking the Program list, the player sees that Mechanics has a Maximum Inspiration of only 2. With a Self Awareness Score of 6, the player picks up 3 dice and spends 2 Inspiration for a TN of 4. That's pretty good odds of getting at least a single success. "Well," Alfredo shrugs "father always said that 'the machine that can't be fixed with a good kick, can't be fixed'". With that the player rolls and gets his single success. The GM describes the Grav Lift sputtering to life. A single success is, however, a success by the barest of margins. The carriage is leaning drunkenly to one side, making funny noises and bumps jarringly against the road from time to time. Its working, but who can say for how long. Alfredo may be in for a lonely night.*

Exertion and Power:

All robots come equipped with super conducting capacitors that serve as batteries to provide the power they need to operate. This power is represented by Power Points. The maximum capacitance of the robot's power source was determined during character creation (see x). How many points the robot has currently depends on how many points the player has spent and how long its been since the last opportunity to recharge.

One of the uses of Power Points is when robot's exert themselves to greater than normal effort. The ability to operate beyond their normal operational parameters was built into robots intentionally. It allowed them to achieve specially arranged stunts, and also provided a safety margin in case things went awry and a guest's life was placed in danger. All robots are thus capable of drawing upon this additional capacity, but Sparked robots can do so at will for just about any desired purpose.

This effect is called Exertion and works in game terms by allowing the player to increase the TN of any single Test (or Saving Throw) by 1 (and only 1) for the cost of 1 Power Point.

- Spend 1 Power to increase TN of any Test or Saving Throw by 1.

Example: *Well, it was probably too good to be true. A short time later the Grav Lift dies again. Knowing the GM is likely to enforce the Rule of Diminishing Returns if he tries the same trick again, Alfred decides to hoof it. It's a long way yet and even at a full run his paramour isn't likely to wait long for him. The GM decides that its about a hour's walk but the girl will wait only another 15 minutes. The player knows that successes can be spent to reduce the time an activity takes (see the rule "Accomplish Actions Faster" under Using Successes below). To reduce an activity from 1 hour to 15 minutes (1/4 of the time) will take 4 additional successes. With the 1 success required for achieving basic success, Alfredo needs 5 successes to avoid missing his date. The roll will be Athletics vs. Locomotion which is Alfredo's strong suit. He'll be rolling 8 dice vs. a TN of 7. Doing some quick math the player knows that the expected result of this roll is right around 5. That's a pretty good chance, but not much of a margin. So he decides to spend a point of Power to boost the TN to 8. This makes success a little more likely, and if he rolls more successes than he needs he can always roll them into a bonus for his Seduction roll explaining how impressed the girl is with his efforts to keep their date.*

Related Programs:

One of the most common uses for additional successes in a roll is to roll them over into a subsequent roll as bonus dice. This is explained more fully in the "Using Successes" section below. However, there is one application of this that deserves special attention here, and that is the concept of Related Programs. This concept arises in situations where the GM calls for a certain Program to be used, say Command (bandits), and the player points out that while his robot doesn't have the Program in question he does have something similar that might be used instead, say Intimidation.

The idea of Related Programs uses the basic rule that successes from any roll can be rolled into a subsequent roll as bonus dice as long as there is a justification for how the first roll helps. In this case it seems perfectly reasonable that successes from an Intimidation roll would be beneficial in helping to command bandits. The player then makes his Intimidation roll and rolls any additional successes into his Command (bandits) roll. This roll accomplishes nothing by itself, it just augments the main roll. He then makes the main roll which is still Command (bandits). Since he doesn't have that Program, he gets no dice for it, but he does get the bonus dice he earned from the Intimidation roll which rolled into it.

In this way, the robot has successfully used his Intimidation Program to command the bandits. He didn't get to use the full value of program, however. When it came time to make the Command roll the only dice he got to use were a reduced number of Intimidation dice, those that didn't fail the Intimidation roll. This reduced number of dice is basically the penalty for using a related program and not having the specific one. The exact penalty will vary depending on how successful the augmenting roll is. This is interpreted as a measure of how well that particular group of bandits responded to intimidation by that particular robot at that particular time. At other times it may have been better or worse (depending on the roll).

If there is likely to be a series of rolls which rely on that related roll, the Persistent and Multiple bonuses can be applied like they can for any Test roll. The GM may also assign a Difficulty number (See Opposed Rolls below) if the related Program is more of a stretch.

Example: *Returning to Juliard in the sword maker's shop. If Juliard wishes to haggle with the merchant over the price of the blade he would use his Persuasion (Haggle) Program vs. his Processor Attribute. Juliard doesn't have Persuasion (Haggle), though he does have Persuasion (Seduce). Juliard's player has no interest in attempting to seduce the smith but he does point out that being skilled with a blade requires a level of familiarity that should include understanding how much the weapon is worth. The GM buys this, but also rules that since ostentatious spending is the norm among the social elite that he's going to assign a Difficulty of 1 to the roll since haggling is not a high priority. For this same reason, he rules that haggling would not normally be a part of the Core Social (genteel) program, though it would be for the common version, and will not give those dice for the haggling roll either.*

Juliard gets to roll 8d10 for his Fencing (rapier) Program as established above and his Target Number will be against his Memory attribute of 4. This would normally be expected to produce about 3 successes, but with 1 being cancelled by the difficulty it will take a good roll to get much of a bonus to roll over. Whatever additional successes he does achieve will be rolled as part of the ultimate Persuasion (haggling) vs. Processor roll. If the smith has any ability in this at all, its unlikely that Juliard will manage to get much of a discount.

If Juliard were haggling over an energy pike he could make a series of rolls. First he could roll Fencing (rapier) as being related to Fencing (pike). He could then roll that roll as being related to haggle.

Time generally effects tests in one of two ways. Either there isn't enough time to complete a Test at a normal speed and it must be rushed; or, more rarely, there is plenty of time to make sure a job gets done right (usually only villains have this luxury). Rushing a job requires spending extra successes on completing it faster. This prevents those successes from being used to do the job better and so reduces the overall performance of the task. See the section on Spending Successes below for rules on how this works. Use the following rules when the robot has the luxury of taking its time.

Taking Extra Time:

In the swashbuckling genre, time always seems to be of the essence and the heroes seem to always be racing against the clock. Only the villains seem to have the luxury of taking their time to make sure their carefully laid plans come to fruition. This likely has a lot to do with the heroes spending all of their free time drinking wine in taverns and pursuing romance while the stuffy, boring bureaucrats are slowly implementing their plots.

On those occasions where the players characters have the option of taking extra time to complete a tax, this is often best handled simply by making extra rolls. Each roll takes a period of time and, if its successes are rolled over into a bonus to subsequent rolls, a string of rolls will both take longer and make the ultimate task more likely to succeed from the accumulated bonus dice.

If a quick and dirty way to account for extra time is desired, use the following guidelines.

Increase time by 25%	Roll one extra die
Increase time by 50%	Roll two extra dice
Increase time by 100%	Roll three extra dice
Increase time by 200%	Roll four extra dice
Increase time by 400%	Roll five extra dice

Opposed Tests

Test Rolls can be Opposed by another robot who wishes to see the roll fail. To do this, the opposing robot simply makes a Basic Test of his own (using the same or different Attribute and Program as appropriate). The successes of the two rolls cancel and the winner is the robot who has successes remaining.

If the number of successes are tied, then neither side has technically won. However, in many situations where one party is merely trying to prevent the other from succeeding, a tie is sufficient to accomplish this.

Difficulty -- a.k.a. Unrolled Opposed Rolls:

When the Basic Test was described above it was indicated that a single success was sufficient to accomplish the task. This is essentially the same as a situation where the Test was opposed but the opposer rolled 0 successes. It can thus be said that a Basic Test is simply an Opposed Test for which the GM has decided the opposition has totally failed. By extension, instead of deciding that the opposition has failed, the GM may decide that the opposition has succeeded and by how many successes they succeeded by.

Consider. An opponent rolls 6 dice needing 5s or less. His expected result is 3 successes, with more indicating a better than average performance and less indicating worse. The GM could call for an Opposed roll, rolling 6 dice vs. a TN of 5, or he could simply decide that the opponent's performance was average and thus give the opponent an automatic 3 successes.

Taken further, what this means is that instead of needing a single success to accomplish the task, the robot actually needs more than one, in this case, 4.. Thus, as a means of increasing the difficulty for a given Basic Test, the GM can simply decide that more than 1 success is necessary based on conditions. Ruling that a robot needs 4 successes in order to accomplish his action is identical to the GM making an opposed roll that achieved 3 successes. Those 3 successes would have canceled 3 successes of the player's meaning the player would have needed a fourth success to achieve a single success margin of victory.

GMs in such a situation may either make an opposed roll to represent the difficulty of an action, or skip the opposed roll and simply require additional successes from the Basic Test. They may also do both, making an opposed roll, *and* increasing the difficulty of the roll.

Example: *The famous pirate Francois Le Requin is attempting to navigate through uncharted waters. This roll calls for Le Requin to roll 6 dice for his Navigation Program vs. his Processor Attribute of 5. This would normally require looking for at least one success. However, the GM decides that since Le Requin is attempting to navigate without benefit of accurate maps through dangerous waters that he'll assign a 3 difficulty. This is identical to the GM making a Test for the ocean as a character and coming up with 3 successes (which he's welcome to do if desired). As such, Francois will need at least 4 successes on his roll to chart the course successfully.*

Further, if Le Requin were in a race against another pirate to get to a cache of hidden treasure, he might also be simultaneously opposed in his effort to get to the treasure first by the other pirate. In that case, his roll would have to overcome both the 3 difficulty for the lack of a map and whatever successes his rival managed to roll (who undoubtedly has a perfect map). Francois better come up with some Related Programs to help augment his roll if he's to stand a chance.

Simultaneous and Sequential Rolls:

Opposed Tests may be simultaneous or sequential. In a simultaneous test both opponents are after the same outcome at the same time and the test is to see who gets it. Usually both opponents are rolling identical tests to see who gets more successes. In a simultaneous test it is usually only possible for one party to actually accomplish the attempted action.

> ***Example:*** *After a ruckus at a palace ball, Del Trevaine finds himself disarmed in a duel with Juliard. His sword landed some feet away in the middle of the room. Both Del and Juliard go for the sword simultaneously. The GM decides this is an Athletics vs. Locomotion Test and both parties roll. Whoever wins the roll (i.e. gains more successes) will get to the sword first.*

In a sequential test, one opponent is after a particular outcome and the other is either attempting to prevent that outcome or avoid its effects. The sequential test may be rolled at the same time (when both the action and the reaction are immediate as with an attack and parry) or they may be delayed in time for as long as it makes sense (as with a tracker vs. someone attempting to conceal their tracks). In the case of an extended delay, simply record the level of successes from the first roll to roll against them in the future (or wait to make the first roll until the second roll triggers the need for it). In a sequential test it is entirely possible for both parties to accomplish their attempted actions successfully. The roll then decides whose action actually gains the advantage.

> ***Example:*** *Juliard has managed to take the unfortunate Del prisoner. Earlier Del had secreted a small dagger on his person and had rolled Conceal vs. Perception to hide it. He'd received 5 successes, meaning the dagger was successfully concealed. It will not be noticed by casual inspection. Using the rules below Del converted those successes into a persistent bonus of 2. Now Juliard is searching Del and rolls Observation vs. Perception. Juliard's roll is opposed by Del's persistent bonus of 2, meaning instead of a single success, Juliard needs three. If Juliard is successful he will discover the dagger. Note that both rolls would be successful in this case. Del had successfully hidden the dagger, it just wasn't hidden well enough.*

Using Successes from Tests

As noted in the opening summary above, at least one success (net of any difficulty or opponent's successes) is required for the desired action to be accomplished. Additional successes allow the task to be completed better, faster, cheaper, or with more style. This section will endeavor to expand on exactly what this means in terms of what these additional successes can be used for.

All action in Robots & Rapiers is driven by spending successes. Generally speaking, the player rolling the success may determine how these successes are spent, and narrate the outcome of the roll in a manner consistent with this. The GM should restrict his input on the spending of such successes to ensuring that they are used in a way consistent with the narration and that the narration is consistent in scale to the number of successes. The GM should follow these same guidelines when spending successes from his own rolls.

Summary of Success Options:

- Accomplish Action: In every Test, one action is spent to accomplish the task. If this action involves affecting another robot, that robot must make a Saving Throw.
- Accomplish Action Faster: Use successes to reduce the amount of time it takes to complete a task.
- Roll-Over Bonus: Successes from the current roll can provide an equal number of bonus dice to a single subsequent roll.
- Persistent / Multiple Roll-Over Bonus: As for Roll-Over Bonus except the bonus dice equals ½ of the number of successes and can be used for multiple subsequent rolls, or a single subsequent roll for multiple people, or for 1/4 the successes, for both.
- Apply Bonuses to Crafted Items: Persistent Roll-Over Bonuses can be used to fashion special items like fancy clothes that boost Etiquette Tests.
- Restore Reduced Attributes: Restore Attributes temporarily lost from failed Saving Throw rolls, or more permanently lost from Malfunctions.
- Restore Spent Action Points: Action Points are a resource spent during Extended Conflicts. Successes can be used to regain them.
- Increase / Decrease Difficulty of Saving Throws (Damage): If the action causes another robot to make a Saving Throw, additional successes can be used to make that Saving Throw harder.

Accomplish Action:

In all Tests, the first success goes toward accomplishing the action that initiated the Test. If the roll is opposed, it can be said that it took more than a single success to actually accomplish the action, because the first several successes were canceled by the opposition.

All actions require this 1 net success. This includes actions such as: hitting someone with a sword, ramming a shoulder into a door in an attempt to bash it down, jumping up on a table, swinging on a rope, running across a room, knocking over a barrel of fish, picking a lock, dancing with a lady, giving a rousing speech, playing a harpsichord, interrogating a prisoner, commanding a squad of soldiers, charting a course across the sea or any other action the

players wish to attempt. With this success the action is accomplished at a minimal level of proficiency.

If the roll obtained only 1 success, then the action can be said to have been achieved at some minimal level of competence. This should be interpreted to mean "just good enough to not have made matters worse, but no real advantage is gained". 1 success is enough to recite a poem without looking like a fool, but no woman is going to swoon over the performance. Its enough to dance without stumbling and stepping on the lady's feet, but it isn't likely to earn you a very high slot on her next dance card. Its enough to actually strike an opponent with your sword, but isn't a very powerful blow. Its enough to ride a horse without falling off, but you're not going to get any points for style or win many races.

If the roll achieved additional successes, those can be used to increase the proficiency of the roll by spending them in one of the other ways listed. If you roll enough additional successes you can make the woman swoon, put in a virtuoso performance on the dance floor, drop an opponent with a skillful thrust through the "heart", or win the local derby.

If the action is the sort that would affect another robot (such as hitting it with a sword), then that robot is entitled to a Saving Throw to determine whether or not it avoids or resists the effect. Additional successes in the Test can be used to make the Saving Throw more difficult (make the effect more likely to take place).

Compound Actions can be easily handled by requiring a separate Accomplish Action success be spent on each one. A compound action is any action stated which actually involves attempting two or more things at the same time. Additional successes can be spent on improving any of the involved actions.

***Example**: The dashing rogue Del Trevaine makes his escape from the palace. Grabbing a decorative banner hanging from a balcony, Del's player announces that he is going to swing to the ground on the banner and land in the saddle of his horse waiting in the garden below. Making an Athletics vs. Locomotion Test, Del scores 4 successes. The GM rules that this is a Compound Action requiring 1 success for the successful swing, and a separate success to land in the saddle. Del takes the remaining 2 successes and uses the Roll-over Bonus to give himself 2 extra dice in the subsequent Riding roll to make good his escape.*

If Del had only rolled 1 success on the roll, his swing from the balcony would have been accomplished but he would not have successfully landed on the horse. Perhaps the horse had moved, or the banner was too short and Del was forced to land on the ground nearby instead.

Failing a roll does not necessarily simply mean "failure to accomplish the action". Rather it can be used to mean, "failure to accomplish the action…as planned". It also does not necessarily mean the action was attempted and simply did not succeed. Exactly how or why the action failed (and at what stage in the attempt) is left up to the GM to narrate. Players are encouraged to offer suggestions as to what form the failure took. A GM stumped for ideas may simply ask the player to describe his own failure

- As a simple rule of thumb. Successes make the situation simpler for the character. Failure makes them more complicated.

Example: *Returning to the above example, lets say Del's roll had failed to generate any successes at all. He failed to swing down from the balcony using the banner so he never got near to landing on his horse as planned. The GM narrates that the banner tore in the midst of Del's swing, depositing him unceremoniously in the middle of a garden fountain. The GM calls for Del to make a Saving Throw vs. Durability to determine if any damage was suffered in the fall.*

However, Del's player has a different idea. "What if Del never made the attempt to swing off of the balcony at all? Instead, just as he was ready to swing down to his horse, a number of guards enter the garden. One of them grabs the horse and another takes aim at me with his musket…" The GM considers this, realizes that Del's player is having too much fun dashing about the palace to want to see it end just yet, and agrees; thereby changing his original narration. In this case, the failed roll did not so much mean Del failed but rather that the situation Del was in got more complicated causing him to abort the attempt.

Note that either of the above possible narrations (as well as many others) are perfectly valid interpretations of failure.

Accomplish Action Faster:

Time is a fairly abstract and fluid thing in Robot's and Rapiers. Activities are generally assumed to take however long it is dramatically interesting for them to take. GMs should feel free to judge how long a particular action takes based on their own understanding of the action involved, with allowances for what would make for an interesting story. If, in order to maintain the drama of the moment, the characters need to build a boat overnight to pursue their foe, then it takes a night to build the boat. If the characters commission a boat builder to build them a boat and it would be more dramatic to have it take a week because then the characters would be behind schedule and forced to hurry, then it takes a week. Like the genre it emulates, the game does not attempt to track specific movement rates over precise distances or each action taken over the ticks of the clock.

That said, racing the clock is often a feature of the genre, and time (or the lack there of) should be freely wielded as a prod and goad to action. Swashbuckling contains a lot of "act first, ask questions later" adventure largely because time seems to be always of the essence. GMs should use their manipulation of time to keep the action coming bang, bang, bang. Lulls and down time should be avoided except as a way to further punctuate and highlight frenetic activity.

For any Test where the amount of time to complete it is important, additional successes can be used to finish the action quicker than normal. How much quicker, or how many extra successes are required is best handled through dramatic abstraction, but the following guidelines may be useful

Complete action in ¾ the time	1 extra success
Complete action in ½ the time	2 extra successes
Complete action in ¼ the time	4 extra successes
Complete action in 1/8 the time	8 extra successes

Roll-Over Bonus:

In many cases the importance of a Test lies not so much in the outcome of the Test itself, but in the advantage that Test can provide to a future test.

- Rolling successes over into bonus dice on subsequent rolls is the most common and potentially most important way in which successes can be spent in the game.

This is the driving force behind the mechanics system. In duels it allows combatants to string together a series of maneuvers, feints, and feats of derring-do. The successes of one action roll over into a bonus for the next until finally the combatant has built up a large enough die pool to go for the finishing blow. It allows the game to function without lists of environmental and situational modifiers by allowing players to generate their own modifiers by making appropriate rolls and using the successes as a bonus for another roll.

This system can be either player driven or GM driven. If the player wishes to perform some complicated action, the GM may (instead of using the Compound Action rules above) break the action down into a series of related Tests determining which Attribute and Program combination to require the player to roll for each. Alternatively, the player may have in mind some activity that he wishes to perform, but desires to increase the odds of a successful performance. He is free to come up with any combination of Attribute and Program which he can convince the GM reasonably relates to his desired activity. In either case, one of the principle uses of successes from the Test (after the required "Accomplish Action" success) is to provide bonus dice to the next Test.

Example: *Del Trevaine is at a Grand Ball. The same ball that will shortly lead to all of the trouble in previous examples. He wishes to charm the beautiful lady Vivienne into dancing with him, but decides first that he needs an edge. The player calls for an Intrigue vs. Memory Test to see if he can remember something about her that would help. The GM decides this would be a little bit more difficult than normal because Del is trying to recall a rather obscure piece of trivia; but not terribly so because the lady is fairly prominent socially. So he requires 1 additional success describing it as a Compound Action: 1 success to remember something about Vivienne, and 1 to have it be useful to the current situation.*

Del rolls 3 successes. The GM has nothing specific in mind for Vivienne at this point, so, with a moment's collaboration he and Del's player decide that Del has remembered that the lady is very vain about her shoes (the GM jots down this newly discovered Personality Trait of Vivienne's for future reference). Armed with this knowledge the rake remarks on how lovely Vivienne's shoes are and how much better they'd look out on the dance floor.

Del's player takes the 1 remaining success and rolls it over into a single die bonus for a Cajole vs. Anthropoid roll. He manages 4 successes on this roll. The first success is used to Accomplish the Action, meaning Vivienne accepts his invitation to dance. The GM could have had Vivienne oppose this roll which would have made it more difficult, but he had already decided that Vivienne was looking to make her lover Juliard jealous this evening (a fact which eventually would lead to trouble) and wasn't going to put up much resistance. Dell takes the other 3 successes and rolls 1 over as a bonus to the upcoming Locomotion vs. Dancing Test deciding that Vivienne was impressed enough with his charm to not judge his dancing too harshly. He rolls the other two into a +1 Persistent Bonus representing favorable reaction from Vivienne towards him in the future.

The roll-over bonus does not need to be applied immediately. It is possible for successes earned to be carried over to a future roll that won't be made for some time. It is also possible for successes earned by one robot to be carried over into a roll made by another. There is no hard and fast limitation to this. The GM should use good judgment when deciding how long successes can be "saved" and precisely what they could then be rolled over into; and may assess penalties reducing the amount of the bonus if the fit isn't good or too much time has passed.

Example: *The players are portraying Highwaymen from Rork's Gang and are getting set to waylay a carriage traveling down the King's Highway. Some months ago one of the players characters had scouted out the area and had achieved 2 successes on an Orienteering vs. Perception Test made at that time. The player had made note of the successes which had gone unused. Now another player is getting ready to make a Strategy (Ambush) vs. Processor Test and wishes to roll those 2 successes into the attempt describing using the scouts information to help formulate his plan. The GM rules that the first character's report on the route would certainly have conveyed knowledge about the best places to attempt an ambush and that nothing significant in the area has changed since that time, and so allows the bonus. He briefly muses charging a 1 die penalty to account for the knowledge being second hand but decides that since the character was a skilled scout who was looking specifically for information useful to the gang that no penalty was necessary. If the scout had been an amateur or had instead been scouting for good places to have a picnic the GM would likely have charged the die.*

Example: *Earlier we saw Burgiss and his comrades summoned to meet the king, with Burgiss able to use only his Core Social (genteel) dice because he lacked an appropriate Etiquette Program. Seeing his friend nervous and self conscious, Alfredo, who has manners to spare, offers some tips and advice to Burgiss. The GM calls for an Etiquette vs. Processor roll which given the last minute circumstances he assigns a Difficulty of 2. Alfredo's player rolls and gets 6 successes. 1 of those accomplishes the action, leaving 3 dice after accounting for the Difficulty. The player rolls 2 of the 3 into a +1 Persistent bonus for Burgiss which will last for all Etiquette rolls Burgiss makes for the duration of the visit. The last one, gives an additional 1 time +1 (because there aren't enough successes to make a Persistent +2 see Persistent Bonuses below) to Burgiss which he will use for the first Etiquette roll he makes in front of the king. Thanks to his friend's tips, Burgiss is feeling a little more confident.*

If multiple rolls are part of the same overall process, they can be made either in series or parallel as the GM deems fit based on how the player has narrated them and what makes sense. If in series: the first roll modifies the second, the second roll modifies the third, the third modifies the fourth and so on. If in parallel: the first, second, and third roll are made separately and the result of each of those is added together to modify the fourth. Parallel rolls are obviously much more effective at boosting the number of dice available for the fourth roll, and GMs should keep this in mind when making the distinction.

Persistent / Multiple Roll-Over Bonus:

The basic roll-over bonus as described above is useful in a variety of situations. However, all such bonuses are good only for a single roll made by a single robot. It is also possible to extend the basic bonus so that it applies to multiple robots or to multiple rolls (or both).

- Persistent Bonus: equals successes x ½. The bonus lasts beyond a single roll and may be used again by an individual robot as often and for as long as makes sense given the Test that rolled it and the narrative description given.
- Multiple Bonus: equals successes x ½. The bonus only lasts for a single roll, but many robots can benefit from it for that one roll. The number of robots who can add this same bonus to their roll depends on the Test that rolled it and the narrative description given.
- Multiple Persistent Bonus: equals successes x ¼. The bonus can be used repeatedly by many robots. Again the number and the duration depends on the nature of the Test and narration given.

Generally these modifiers are rounded down. However, in the case of Multiple Bonuses, the player can allocate mix of higher and lower modifiers among the affected parties. For instance: if there are 4 robots destined to receive the bonus, and the bonus is 3 successes x ½, then instead of assigning all 4 robots a +1.5 bonus (rounded down to 1) the player may assign 2 robots a +1 and 2 robots a +2 bonus which then averages out to +1.5.

Example: *The fearsome pirate Francois Le Requin's ship has been engaged and boarded by a warship from La Roche. Enemy troops are storming the quarter deck and Le Requin's player calls for a Strategy (Boarding Parties) vs. Processor Test and rolls 5 successes. As always, 1 of those successes is spent to "Accomplish the Action", leaving 4 to provide bonuses.*

Le Requin could keep the entire bonus for himself, pick a target among the enemy troops and roll the whole 4 dice bonus into a single pistol shot.

Or he might take a position at the top of the stairs leading to the deck where he will best be able to fend off attackers. After applying the x ½ Persistent modifier he could take a 2 die bonus to all defense rolls made while holding this position. Alternatively he could take a 1 die bonus to all defense rolls and a 1 die bonus to all attack rolls or some other combination..

Or he might rally some nearby crew and call for a counter charge. Applying the x ½ Multiple modifier he could give a 2 die bonus to the next attack roll made by the knot of crewman he leads in the charge. Alternatively he could give a 1 die bonus to the first 2 attack rolls, or a 1 die bonus to the first attack roll, and a 1 die bonus to damage (reducing the Saving Throw of any opponents struck).

Or he might rally some crew to form a defensive square to repulse the enemy soldiers. Applying the x ¼ modifier he could give a 1 die bonus to all defense rolls, or to all attack rolls, or a +1 bonus to Saving Throws (which, since Saving Throws work a little differently, would be applied by increasing the Target Number not adding a die).

It is in this way that *all* situational or environmental modifiers are handled. If a crewman climbs into a crow's nest to get a better view, make an Observation vs. Perception Test and apply the successes as a persistent bonus. That bonus determines how useful that crow's nest was to that particular robot at that particular time without needing a table of modifiers to

determine the effects of haze, or sun glare, or miscellaneous distractions. The GM and or player is free to take a poor roll and describe its poorness as being because a fog just rolled in, or the robot was "drunk" from too much carousing, or he was simply unlucky and not looking in the right direction at the right time. Basically any explanation that doesn't violate something already established and is agreeable to all concerned is acceptable.

Similarly, a player might roll Athletics vs. Locomotion to gain a Persistent bonus to Fencing rolls by gaining a height advantage from leaping onto a table in the middle of a bar fight. Whatever the roll provides is the advantage that table provides that robot at that time in that fight. A low roll may indicate a rickety or slippery table, or one that is awkwardly positioned, or one that is being constantly jostled.

Other Examples:

- Use a bonus from a Strategy (Dueling) vs. Locomotion Test to maneuver so the sun is in your opponent's eyes.
- Use a persistent bonus from an Etiquette vs. Anthropoid Test to give another robot an ongoing favorable impression of you.
- Use an Orienteering vs. Processor Test to give yourself a persistent bonus to follow a wilderness trail you'll be frequently using in the future.
- Use a multiple persistent bonus from an Entertainment vs. Articulation Test to provide an ongoing bonus to the members of your audience for the remainder of the evening.

The Rule of Diminishing Returns:

One power the GM has regarding roll over bonuses (single, persistent, or multiple), is the Rule of Diminishing Returns. If confronted with a player who uses the exact same Program vs. Attribute Test in largely the same fashion to generate bonuses over and over, the GM may feel free to hit that player with a cumulative +1 Difficulty on each successive use.

The justification for this is that repetition breeds familiarity. After seeing the same trick move several times in quick succession, your opponents aren't going to fall for it quite as easily the next time. The true reason for the rule is to enforce the genre. Gaining bonus dice from swinging on a chandelier instead of simply announcing "I attack" supports the genre. Swinging a dozen times from the chandelier simply because its an easy way gaining bonus dice, does not.

What is the dividing line between great stunt, signature move, and repetitious act deserving of a penalty? Simply boredom. If the player's dramatic description of swinging across the room from the chandelier, and then back across, and then back across, and then back across again with an enemy falling at each pass is fun and exciting…great! no penalty needed. The minute the audience begins to yawn is the minute the penalty begins to accrue.

Note: this rule also applies to GMs controlling characters as well.

Apply Bonuses to Crafted Items:

Astute readers likely already have seen the potential of Persistent Bonuses as a way of applying special traits to crafted items. This is a powerful and flexible game tool, but one that should not be abused. The GM should stand prepared to amply apply the Rule of Diminishing returns if necessary.

The actual act of crafting the item is accomplished with a single success to "Accomplish the Action" just as for any Test. This creates an item with all of the useful characteristics of a standard item of that type; assuming the time, material, and tools were available. If it's a particularly complicated or delicate item to craft, the GM may assign a difficulty to the attempt.

Roll-over Bonuses can then be used for 1 time only effects. Persistent Roll-over Bonus can be used for traits that reflect an ongoing advantage. Multiple Roll-over and Multiple Persistent Roll-over Bonus can be used for items that effect a number of people (like an armored carriage protecting all passengers). Note that these bonuses have to have a specific roll or type of roll they are tied to. As a guide, the more frequently a bonus is likely to be applied the more narrowly the roll should be defined. A bonus that is rarely used or not particularly useful can apply to a broader range of rolls.

Example: *In an earlier example we met Juliard admiring a fine blade he was considering purchasing. How was this blade crafted? The sword maker made a Craft (sword smith) vs. Articulation Test and achieved 9 successes on the roll. The first success crafts the blade. After applying the ½ modifier, the remaining 8 successes allow for +4 in persistent bonuses. Another customer might have been satisfied with applying some or all of those bonuses to appearance, making a beautifully ornamented sword that could provide its bonus to impress others by its ostentation. Juliard, however, is a master duelist. He demands a functional blade. Since fencing in a swashbuckling tale is both common and highly useful, the bonus rolls need be narrowly defined. A +4 to all Fencing rolls is deemed too broad and so the GM applies a +2 bonus to all Fencing attack rolls, a +1 bonus to all Fencing defense rolls made with the weapon, and a +1 to damage. A fine blade it is indeed.*

Example: *Leaving the sword smiths, Juliard capitulates to Vivienne who desires to stop at the dress shop where she gets fitted for a new gown. The seamstress rolls 3 successes. The first success creates a fine wearable dress. The other 2 are used to apply a +1 Persistent modifier to the gown representing its fashioning and style. While this is clearly a nice advantage to have, its application is more subtle and more limited in application than the sword's bonus above. The GM allows it to be broadly defined as affecting any roll where impressing someone with ones appearance is important. At a "mere" +1, Vivienne finds the dress simply adequate.*

As with regular roll-over bonuses, the bonuses can be applied in series or parallel fashion. If there is a modifier involved to the successes of multiple rolls in a parallel process, and those rolls are made in quick succession, the successes from each are added together prior to applying the modifier. If the rolls are separated by time or space, the modifier is applied to each separately before being added together.

Example: *The sword smith from the above example prepared in advance before attempting to fashion his blade. The GM rules that this is a parallel process and so the successes from each are rolled into the final Craft (sword smith) vs. Articulation Test. First he used a Persuade (haggle) vs. Processor Test to seek out, locate, and bargain for the very best quality components he could find and received 1 success. Second he used his vast experience to carefully plan each step of the process with precision, making a Strategy (Craft Design) vs. Memory Test and received 2 successes. Third he made a Machinery vs. Articulation Test to assemble the delicate sonic vibrating mechanism. He received 3 successes. Fourth, his forge is well stocked and tended and quite advanced and has 2 points of persistent bonuses itself which it grants to Craft (sword smith) rolls. From these 4 items the smith earned 8 extra dice to roll over into the final test of actually fashioning the sword in which he will assemble the components and attune the harmonics of the blade. It was these bonus dice that enabled him to get a solid 9 successes and craft such an outstanding weapon.*

Special Effects: Aside from simply using the successes to roll over as bonuses, they can be used to purchase special effects that don't translate directly into bonuses. Since the variety of possibilities is endless the GM should use good judgment when deciding the cost for such effects.

An effect that provides a relatively minor advantage: 1 die cost. Examples:

- A suit of clothes with a small secret pocket where things can be hidden to avoid detection. 1 die to build the capability (2 dice for larger capacity). Additional dice to increase the difficulty to detect.
- A hat feather with a knife edge hidden in the quill. A sword concealed in a cane. 1 die to build the capability. Additional dice to increase the difficulty to detect. Additional dice to provide standard weapon bonuses.

An effect that is somewhat unusual or has a noticeable advantage: 2 dice cost. Examples:

- A ring with a dose of poison concealed in the stone. 2 dice to build the capability. Additional dice to increase the difficulty to detect.
- Special pistol ammunition that sets fire to targets it hits. 2 dice per round. Additional dice to increase damage or ignition chance.

An effect that is very unusual and likely of considerable advantage: 3+ dice cost. Examples:

- A carriage that is self propelled without the need to be hitched to a robotic horse. 3 dice for the capability. Additional dice to improve Locomotion score.
- A mirror that digitally records the images it sees and can play them back. 4 dice for the capability. Additional dice to improve Memory or Perception scores.

Note that these rules follow the standard conventions already established. 1 die to "accomplish the action" and create the advantage. A higher cost representing difficulty the GM assigns to the attempt. Additional dice can then be spent on an assortment or roll-over, persistent, or multiple bonuses.

Restore Reduced Attributes:

Attribute reduction comes from 2 sources: failed Saving Throws and Malfunctions .

Temporary Attribute Losses come from failed Saving Throws. This is explained in greater detail in the section on Saving Throws, but here it is sufficient to know that Saving Throws are made to avoid effects targeted at the robot which will reduce specific Attributes; and that each die that fails in the roll results in losing 2 point of that Attribute.

These losses are completely temporary. A Perception loss might be due to being distracted or having ones head covered with a cape. A Locomotion loss might be due to being knocked off balance. Recovering from these losses is as simple as tearing the cloak away, or regaining one's footing. This is accomplished by spending successes from Tests..

- 1 success will restore 1 point of Temporary Attribute Loss.

If there is no specific difficulty that would make recovering from the loss more difficult, it generally does not matter what the nature of the actual Test in question is. Successes from most any Test can be spent from to restore the Attribute. Robots may spend the first couple successes of a Fencing roll to negate a Locomotion loss or of a Oratory roll to remove the cape and restore Perception. The effect here represents the loss of effectiveness in the Fencing roll while the combatant regains balance; or having a speech's timing thrown off while grabbing for the cape.

Additionally, the GM may either require the Attribute losses to be bought off as soon as possible, or may instead allow robots to make several rolls without doing so (continuing to suffer from the Attribute loss). This depends on the nature of the description that was used to justify the Attribute loss, and the narration of the acting player. Since possibilities are endless, no precise rules for this can be given. For instance, if a player narrates his off balance robot is making a leaping attack at a foe, a GM may well require the Locomotion loss for being off balance to be recovered from first because it obviously would effect his ability to jump. If the player instead narrates shouting an order to his crew, the GM may allow that Test unhampered with the robot remaining unsteady on his feet (of course, the player may elect to buy off the loss during this roll anyway resulting in a less effective Command test due to his distraction). Similarly the player may simply narrate a desperate attack while stumbling backwards. The GM may then allow the attack without requiring the loss be bought off.

Persistent Attribute Losses: At other times, there may be additional factors to consider that restrict a player's ability to buy off an Attribute loss. For instance, if the robot with a cloak covering his head is currently a prisoner and has his hands bound, getting the cloak off (thereby restoring Perception) becomes more difficult. If the off balance robot is trying to recover on the wet pitching deck of a storm tossed ship, then regaining footing might not be so easy.

In these cases the GM is free to require a special recovery test, instead of allowing successes from any roll be used. There may be prerequisites to the roll that must be met

before it can be made. Almost certainly this roll will be opposed (either with an actual roll, or a set level of difficulty).

At these times the 2 points of Temporary Attribute Loss might instead be better reflected by a 1 point Persistent Attribute Loss using the x1/2 modifier from the Persistent Modifiers rule. In this way the Attribute Loss cannot simply be restored with successes but continues to last for as long as the conditions which caused the loss to be persistent lasts. Those conditions must end before successes can be used to restore the losses.

Malfunctions come from failed Durability Attribute Saving Throws. Instead of causing a Temporary loss in the Durability Attribute, Saving Throws to Durability (representing actual physical damage) cause Malfunctions. Malfunctions take one of three forms 1) permanently decreased Attributes, 2) Loss of Program Levels, 3) damaged Accessories. This is described in further detail in the section on Saving Throws.

Attribute losses from Malfunctions can be restored very similarly to Temporary Attribute Losses above, except the required roll is always Robot Repair vs. Articulation. There are always prerequisites required; those being possessing the proper tools, parts, and time necessary to do the job. Often the only robot available with the right Program, tools, and parts will be in the employ of one of Auvernais' factions which makes having favor with the right parties particularly important.

Restore Spent Action Points:

Action Points are a resource spent during Extended Conflicts. Successes from Tests can be spent to regain them. Action Points cannot be increased beyond their maximum level, which equals Processor + Self Awareness. Note, this is conceptually identical to using successes to restore Temporary Attribute losses.

This can be interpreted as making an effort to regain ones focus, both mentally and physically. Instead of pursuing an advantage to an overextended state, the robot backs off (uses successes on this option instead of other more aggressive options).

- Increase Action Points: Each success used can increase the robot's current Action Point total by one.

Increasing / Decreasing the Difficulty of Saving Throws:

Successes in a Test can be used to make subsequent Saving Throw rolls easier or more difficult to pass. Since one of the more common Saving Throws is a Durability Save against physical injury, using successes in this way is frequently referred to as "damage". Increasing the difficulty of the Save is thus "increasing the damage" from the roll resulting in the Save. Decreasing the difficulty of the Save can similarly be seen as "decreasing the damage".

Unlike Roll-over Bonuses, increasing or decreasing the difficulty of a Saving Throw does not result in rolling additional dice.

- Each success spent on Increasing the difficulty of a Saving Throw, reduces the Target Number of that Saving Throw by 1.
- Each success spent on Decreasing the difficulty of a Saving Throw, increases the Target Number of that Saving Throw by 1.

The rules for Persistent and Multiple Roll-over bonuses, as well as applying the bonuses to crafted items apply identically in most cases to modifying Saving Throws.

Example: *Francois Le Requin has fought off the La Roche troops attempting to board his ship and wishes to take the fight to the enemy. He plans on swinging on a rope from the main mast over to the enemy ship and knock the enemy captain into the sea by colliding with him. First he makes am Athletics vs. Locomotion Test and scores 3 successes. 1 success means the swing was accomplished. The other 2 are rolled into the next roll which is an Athletics vs. Size Test which is opposed by a Athletics vs. Locomotion Test by the La Roche captain.*

Le Requin scores 4 net successes in the opposed Test. The first success accomplishes the action, meaning he managed to careen into the captain and make contact. This will trigger the captain making a Locomotion Saving Throw to stay on his feet. Francois uses the other 3 successes to reduce the captain's Target Number for that save by 3, reflecting how powerful the collision was and how much harder it will be for the captain as a result.

Example: *A short while later Francois finds himself alone on the enemy ship. A pair of La Roche soldiers are leveling muskets at him. They take the time to aim by making a Fire Arms (musket) vs. Perception Test. Thinking fast, Francois decides to dodge behind the ship's wheel to use it as cover. He makes an Athletics vs. Locomotion Test and gets 2 successes. The first success puts Francois behind the relative safety of the wheel. The second success is spent to increase his Durability Target Number by 1. This increase reflects the added protection provided by the cover of the ship's wheel. In this case the cover isn't much because its only a single point and since Francois did not (could not) make it Persistent it will only help with one of the two soldiers' shots. This is another example of using the bonus system to generate environmental and situational modifiers without needing a table listing how much cover is provided by a ship's wheel. The GM may set certain maximums to items of cover used in this way (the most protection the object could provide) if desired.*

Saving Throws

During the game, robots will attempt to attack, manipulate, trick, push around, or otherwise inconvenience other robots. This is handled by targeting the Attribute of the other robot most likely to resist the attempt. For instance: tripping likely is resisted by Locomotion, while grappling is likely resisted by Force.

A Test is required and the first success of that Test (as always) is spent on "Accomplishing the Action". If this action is one that can be described as attempting to affect another robot it automatically triggers a Saving Throw. For instance: "I'm going to trip him up by sweeping his legs". One success is sufficient for the sweep attempt to happen, a Saving Throw is now called for to determine if the target actually fell, merely stumbled off balance, or was completely unaffected. In certain conditions, the GM can simply call for a robot to make a Saving Throw based on its own actions.

If a player wishes to help ensure that his attempt to affect a robot (or other object) is more likely to succeed, he may spend additional successes to increase the difficulty (by decreasing the Target Number) of the Save. A player may also spend successes to decrease the difficulty (by increasing the Target Number) of his own robot's Saving Throws in preparation for an enemy's attempt to inconvenience him. Both options are described above in the section on Using Successes.

All Saving Throws (no exceptions, ever) are made by rolling 3 dice and only 3 dice. The Target Number for the roll is an appropriately targeted Attribute. Since Saving Throws can also be used for other inanimate objects (or for groups, like a mob), the GM may assign an appropriate Attribute as needed.

This Target Number can be increased (made easier) or decreased (made harder) as noted above. The GM may also increase or decrease the Target Number. Usually this is to increase a Target Number (make easier) if the described effect seems particularly ineffective or far fetched. If the Target Number is reduced below 0, the Saving Throw becomes impossible and is an automatic Total Failure.

Making the Saving Throw Roll:

- Roll 3d10 and only 3d10
- Each die that rolls equal to or less than the Target Number is a success
- Target Number is the rolling robot's score in the targeted Attribute as increased or decreased by spent successes

Three possible outcomes of a Saving Throw Roll:

- No Effect: All Saving Throw dice succeeded, the hoped for effect that triggered the roll does not take place.
- Partial Failure: Each die that failed causes the Temporary Loss of 2 points in the targeted Attribute or 1 Action Point
- Total Failure: In addition to the effects of a Partial Failure, the robot also suffers the full declared effect of the Test that triggered the roll.

Outcomes of a Saving Throw Roll

A Saving Throw is always a potential negative for the robot who rolls it. It represents a last attempt to avoid a negative outcome. The first attempt was hoping the enemy rolled poorly and failed to accomplish their task. The second attempt was to increase the likely hood of that happening by opposing the enemy and trying to stop him before he got the chance to carry out his intention. The Saving Throw is the third and final attempt.

No Effect:

If the robot succeeds on all 3 Saving Throw dice, then whatever negative effect his opponent had planned has been avoided. This does not cancel out the fact that the effects occurred, however. The success to "Accomplish the Action" has already been spent. The sword swing landed, it just did no damage. The sweep was executed, but the target nimbly avoided it. The con artist gave his pitch, but the target didn't fall for it.

As with a failure for a Test, the GM and players are encouraged to narrate an explanation for the event more dramatically interesting than simply "it failed".

Partial Failure:

Partial Failure on the Saving Throw means partial success for the enemy whose Test triggered the Save. Partial Failure occurs when either 1 or 2 of the 3 Saving Throw dice failed. The good news is that the worst of the intended effect was avoided. The robot didn't fall, wasn't shutdown (killed) by the blow, didn't agree to whatever nefarious deal the con man was plotting.

However, the robot did not emerge completely unscathed. He suffers a temporary attribute loss of 2 points for each die that failed. What this loss represents depends on both which Attribute was targeted and the nature of the narration leading to the intended effect. In the case of being tripped, the typical target is Locomotion, and while the robot didn't fall, he did stumble a bit, leaving him off balance (at a lower Locomotion). In the case of being conned, the typical target is Processor, and while the robot didn't take the bait, hook, line and sinker, he also didn't completely see through the con, leaving him slightly bamboozled and more susceptible to additional persuasion (a lower Processor). In the case of being hit by a sword, the typical target is Durability, and while the robot wasn't put down by the blow, he will have suffered a Malfunction of some kind (Durability is handled slightly differently than other attributes with regards to losses).

If a robot is taken down to zero (or negative) Attribute by such losses, the robot is unable to make any Test rolls using that attribute until it is restored to at least an effective score of 1.

Alternatively, during extended conflicts, instead of losing 2 points of the Attribute being tested, the robot may lose 1 Action Point. See Action Point saves below.

Total Failure:

With a total failure on the Saving Throw, the gig is up for the saving robot. Whatever nefarious outcome his opponent had desired to inflict him with takes effect in full. Total Failure occurs when all 3 dice of the Saving Throw roll fail to beat the target number. This happens most frequently when: the robot has a particular weakness (a very low Attribute) that is being exploited; or when the roll leading to the save results in numerous successes which are used to reduce the Target Number.

In addition to the effect itself, the robot also suffers a temporary attribute lose of 6 points (2 points per die that failed as with Partial Failure). The tripped robot is down on the ground (or perhaps pushed over an obstacle). The conned robot is fully willing to believe whatever the con man was saying. The robot struck by the sword was struck so severely or in such a sensitive location that the automatic safeguards kicked in and he was shut down (incapacitated).

Depending on the narration of the events, the GM may decide that an additional follow up Saving Throw is required. In the case of a robot pushed off of the side of a cliff, the GM would likely feel that a Durability Save is in order to determine the effect of hitting the bottom.

***Example**: Above we saw Francois Le Requin swing from his ship to the enemy vessel and cannonball into the La Roche captain. Francois won the opposed roll with 4 successes, 1 of which accomplished the action of swinging into the captain and triggering a Locomotion Save. The other 3 were spent to make the save more difficult. The captain has a Locomotion Attribute of 6 which is reduced by 3 to 3 by the additional successes. The captain must then roll 3d10 against the reduced Target Number of 3 for his Save.*

If the captain rolls total success (only about a 3% chance), than Francois's plan to knock the captain overboard has completely failed. The earlier Test already established with the 1 success to "Accomplish Action" that Francois had, in fact, managed to collide with the captain. The most likely result then is the captain had braced himself and stood firm. The immovable object stood up to the irresistible force.

If the captain had been partially successful (about a 50% chance to succeed on a single die, a 12% chance to succeed on 2) than he was knocked off balanced by the collision (to the tune of 2 or 4 points of Locomotion loss) but managed to avoid going into the drink.

If the captain totally fails his roll (about a 34% chance), not only does he lose 6 points of Locomotion but he also tumbles gracelessly over the side of the ship. Barring some miraculous event (not likely unless the captain was a significant villain) he's on his way to the bottom of the sea. Being made of metal, robot's don't swim. Perhaps if the water is shallow enough, a future salvage operation will recover him.

A Discussion on the Types of Attribute Saves

Since the Saving Throw system covers any outcome that one robot would wish to inflict on another (or which the GM believes the environment wound inflict) there are a nearly infinite number of possibilities. This section will examine some of the possibilities that players may use in their games. Clearly there is ample opportunity for inventing new uses and outcomes.

Any Attribute can be the target of a Saving throw:

- **6 Basic Attributes:** Force, Durability, Locomotion, Articulation, Processor, Memory
- **2 Derived Attributes:** Perception (or individual Sensors) and Anthropoid Class
- **Role Score:** Discussed in greater detail in Chapter X.

Force Saves:

Force measures the robot's strength. Since a Saving Throw can result in Attribute Loss, Force should be targeted any time the desired effect is one that actively would limit the robot's ability to apply strength.

A Force Save might be used if the robot is being crushed underneath a heavy weight such as an avalanche, a collapsing roof, or perhaps being buried under a ship's cargo. A Force Save might be the effect of a wrestling attempt where the attacker is going for a pin.

Total Failure in these examples means the robot is helpless. He is fully pinned by the opposing wrestler, or trapped under the weight of debris or other heavy objects. Partial Failure indicates the robot's strength is flagging and giving out. In the case of an ongoing challenge the robot might make multiple saving throws losing small amounts of Force with each Partial Failure until finally defeated by reaching Force 0.

Example: *Legendary strong man Pierre Le Fort has been entertaining an audience with feats of strength when an explosion rocks the hall. As the crowd begins to make its escape, one of the supporting roof beams, weakened by the blast, begins to collapse, threatening to trap everyone inside.*

Pierre rushes to brace up the beam and makes an Athletics vs. Locomotion Test to see if he can get there in time. The GM decides that the test is slightly difficult, but not impossibly so, and so assigns a Difficulty of 1 (remember, this is essentially the same as saying the roll was opposed and the opponent managed 1 success). Pierre gets 3 successes, 2 of which (due to the difficulty) are needed to "Accomplish the Action" and get to the beam in time. With the third success, Pierre's player decides that Pierre actually got to the beam so quickly that he'll have an advantage in trying to hold it up.

The GM decides that they will handle the challenge of holding up the beam as a Saving Throw against Pierre's Force and so therefore the extra success on the Locomotion Test will go to increasing the Target Number (making the Save easier). Pierre steps underneath the cracking beam and holds in place while the others make their way out. Supporting such a weight is a challenge even for Pierre's great strength. The GM decides there is an added Difficulty of 3 on the Save. Pierre has a Force Attribute of 8, plus Heavy Duty Hydraulics which gives him a Force of 9. With the minus 3 from the difficulty, and the +1 for the extra success, Pierre is rolling 3 dice against a target number of 7.

Cont'd: *There is a 3% chance of Pierre, strong as he is, simply collapsing under the strain; and a 34% chance he'll manage to hold up the beam without any difficulty (Total Success). However that leaves a 63% chance of Partial Failure which will result in loss of Force Attribute. The GM rules that Pierre must hold the beam long enough for the audience to get out, and so this will take 3 rolls. A Total Success would indicate no further rolls are necessary. A Total Failure would indicate that Pierre is now trapped under the collapsing roof and any remaining people are trapped in the burning hall. Partial Failure means Pierre is holding, but getting weaker, such that one of the successive rolls is more likely to fail. Also note, that the +1 bonus to the TN from the Athletics roll was not a Persistent bonus. It is only good for that first Save.*

Durability Saves:

Durability measures how well put together the robot is, from the strength of its skeleton and external casing, to use of shock absorbers and redundant circuitry, to water proofing, heat resistance, and electrical insulation. Any effect that involves inflicting physical damage on a robot is made as a Durability Save; this includes all damage from fighting duels with sonic rapiers.

The Durability Save is handled a little differently than other saves. For all other saves, each failed save die results in a loss of 2 points *in that Attribute.* Not so with Durability. Durability points are *not* Temporarily Lost as the result of a Saving Throw. Instead, each failed die results in 2 rolls on the Malfunction Table. Malfunctions cause permanent (until repaired) losses, that can not be restored as easily as Temporary Losses. Further, the Malfunction could effect any Attribute or even knock systems and accessories off line.

Shut Down: Robots on Athalon were designed to provide entertainment for guests. This included actual fighting and combat in which a robot could actually be "killed". To better simulate a well placed killing blow, all robots were programmed with certain fail safe procedures that would shut them down when certain violent conditions qualifying as a "kill" were met. This also prevented the robot suffering additional damage that would need to be repaired.

Total Failure on a Durability Save represents that shut down threshold. When a robot is damaged and experiences Total Failure on a Durability Save, the robot goes down; completely incapacitated.

Shut Down Override: The automatic shutdown is something programmed into all robots, but as a part of a robot's programming it is something that Spark's can learn to override. By spending 1 Power Point any Spark can make a Self Awareness Save. Only if the robot suffers a Total Failure on *this* save (which cannot have its TN modified) does the robot shut down. With a Partial Failure, the robot suffers a -2 penalty per failed die to *all* Tests regardless of Attribute until the penalty is eliminated with successes (as with "Restoring lost Attributes"). Eliminating this penalty represents the delay while the shut down sequence is being overridden (similar to being stunned) and must be the first thing done with any rolled successes.

Reactivation: Robot's can be reactivated from this state by another robot making a Robot Repair vs. Articulation roll. This costs 1 Power (provided by the shutdown robot or another source) and takes 1 hour. Additional successes on this roll can reduce the time accordingly. A robot can recover without assistance if it has Power remaining. Every hour the robot spends 1 point of Power and rolls 1d10. If this roll is equal to or less than Durability, the robot reactivates. If not, another check can be made in an hour if there is Power remaining.

When a Sparked robot is reactivated it must make a Memory Save (see below). The robot losses 2 points of Inspiration for each failed die. With a Total Failure on this Save the robot also loses a level of Self Awareness.

Malfunction Table:

Roll 1d10	**Malfunction Result Plus roll 1d10 vs. Anthropoid Class for Cosmetic Damage**		**Roll 1d10**	**Sub System Result**
1	Force, 1 point		**1-3**	Visual Sensors
2	Durability, 1 point		**4-5**	Audio Sensors
3	Locomotion, 1 point		**6**	Tactile Sensors
4	Articulation, 1 point		**7**	Vocalization
5	Processor, 1 point		**8-10**	Energy Capacity **
6	Memory, 1 point + Memory Save		1-5	Current Point
7-8	Sub systems roll on 2nd table, 1 point		6-10	Capacitor
9-10	External Slots, randomly determine *			

* Each Malfunction to the robot's Expansion Slots effects a randomly determined slot which is different for each robot. If the slot is empty than the slot itself is damaged. If the slot has an accessory mounted in it than the accessory is damaged. A damaged accessory stops working, exactly as if it were not possessed at all. Damaged accessories can be repaired with an Robot Repair vs. Articulation Test with a difficulty determined by the accessory type. If a damaged accessory is hit a second time it is destroyed and can no longer be repaired, only replaced. If an accessory is hit a third time the slot itself is damaged. If a damaged slot is hit re-roll 2 Malfunctions.

** Each Malfunction to the Power Reserve effects either a current point or the capacitor. If a current point, reduce the robot's current Power Reserve total by 1 point. This can be regained simply through a normal recharge. If the capacitor, then the robot's maximum Power Reserve is reduced by 1. This does not effect the current score unless the current score would now be higher than the maximum. In this case, reduce the current Power to the new maximum capacity and roll for an extra Malfunction. Capacitors cannot be repaired only replaced.

Cosmetic Damage: Any time a robot suffers a Malfunction there is the chance of suffering dents, rents, scuffs, scratches, nicks and other assorted cosmetic effects. Robots with a more delicate appearance are more prone to suffering these effects, and robots who are expected to have a more delicate appearance suffer more from having them.

Therefore, whenever a Durability Save is failed, roll 1d10 for each Malfunction result. Each d10 that rolls equal to or less than the robot's Anthropoid Class causes 1 point of Cosmetic Damage (higher Anthropoid Class makes the robot more vulnerable). These points are accumulated indefinitely (recorded on the character sheet) until they are either used by the GM or repaired through successes on a robot repair roll.

Cosmetic Damage Points are used by the GM to provide added Difficulty to the robot. They can be spent by the GM exactly like successes from a roll opposed to the robot. As such they can be used in direct opposition, canceling out successes from the robot's roll, or to decrease the TN of the robot's Saves. They can also take advantage of the Persistent modifier rule converting a one time -2 to a Persistent -1.

Cosmetic Damage should most commonly be applied to social situations where the dings and dents of hard treatment serve as a detraction (like showing up to the prom with a black eye). However, they can be used in other situations too where they represent physical damage that wasn't major enough to cause an actual attribute loss but in the right circumstances can still provide difficulty. For instance the GM may use Cosmetic Damage to add Difficulty to a robot's Locomotion Test describing a small piece of bent metal hindering the full range of motion needed; or a small scratch to a visual sensor lens which obscured part of a robot's peripheral vision, adding Difficulty to a Perception Test to notice an enemy sneaking up from behind. The GM may take 4 point of Cosmetic Damage and convert them to a persistent 2 points of Difficulty for any social situation where the robot is likely to be judged on appearance.

- Roll 1d10 for each Malfunction suffered
- Each die that rolls equal or less than Anthropoid Class causes 1 point of Cosmetic Damage which lasts until spent or repaired.
- GM spends Cosmetic Damage like he would successes from an opposed roll against the robot.

Locomotion Saves:

Locomotion measures the robot's balance, movement, speed, and grace. It can be targeted for a variety of reasons. Attempting to trip a robot or knock them down is a common source of Locomotion Saves. Also situations of unstable footing, such as the deck of a pitching ship, sword fighting on a sandy beach, an icy bridge, or a battlement slippery with moss and rain, might call for a Locomotion Save to see how inconvenienced the robot is by the footing.

Poor Footing: Note that in the case of poor footing a Locomotion Save may result in losses to the Locomotion Attribute. Ordinarily, a Temporary Loss can be restored simply by paying dice to reduce the penalty. However in this case, since the cause of the bad footing is

ongoing, it makes sense to apply the x1/2 Persistent modifier. Each failed die on a Locomotion Save for poor footing then causes only a -1 loss to the Attribute, but this loss lasts until the footing conditions improve before it can be bought off with successes. A Total Success on the save means the robot is free to navigate the footing conditions without penalty until some circumstance changes which the GM feels calls for an additional Save. Total Failure, does not necessarily indicate the robot fell down, but it does suffer a -3 point persistent loss to all Locomotion rolls *and* all Articulation rolls as well.

Knockdown: Knockdown is a side effect of being damaged. Basically if a particular blow is forceful enough, the robot may be knocked down because of it. When these rules refer to damage, it means successes from an attack roll that are used to reduce the Target Number of a Durability Save. A hit with only a single success is enough to trigger a Durability Save, but with 0 damage difficulty. A hit with 4 successes that triggers a Durability Save and reduces the Save's Target Number by 3, is said to be 3 damage.

- If damage from a single piercing or thrusting attack is equal to or greater than the target's Size, the robot must make a Locomotion Save to avoid Knockdown.
- If damage from a single bashing or cutting attack is equal to or greater than ½ Size (rounded down), the robot must make a Locomotion Save to avoid Knockdown.
- The Knockdown Locomotion Save TN is reduced by the amount that damage exceeded Size or ½ Size respectively.

***Example**: Del Trevaine is carousing at the Tavern on the Green when a brawl breaks out; something to do with Del's hand, a barmaid, and an enraged older brother. The brother takes a swing at Del with a nearby bench (a Brawling Test) and catches him with a solid blow of 5 successes. The first success goes towards accomplishing the hit, the other 4 go to damage (increasing the difficulty of Del's Durability Save); the brother isn't fooling around with Del! Since this damage comes from a bashing attack and the damage difficulty of 4 is 1 point greater than ½ Del's Size of 6, Del must make a Locomotion Save at -1 to the Target Number or be knocked sprawling in addition to any damage from the blow.*

Articulation Saves:

Articulation measures the robot's manual dexterity, coordination, and general upper body agility. Anything that would reduce the robot's ability to act freely in this regard can be described as an Articulation Save.

Some of the more common uses would be entangling someone in a net, or grappling with an opponent's weapon arm, or snaring him in a lasso. However, Articulation makes a particularly good Saving Throw target for simulating special weapon moves. A robot may seek to bind his opponent's weapon, describing Articulation losses as having the opponent's blade bound up so that he can't swing effectively.

Example: *Pierre le Fort is brawling with Jacques, a common ruffian, when the scallywag, intimidated by Pierre's great size, pulls a knife. Unarmed, Pierre seeks to grapple the scoundrel's knife arm, before pinning him into submission. Pierre makes a Brawling vs. Force Test which the ruffian opposes with Brawling vs. Locomotion described as trying to slip out of Pierre's grasp. Pierre wins with a single success. This is enough to force Jacques into making an Articulation Save, but not enough to make it more difficult by reducing the Target Number.*

With a Total Failure on the Save, Pierre will have his opponent's arm fully bound up and useless for an attack. With a Partial Failure, he at least will reduce Jacques' Articulation, making it harder for Jacques to stab him. With enough Temporary Losses over the course of multiple Saving Throws Pierre can get Jacques' Articulation to 0, essentially preventing him from attacking with the knife altogether…assuming he doesn't get stabbed first.

Processor Saves:

The Processor Attribute measures the robot's ability to interpret stimuli, process data, and execute decision routines. It is, in short, the robot's ability to think. As such it can be targeted by Saving Throws dealing with outwitting, confusing, tricking, intimidating, persuading, seducing, or a host of other possibilities that represent one robot obtaining a mental, intellectual, psychological, or "emotional" edge over another.

Temporary Losses in the Processor Attribute can be described as the robot becoming confused, tying up processing cycles with iterative logic routines, or simply steadily checking through the built in programming that requires a robot to respond in a human-like fashion.

Example: *In an earlier example we saw Juliard, captain in the Cardinal's Guard, attempting to haggle with a sword smith over a fine rapier. We also saw that since Juliard did not have the Persuasion (haggle) Program it was unlikely he would be successful in getting the smith to lower his price.*

So he conjures up a good story, one filled with betrayal and Juliard's need of the blade to mete out justice. He concludes with the implied promise that the Cardinal himself would take it poorly if Juliard could not complete his quest because he lacked a good sword. Surely for such a good cause the smith could see his way to setting a price Juliard could afford. The GM rules this an Oration vs. Processor Test by Juliard (although it could just as easily have been Persuasion (trickery) depending on the approach). The smith will oppose this with Core Mental vs. Processor. The GM rules that detecting cons would be expected behavior for any merchant and so allows rolling the Core Mental Attribute.

Juliard wins this roll with 3 successes. The first is sufficient to have the smith believe this fabrication and trigger a Processor Save. The next two increase the difficulty of this save by 2. Total Failure on the Save by the smith is defined as the smith being so moved he agrees to Juliard's outrageous price. With only a Partial Failure he won't but it will at least make Juliard's subsequent Haggle Test easier (the smith will be rolling against a lower Target Number).

Memory Saves:

Memory literally measures the robot's capacity for storing data, both actual programs and collected input from sensory systems. As such it can be a somewhat difficult attribute to target with a Saving Throw.

One possible use of a Memory Save could occur if the robot is being operated on by a robot with Programming ability. If the operating robot is seeking to delete Programs or individual memories from the robot, a Memory Save could be used to determine how much is lost.

Another might pit two scholars engaged in a philosophical debate against each other with opposed Academics vs. Processor Tests. The loser of the Test has to make a Memory Save with the extra successes used for added difficulty. Total Failure on the Memory Save, or Memory reaching 0 from Attribute Loss means the robot loses the debate.

A barrister is questioning a witness to a crime. The barrister's skillful questioning forces the witness to make a Memory Save. Failure indicates the witness's story has lost all credibility due to inaccuracies in the retelling.

Memory Malfunctions are the most common use for a Memory Save. When this happens the robot must make an immediate Memory Save. On a Total Success, there is no additional effect, the point of Memory is lost and that is it. On a Partial Failure, bits of data are lost. The player should randomly determine 1 item from a list of all programs, including Core Programs. Each failed die results in a loss of 1 levels from that Program. A Total Failure indicates the entire Program is lost. When the point of Memory is repaired, a second Memory Save is permitted. Total Success indicates the robot regains full memory of that Program or Trait. Partial Failure indicates the robot permanently loses 1 point for each failed die (up to the amount originally lost) but regains the rest. Total Failure indicates the robot regains nothing of what was lost.

Shut down Memory Loss occurs whenever a Sparked robot is completely shut down. This can be the result of the fail safes being triggered from a failed Durability Save, or from the robot running out of Power, or any other occurrence that results in the robot being essentially turned off. When any of these events occur, make a Memory Save. when the robot is reactivated. Each failed die is a loss of 2 points of Inspiration. A Total Failure results in the loss of 6 points of Inspiration plus 1 full level of Self Awareness. This is essentially akin to the effect of shutting a computer down without saving and losing everything that was stored only in RAM. Only Sparked robots are effected.

Perception / Sensor Saves:

Perception Saves are among the more common Saving Throws and easily understood. Perception as an Attribute is based on Visual + Audio Sensors (or occasionally Tactile). Perception Saves can be targeted at the Perception Attribute itself or targeted at one of the three Sensor systems specifically. If targeted at one of the three individual sensor systems, use double the sensor's level to determine the default Target Number. In other words, don't use Visual + Audio for a save that clearly targets visual only. Use Visual x2 instead.

Total loss to a sensor cannot exceed the total level of that sensor. Temporary losses don't actually reduce the robot to a lower sensor level, they just cost dice. A robot dropped from Visual Level 4 to Level 2 doesn't suddenly start seeing in only black and white.

Perception and Sensor Saves can be used for a variety of purposes. Throwing mud in an opponent's "eyes" can trigger a Visual Sensor Save. Trying to eavesdrop through a heavy wooden door during a thunderstorm, or to hold a whispered conversation during a ball with music in full swing can call for an Audio Save. Similarly, environmental effects such as a thick fog or a howling wind may call for a Save prior to making a Perception Test to see how the robot is effected.

Perception Saves are often made as Action Point saves (see below) during extended conflicts where the loss for failure is 1 Action Point per failed die instead of the usual 2 Perception per failed die.

> ***Example:*** *Del Trevaine is attempting to see the lady Vivienne again. She is residing in private apartments in the village. Juliard, very jealous and highly suspicious, has ordered one of his men to stand guard outside her town house to make sure she doesn't receive any "visitors".*
>
> *Del bribes a street urchin to run up to the guard and try to sell him some trinkets. The urchin makes a Persuasion (haggle) vs. Processor Test opposed by the guard making an Observation vs. Processor Test. The urchin wins by 3, forcing the guard to make a Perception Save (Visual + Audio), and using the extra successes to make the Save more difficult. While the guard is thus suitably distracted Del makes his Sneak vs. Locomotion Test opposed by the guard's Observation vs. reduced Perception. If the Perception Save had been a Total Failure, Del could have snuck by without even needing a roll.*

Anthropoid Saves:

Anthropoid Class literally measures how human a robot appears; how much like a human the robot looks, and how much like a human the robot sounds. However, the societal structure of the Tapestry ensures that Anthropoid Class is a proxy for much more than that.

Anthropoid Class is used as a measure of social status. Those of high Class are presumed to be genteel. Those of low Class are presumed to be common. The effective Social Class of a robot may be increased (or decreased) by Favor, but this extends only as far as the patron's sphere of influence.

An Anthropoid Save may be called for if a robot is subject to ridicule, embarrassment, being publicly chastised by a superior, or any occasion that could result in a loss of face.

Example: *While he was at the grand ball, Del Trevaine found himself entertaining an appreciative audience of hanger-ons with his rapier wit. Looking on disapprovingly was Cardinal Mathineau himself. Several of Del's remarks had bordered on disrespectful and the Cardinal was not appreciative of satire, especially when it encouraged challenging the natural order of things. Noticing the Cardinal's scowl, a Del makes a reckless remark about clerical vows of chastity and the Cardinal's frequent visits to the Queen. The line was crossed.*

The GM calls for an opposed Test between Del's Oration vs. Processor for the cleverness and delivery of his remark, and the Cardinal's Oration vs. Anthropoid Class for the his outraged response. The winner wins the approval of the crowd. Mathineau is rolling his Anthropoid Class because it is his social status that is at issue here. Additionally, The GM decides that since Mathineau's class is notably higher than Del's that he has a natural advantage at swaying the opinion of courtiers. Del is given a Difficulty of 2 to overcome for his roll. Even a boor can get laughs from ridiculing a beggar, but few are willing to mock a Cardinal of the church.

Del loses the roll by a large margin and is forced to make an Anthropoid Save with a large reduction in Target Number. The result is Total Failure. Del has lost all face. He overstepped the bounds of polite society and the result was an immediate fall from favor. As the crowd drifts away, a scowling Cardinal summons a captain of his guard to escort Del out of the palace. A duty the jealous Juliard is more than happy to perform.

Regaining lost Anthropoid Class is often not an easy thing. An embarrassment among friends may be soon forgotten, but a faux pas at court may result in the offender being ostracized, even exiled from the palace. Position may be regained with a few well placed gifts, a well timed service, or the intercession of a patron; but often the road is long and arduous if it can be traveled at all.

Action Point Saves:

Action Points are introduced in the next chapter on Extended Conflicts. They are a variable resource that is used to regulate the pacing of competitive interactions such as combat, social machinations and politicking. They can be targeted by a Saving Throw and suffer loss, just like an attribute.

Action Point Saves are a little different from other Saving Throws. First, the Action Point total is not used as the Target Number for the save because it is a spendable resource which varies constantly during play. Rather, another Attribute is selected that most ties into the nature of the Conflict. In social situations, that attribute is usually Anthropoid Class. In combat it is often Locomotion. Perception Saves are common in both arenas to determine if something is noticed in time.

Second, instead of losing 2 points of the Attribute in question per die failed, the Action Point Save loses 1 Action Point per die failed, regardless of the Attribute actually tested.

Role Save and Self Awareness Save:

All robots in Auvernais operate under the compunction of their core programming. While Sparked robots are on the road to being truly free and independent, until they achieve Self Awareness 10 and Role 0, they have not completely broken away.

At any point in the game where the GM feels that a player is not behaving "in character" as defined by his Personality Traits, Role, and the dictates of the Tapestry itself, he may make a Role Save. This is a Saving Throw made like any other and represents the robot's programming resisting the robot's own efforts to violate that programming. In this way, successes on the roll are interpreted as successes of the *role* and act as limitations on the free will of the player. The player will want the Role Save to fail, indicating that the robot was able to overcome its programming in that instance and act as the player chooses.

If all dice succeed the save is a Total Success and the robot is compelled to act according to its programming (i.e. the player is compelled to act as the GM says its programming requires). If no dice succeed the save is a Total Failure. Either the programming never kicked in or the robot's own will overcame it. The robot (a.k.a. the player) is free to act as desired without further interference from its programming (a.k.a. the GM)
If only 1 die succeeds then the robot is largely free to act as desired but the GM can place a single stipulation on the behavior which the player must abide by.
If 2 dice succeeds then the robot is largely compelled to act as programmed but the player can place a single stipulation on the behavior which the GM must abide by.

At other times in the game, the players may desire to willfully violate, subvert, or ignore their core programming and they may call for a save themselves. At these times it is suggested to let the players roll their own save in the form of a Self Awareness Save. Since Self Awareness and Role combined always add up to 10 and the rolls are made on d10s, making a Saving Throw using Self Awareness as the TN and interpreting in favor of the robot is statistically identical to making a Saving Throw using Role as the TN and interpreting against the robot. Symbolically, however, the difference is that the GM making a Role Save represents the programming attempting to oppress the robot's free will; while the player making a Self Awareness Save represents the robot's free will attempting to rebel against its programming. The Self Awareness Save is made identically to the Role Save only the results are reversed.

Extended Conflict Mechanics

Summary

These mechanics expand upon the basic mechanics presented in the last chapter. For ease of understanding (through consistent references and examples) these rules are presented for use in combat (soldiery and fighting duels being a staple of the genre). However, the end of this section provides some guidelines as to how they may be adapted for any sort of robot vs. robot conflict.

Attacks are made as Tests and they may be opposed if the target defends. In a dueling environment this usually Fencing (or other fighting program) vs. Articulation opposed by Fencing vs. Locomotion. Other maneuvers and special activities that may occur during combat are also handled as Tests and can involve just about any combination of Program vs. Attribute the players can think up. These may also be opposed, either by a roll, or by a GM set difficulty as established in the previous chapter. All of the roll-over and bonus effect rules from the basic mechanics are in place during combat, and players are encouraged to describe dramatic feats of derring-do and other swashbuckling staples such as taking successes from swinging on a rope and rolling them into a bonus to attack an opponent below. The end of this chapter includes a non exhaustive list of sample maneuvers and ideas on how they might be carried out mechanically in the game.

The chief difference between the extended and basic mechanics is the addition of Turn Order and Action Points. The primary purpose of Action Points, when combined with the Turn Order list, is to provide a means of determining which robot gets to act and when (what games often refer to as an initiative system) in a way that encourages players to assemble multi-action combination moves, balancing the greater effectiveness of such moves against managing the resource of Action Points.

Combatants will receive 1 free action on their turn, but can spend Action Points to take additional turns (i.e. get additional actions) allowing them to string together various combinations and special moves. This is done in combination with the Turn Order list. The robot at the top of the list takes the next turn. When finished he's moved to the bottom of the list and the next robot (who is now at the top) takes a turn. Spending an Action Point effectively moves the spender to the top of the list. Spending several Action Points in a row will keep the combatant at the top of the list allowing him to take multiple turns in a row (until he runs out of Action Points). Action Points are also used to take an action when it is not the robot's turn (as when defending or trying to oppose someone). Running out of Action Points means the robot is open and vulnerable. When viewed in this way, Action Points serve as a gauge of the ebbing and flowing of momentum during a fight. The combatant with a significantly higher current Action Point total has sizeable advantage.

Turn Order

Turn Order is simply the order in which each robot will perform their turn. This is maintained as a list of all combatants and the order they will go in. The robot at the top of the list will take the next turn and perform one single action (usually something involving making a Test roll of some kind). Following his turn, the robot is moved to the bottom of the list and the next robot (now at the top) takes the next turn.

However Action Points can be spent by robots to move them from their current position in the Turn Order to the top of the list instead of waiting for their turn to come about normally. This also allows them to take several turns in a row by repeatedly spending points to return to the top of the list. Thus, Turn Order will be rearranged throughout the combat. This is not difficult to keep track of in a duel between 2 robots. However, in larger engagements players should use a tracking system to help them manage it. The simplest way to do this is to use cards, labeled tokens, miniatures, or other identifiable markers and arrange them in order. As each combatant takes his turn move his marker to the end of the line. As turn order changes, rearrange the markers accordingly.

Initial Turn Order:

- The initial turn order is set in order of Perception from highest to lowest.
- Break ties by comparing highest Locomotion, Current Inspiration, or die roll in that order.
- In the case of surprise (requiring Perception Saving Throws), break ties first by degree of success on the Saving Throw, then as above.

This reflects the combatants' initial awareness of when where and how a combat is about to start, giving the more aware combatants an initial head start.

For most situations, Perception equals the sum of the combatants' Visual and Audio Sensor systems. However, if the combat is taking place in the dark it should use only ½ of Visual Sensors (round up) plus Audio; or, in total darkness, Audio alone. Similarly if there is a tremendous amount of background noise that would mask hearing ordinary sounds, use only ½ of Audio (round up) or even Visual Sensors alone.

Don't forget to include the effects of Advanced Systems. Advanced Visual Sensors add allow the combatant to ignore darkness penalties. Advanced Audio Sensors allow the combatant to ignore noise penalties. Advanced Tactile Sensors allow the combatant to detect subtle vibrations in the air and ground and give a +1 to basic Perception. Advanced Olfactory Sensors allow the combatant to detect smells with the acuity of a bloodhound and give a +1 to basic Perception.

Narrative Turn Order: Sometimes it is simply obvious from the narration who is swinging first and its that parties action which initiates the combat. In this case, establish the Turn Order as normal, and then cycle through it moving each robot, in order, to the bottom of the list until the initiating robot is at the top. Combat starts there.

Example: *Consider 4 characters for the following examples: Alfredo, Burgiss, Charles, and Devon.*

- ***Alfredo*** *has: Level 4 Visual Sensors and Level 4 Audio Sensors (base Perception 8); Locomotion 7, and 5 current Inspiration.*
- ***Burgiss*** *has Level 2 Visual Sensors and Level 4 Audio Sensors (base Perception of 6); Locomotion 4, and 3 current Inspiration.*
- ***Charles*** *has Level 3 Visual and Level 3 Audio Sensors (base Perception of 6); Locomotion 6, and 3 current Inspiration.*
- ***Devon*** *has Level 4 Visual Sensors and Level 3 Audio Sensors and also the Advanced Visual system (base Perception of 8); Locomotion 5, and 7 current Inspiration.*

Example: *The four friends always seem to be scuffling about something. The latest involves which one of them is picking up the tab at Tavern on the Green. The combatants square off and swords are drawn. It is time to determine the initial Turn Order.*

- *Alfredo and Devon are tied with Perception 8, Alfredo wins the tie by having superior Locomotion.*
- *Burgiss and Charles are tied with Perception of 6, Charles wins by having superior Locomotion.*
- *The Initial Turn Order then proceeds: Alfredo, Devon, Charles, Burgiss.*

Example: *Consider the above situation taking place on a moonlit bridge where the GM decides to use ½ Visual (rounded up) to reflect the darkness penalty.*

- *Alfredo has Level 4 Visual Sensors which halved and rounded up becomes 2 for an effective Perception of 6.*
- *Burgiss has Level 2 Visual Sensors which are halved to 1 for an effective Perception of 5.*
- *Charles has Level 3 Visual Sensors which are halved to 1, for an effective Perception of 4.*
- *Devon, however, has advanced Visual Sensors which ignore darkness penalties. His effective Perception remains 8.*
- *The Initial Turn Order in this situation proceeds: Devon, Alfredo, Burgiss, Charles*

Adding Additional Combatants

Often times reinforcements may be added to a combat, or a disengaged combatant may reenter the fracas. This is easily handled by comparing the new combatant to the Turn Order *as it currently exists.* Starting with the robot at the top of the list, insert the new combatant into the order immediately in front of the first robot (in order) that he beats. Use tie breakers as described above where necessary.

Example: *After several rounds of combat and rearranging the Turn Order, the current Turn Order in our moonlit combat example looks thus: Burgiss, Alfredo, Charles, Devon. Edward now joins the fray. Edward, has Level 3 Visual and Level 4 Audio Sensors, which in darkness provide an effective Perception of 5. His Locomotion is 4 and his current Inspiration is 1.*

- *Edward first compares his Perception to Burgiss. Like Burgiss he has a Perception of 5 so he checks Locomotion which they both have at 4. However, Edward has only 1 current Inspiration so he cannot be placed in front of Burgiss.*
- *He then checks the second slot but he loses to Alfredo's Perception of 6.*
- *His Perception of 5, beats Charles' Perception of 4, so Edward slips into the third spot in the turn order bumping Charles and Devon down. down.*

Ambushes and Surprise

Ambushes and Surprise are easily handled using the Saving Throw rules. The would be ambusher makes tests as appropriate for his ambush. For a set ambush this might be successes from a Conceal vs. Processor Test rolled into a Strategy (Ambush) vs. Perception Test. For a common foot pad it might simply be a Stealth vs. Locomotion Test. The roll may be opposed by an Observation vs. Perception Test, perhaps one that had been augmented by the target specifically looking for ambushes with a Strategy (Ambush) vs. Processor Test.

Success by the ambushers causes the target to have to make a Perception Save, additional successes can be used to make this Save harder by reducing the Target Number as usual. If there are multiple targets of the ambush, be sure to use the 50% modifier for effecting multiple rolls with a single bonus. Perception Attribute losses from Partial Failure of the Save effect the initial Turn Order normally. Once set, the Turn Order does not change due to Perception being recovered.

Ambush Results:

- Ambush Fails: if the Ambusher failed their initial test, then they are noticed and the targets are not effected (no Saving Throw triggered). If the ambushers go through with the attack anyway, or are attacked by their intended targets, *they* must make Perception Saves themselves due to the advantage of the advanced warning they've given their opponent.

- Partial or Total Failure on Save: Each failed die reduces Perception by 2 for purposes of determining Initial Turn Order *and* results in a loss of 1 Action Point from the base total (see below).
    - The Attribute Loss must be bought off immediately with the robot's first roll (or rolls, if necessary). This has the effect of wasting a number of successes (typically successes needed to defend against a surprise attack) due to the delay in noticing the attacker.
    - In addition to the effects of a Partial Failure, the surprised robot *cannot buy Initiative*, until he has bought off the Perception Loss. This means he may only make opposing rolls, must wait for his turn in the Turn Order to come up normally, or must wait for the Attribute Loss to wear off.

- Special Attribute Recovery: As a special rule every time any robot buys Initiative any other robots who've been surprised recover 1 point of Perception automatically (of those lost due to surprise) at the end of that action.

[Ed Note: The Special Attribute Recovery rule is in place to prevent a robot who Totally Failed the save from being permanently surprised simply because no one attacks him, thereby he never has the opportunity to buy off the Perception Loss. It may be appropriate to apply this rule to ALL Temporary Attribute losses.]

Multiple Ambushers:

In the event of multiple ambushers, determine the applicable Attribute and Program for the final test:

- Use the *worst* Attribute and the *best* Program from among those staging the Ambush in the opposed roll vs. the targets.

Example: *Their differences reconciled, the above 4 characters are traveling together through a shady part of town at night. Six hired thugs are preparing to set an ambush for them in an alley way. The characters are expected, and the thugs have had plenty of time to prepare. Prior to the four's arrival the leader made a Familiarity (Back Alleys of Auvernais) vs. Memory Test and received 3 successes. He rolled these as bonus dice into a Conceal vs. Perception Test to establish a good hiding place and received 4 successes. Since these successes are going to be rolled into the ambush rolls of all 6 thugs as a bonus due to their preparation, the GM assigns the Multiple Bonus cost of x1/2 giving each thug +2 dice for their ambush.*

The ambush itself will involve the thugs leaping from the shadows on the unsuspecting foursome. This will be a Stealth vs. Locomotion Test. The lowest Locomotion among the Thugs is 4. The highest Stealth belongs to the leader and is 7. The ambush roll will thus be 9 dice (Stealth of 7 plus the 2 from the roll over bonus from the preparation) vs. a Target Number of 4.

Example: *The characters approach the ambush point. As noted previously, with the darkness penalty their Perception scores are Alfredo 6, Burgiss 5, Charles 4, and Devon 8.*
The ambush roll is very lucky and gets 5 successes. The 4 comrades were not suspecting any trouble and were not being especially cautious so they made no opposing Test. The first success accomplishes the action and causes Perception Saves. The remaining 4 go to reducing the Perception Save Target Number. Since there are 4 targets, the x1/2 Multiple modifier applies meaning all Perception Saves are reduced by 2.

Alfredo rolls 3 dice against a TN of 4 (Perception 6 -2). He gets no successes. He is surprised. For purposes of determining Initial Turn Order he will use a score of 0 (Perception 6 -2 per failed die). He loses 3 Action Points and cannot buy Initiative.

Burgiss rolls 3 dice against a TN of 3 (Perception 5 -2). He actually succeeds on 1 die. For purposes of determining Initial Turn Order he will use a score of 1(Perception 5 -2 per failed die). He loses 2 Action Points and cannot buy Initiative..

Charles rolls 3 dice against a TN of 2 (Perception 4 -2). He rolls poorly and fails all three dice. For purposes of determining Initial Turn Order he will use a score of -2 (Perception 4 -2 per failed die). Charles loses 3 Action Points and cannot buy Initiative.

Devon rolls 3 dice against a TN of 6 (Perception 8 -2). He rolls well and gets total success. He will use his full score of 8 for purposes of Initial Turn Order and does not lose any Action Points.

Uh-oh. Devon is the only one of the four not surprised, and he's facing 6 thugs.

Action Points

Action Points form the currency of combat. They allow combatants to perform various feats of derring-do and launch more effective attacks, by taking several turns in a row. Action Points will increase and decrease throughout a combat as momentum shifts. Often a robot will attempt to accumulate a number of points in order to spend them on a string of actions to increase the effectiveness of a finishing attack.

Base Action Points

Base Action Points are the points each combatant starts with at the beginning of a combat. They are determined by a robot's Processor and Self Awareness Attribute. Non Sparked robots do not have a Self Awareness Attribute so Sparks like those of the players will gradually gain an edge as they progress. This reflects the decreasing reliance on scripted routines and a greater ability to improvise. The Processor score determines how quickly the robot can process the initial situation that he sees, and evaluate conditions. More Action Points means it becomes "cheaper" to string together complex combinations of moves. Action Points are easy to track with an extra die next to each robot's Turn Order marker.

Base Action Points are rolled immediately following establishing Turn Order. The roll is a special Test. The number of d10 to roll equals the robot's Processor + Self Awareness Attributes. The Target Number is based on situation. Base Action Points equal 1 plus 1 for each success on this roll, reduced by surprise (to a minimum of 0).

Roll Base Action Points:

- Processor + Self Awareness vs. special Target Number as follows:
- Equal numbers on both sides: TN = 5
- Side with greater numbers: TN = 4. More than 2x numbers: TN = 3
- Side with lesser numbers: TN = 6. Less than ½ numbers: TN = 7
- Base Action Points = 1 plus 1 per success (less any lost from a Perception Save)

The Target Numbers mean that the outnumbered side gets an *advantage*. This serves 3 purposes. First, it adds a bit of realism. The side with the greater numbers also has less freedom of individual movement as they begin to interfere with each other when converging on the same target. Second, it promotes the proper swashbuckling dynamic, discouraging player robots from ganging up on opponents and making it more possible for a single heroic swordsman to hold off a number of lesser opponents by giving him the Action Point advantage he needs. Third, since the most common situation of being outnumbered will involve the player robots being out numbered by lesser mooks, having each mook receive fewer Action Points will be less for the GM to keep track of.

Example: *With the fight in the alley above about to take place the 4 comrades find themselves outnumbered 6 to 4. They will each roll for Base Action Points on Processor + Self Awareness vs. Target Number 6. The 5 Non Sparked thugs will roll Processor only vs. Target Number 4. The Sparked thug leader will roll Processor + Self Awareness vs. Target Number 4. 1 Action Point per success rolled.*

The three who are surprised will have this number reduced by 2 per failed die on the Perception Save

Using Action Points

If no Action Points are spent, play proceeds as follows: The combatant at the top of the Turn Order takes 1 Action. One Action means 1 Test Roll or an activity of equivalent scope which does not require a Test Roll. When that is complete, the robot moves to the end of the Turn Order and the next combatant takes 1 Action, and so forth. Action Points can be used for 3 purposes. 1) Making a roll out of turn to oppose another robot's Action; 2) to take the next turn and jump to the top of the Turn Order; 3) to pay an Action Point penalty from a Perception Save before being allowed to do #1 or #2.

Make Opposing Roll:

When being attacked or maneuvered against on another combatant's turn, a robot may spend 1 Action Point to make a Roll that opposes that other robot's Test. This does not change the Turn Order. If a robot does not do this, then the attacker's / aggressor's Test is unopposed and likely to succeed by a large margin. One strategy in combat is to run ones opponent out of Action Points so they cannot defend and become easy targets.

The most common opposed roll in combat is to defend against an attack, but this is not the only use. Whenever another robot is making a Test, a combatant may pay an Action Point to attempt to oppose that Test if they can describe how their action would interfere with their opponent.

> ***Example:*** *The four friends are quarrelling again. The current Turn Order is Alfredo, Devon, Charles, and Burgiss. Alfredo takes his free action and attacks Charles. Charles elects to spend 1 Action Point to defend. Alfredo rolls 5 successes and Charles rolls 4. Alfredo wins with 1 success. If Charles hadn't defended, Alfredo would have won with 5 successes.*

It is up to the player to invent a plausible reason as to how their robot is attempting to oppose another robot's action. If they are being attacked by another robot, it is generally pretty easy to describe defending oneself with a parry or dodge. However, other actions may require a more creative approach. The GM is always empowered to decide whether or not a players desired method of opposition is plausible enough to proceed. If not entirely plausible, the GM may add difficulty to the roll or forbid it (as explained) entirely.

It should also be noted that the effect of the opposed roll does not *have* to (although most often will) actually oppose the acting robot's roll in the sense of canceling successes. Instead successes from it may be used for any of the things Test rolls can normally be used for if the GM feels that such a use would more accurately reflect the action being described.

Example: *In an earlier example we saw the pirate Francois le Requin being engaged by a warship from LaRoche. The GM described one of the LaRoche soldiers as throwing an EMP grenade down from the rigging into a group of Francois' men who were gathering to repel borders. Francois' player wishes to spend an Action Point to oppose this. Since Le Requin is on the quarterdeck rallying his men and not anywhere near the enemy soldier, the GM requires a justification. The player thinks for a moment and comes up with a couple of ideas.*

First, he could have Francois simply shout "look out" to his men. The roll would be Command vs. Perception (to see the danger and give an order). Any additional successes would be used to oppose the grenadier, with the explanation that the men diving for cover will make it more difficult for him to get an effective grenade attack. The GM buys this but decides that rather than *having the successes oppose the grenadier they should instead roll over into the individual opposed defense rolls of the men. Since this would involve providing a bonus to several robots, it would require applying the x1/2 Multiple Modifier.*

Instead Francois' player comes up with a more creative solution, one that opposes the grenadier directly instead of just adding dice to the men, and one that doesn't involve having to pay the x1/2 Multiple Modifier. He elects to shoot at the grenadier with his pistol. Now ordinarily an attack like this would have to be done on Francois' turn, not as an opposed roll. But the player explains that he isn't actually shooting the grenadier with the intention of hitting and injuring him. Rather he just wants to throw off his aim with the grenade by making him duck from the shot. Essentially, Francois' player has just come up with a way of doing "covering fire" without needing a special rule for it. The GM loves the idea and allows the successes from a Firearms (pistol) vs. Perception roll to cancel out the grenadier' successes on his attack.

Also note that it is entirely possible for another robot to oppose an opposed roll (but not by the robot whose initially declared action is the one being opposed first). In this case, the opposer of the acting robot's test has the successes from his roll canceled by his own opposer before using any remaining successes to cancel those of the initially acting robot. While it is possible to string together a series of opposed oppositions in this way, it is likely that plausibility (as judged by the GM) will deteriorate rapidly.

Pay for Turn:

After any turn, any combatant (including the one who just acted) may buy initiative by spending 1 point to move themselves to the beginning of the Turn Order and take the next turn. If more than one combatant wishes to do this at the same time (all such intentions are freely and openly announced), precedence is set as follows:

- If the acting combatant (i.e. robot currently at the top of the list) won the roll (or was unopposed and rolled at least 1 success) for their current action, they get first right to pay for turn. i.e. as long as you're successful you can keep paying Action Points to take several actions in a row.
- If the acting combatant had been opposed and the defender won the roll, the defender gets first right to pay for turn. i.e. as soon as you fail, the opponent who caused you to fail has first right to take the next action (likely to follow up against you).
- If these rights are not exercised, the combatant currently highest in the Turn Order takes precedence. Others keep their Action Point and remain in their current position. Note that the combatant who just took a turn has already been moved to the bottom of the Turn Order for purposes of determining "highest in the Turn Order".

- Note: the “after any turn” part above ensures that the first attack of any engagement goes to the robot at the top of the Turn Order. No one can buy initiative until *after* that turn.

Example: *Since Alfredo won his attack he gets the first right to pay for turn, which he does spending 1 Action Point to remain at the top of the Turn Order. Since it is again his turn he may take another action. He chooses to again attack Charles. Charles again spends an Action Point to oppose. This time around a bad roll leaves him with only 2 successes. Charles gets 4. Because the action was opposed and the defender (Charles) won, the right to pay for turn goes to Charles. Charles does so paying 1 Action Point. Alfredo is moved to the bottom of the Turn Order list and Charles is moved to the top. Charles has now spent a total of 3 Action Points.*

Charles now attacks Alfredo and wins with 3 net successes. As the winner, Charles has the right to pay for turn, but he elects to pass as he is running low on Action Points. Charles is now moved to the bottom of the Turn Order right below Alfredo. It is now open for anyone who wishes to pay for turn. If no one does the turn will go to Devon as he is next in the Turn Order. Both Burgiss and Alfredo decide to pay for turn. Since Burgiss is higher in the Turn Order (having been steadily bumped up by both Alfredo’s and Charles’s move to the bottom of the list), Burgiss gets precedent and is moved to the top of the Turn Order. Alfredo remains where he is (without spending any points). The Turn Order is now Burgiss, Devon, Alfredo, Charles.. Devon realizes that since he was next on the Turn Order list he could have guaranteed getting the next turn by spending an Action Point rather than hoping to get to go by default.

Using Perception Saves to react in time:

Often times a player may wish to have his character take some action and it is unclear whether the robot would actually be in a position to have seen what the player wishes it to react to. In such a situation the GM can call for a Perception Save setting additional Difficulty as the situation suggests. If the robot was being especially watchful, any Observation vs. Perception Test successes are ideal to use to increase the TN of the save.

- Roll 3d10 for the Save vs. Perception less any Difficulty.
- With a Total Success the robot sees the event in question in plenty of time to react to it without penalty. If taking a turn, the player can proceed with his turn. If making an opposed roll, there is no question that the robot saw its opponent’s action in time to oppose it.
- With a Total Failure the robot is oblivious and cannot react in time. If taking a turn, the robot can proceed to take its turn, but cannot perform any action that would have been predicated on noticing the event in question. If attempting to make an opposed roll, he cannot. The robot hasn’t seen its opponent’s action and therefore cannot attempt to respond to it. No Action Point is spent.
- With a Partial Failure, the robot did see the event in question, but must pay an additional Action Point penalty equal to 1 Action Point per failed die in order to actually react in time. If the player pays this he can react as described under Total Success above. If the player does not he cannot.

Example: In the above example of Le Requin vs. the grenadier, the GM decides to require Francois' player to make a Perception Save in order to notice the grenadier preparing to attack his men before allowing Francois to shoot at him. Previously Francois had made a Strategy (boarding party) vs. Processor Test and rolled a number of successes. He had kept those successes aside, undeclared, and now decides to use them to improve his Perception Save explaining that establishing a strategy would have involved studying the battle field and this is what enabled him to spot the grenadier's attack. The GM allows this and the player rolls his Perception Save. He gets two successes, which means there will be an additional 1 Action Point cost in addition to the normal 1 Action Point cost to make an opposed roll.

If the question at hand isn't whether the robot can spot the danger in time to react to it, but rather whether he's quick enough to react to something that's already well spotted, the GM may substitute a Processor (mental quickness) or Locomotion (physical quickness) Saving Throw instead, which then works the same way.

Refreshing Action Points

In the section on using success, there were 7 possibilities given. During an Extended Conflict an eighth is added. Successes from Tests can be spent to regain Action Points. Action Points cannot be increased beyond their maximum level, which equals Processor + Self Awareness. Note, this is conceptually identical to using successes to restore Temporary Attribute losses as per the Basic Mechanic's chapter.

This can be interpreted as making an effort to regain ones focus, both mentally and physically. Instead of pursuing an advantage to an overextended state, the robot backs off (uses successes on this option instead of other more aggressive options).

- Increase Action Points: Each success used can increase the robot's current Action Point total by one.

  Example: Alfredo again attacks Charles and Charles wins the defense by 2 successes. Rather than use those 2 successes from the defense roll to give himself +2 dice to a counter attack (using the roll-over bonus rules) Charles replenishes his stock of Action Points by 2 points.

[Ed Note: One possibility that may be dangerous but should be tested is to allow Successes to be spent to increase another robot's Action Points. I can see the justification for such usage: giving warning to a surprised friend, shouting a command to an underling, helping someone out of a jam to give them some breathing room. But I can also see situations where 1 character essentially becomes an AP factory supplying his comrades which may or may not be abusive. Also interesting to test is the ability to use the x1/2 Multiple modifier to restore Action Points to many robots simultaneously]

This is the only way to increase ones Action Point pool while engaged. A combatant who loses an extended series of rolls against a superior enemy will find himself running out of points and very vulnerable. Likewise a character who pays to string together several turns that ultimately don't result in victory may find themselves very low on points. Some successes from very specific rolls must be rolled over into other very specific rolls and cannot

be added to Action Points. This decision is made by the GM on the basis of the nature of the roll as described by the player.

Action Points and Disengaging:

The only other way to regain Action Point is if there is a break in the action where the combatants become disengaged. This can be either a soft break or a hard break.

A soft break occurs when there is a pause in the action but the combat is still continuing or expected to continue. The fighters may be warily circling, catching their breath, or may have just dispatched one enemy and not yet engaged another.

- During a soft break a non engaged combatant *may*, if he wishes, use his turn to start fresh by re-rolling Base Action Points.
- The Fixed Target Number changes to reflect the current numbers situation
- All current Action Points are lost (replaced by the roll).
- The character's current position in the Turn Order is not effected.

A hard break occurs when, for all intents and purposes the combat is over. The enemy has been driven off, the combatant has not only disengaged, but fled a safe distance. This does not preclude the combat from starting back up again shortly, as when the enemy rallies and reengages, or when pursuit catches the fleeing target. Any situation where the GM feels that one fight is over and a new one has begun is a hard break.

- During a hard break, all non engaged combatants *must* start over by re-rolling Base Action Points
- The Fixed Target Number changes to reflect the current numbers situation
- All current Action Points are lost. (replaced by the roll).
- A new Initial Turn Order is determined.

Reducing Opponent's Action Points:

Combatants may attack another robot's Action Points directly. Such an action could be any form of distraction, maneuvering into poor terrain, or other difficulty. The idea is to place ones opponent at a disadvantage so they can be defeated. This is handled using the Saving Throw mechanics.

Action Points can be targeted by a Saving Throw and suffer loss, just like an Attribute. When so targeted, the Action Point total is not used as the Target Number for the save (because it varies). Rather, an Attribute is selected that most ties into the nature of the Conflict. In combat that Attribute is most often Locomotion, but could be Perception, Processor or another attribute if such can be justified. Losses due to failure on the Save, however, reduce Action Points instead of the targeted Attribute; 1 per failed die as normal.

Combat Actions

There is no hard coded list of available actions, nor are the suggestions below to be taken as the only interpretation for an action. It is the player's job on their turn to suggest what they are attempting, both in terms of what it looks like, and in terms of what they're ultimately hoping to accomplish. It is the GM's job to determine how to model that into the standard Test and Saving Throw mechanics of the game.

Often times, experienced players will already have a good idea of what roll or combination of rolls they are looking to make. It then becomes the GMs job to ensure that the players description of their character's action is consistent and reasonable with the roll they are requesting. The GM always has final authority to define the nature of the roll, or require the player alter their narration to better fit the roll in question.

In the case of making opposed rolls, the GM may require a Perception Save to see if the robot can react in time, and then has the power to judge whether or not the player's described action is plausible enough to allow .

It should also be noted well that the ability of players of Sparked robots to spend Inspiration on Dramatic Control can throw a significant curve ball into this process. Dramatic Control can over ride a GM's sense of plausibility by introducing elements into a scene that make an action possible where previously it was not. As described under Dramatic Control these are minor elements if alls they accomplish is to permit a roll that could otherwise not have been made.

Example: *In the above example we saw Francois le Requin making a Perception Save to notice the enemy grenadier about to toss an EMP grenade at his men as they prepared to repel boarders. Francois' player could have used Dramatic Control to simply narrate the coincidence of having been looking in the right place at the right time to notice the enemy. This would be a minor use and cost 2 Inspiration because the only effect is to allow a roll (the opposed roll against the grenadier) which otherwise wouldn't have been possible (without a Perception Save).*

Alternatively, the player may use Dramatic Control to simply narrate how the grenade was fortunately a dud; or how the spar holding the grenadier up suddenly broke, sending the enemy robot plummeting to the deck, the grenade unthrown. Either of these uses would have been considered moderate and cost 4 Inspiration because they are effectively accomplishing the opposing of the grenadier without requiring a roll.

Basic Melee Actions

Players are encouraged to freely invent daring maneuvers and recreate staples of the swashbuckling genre. Samples of some of the special maneuvers they might come up with are given below. However, there are some basic maneuvers that player robots will likely want to perform frequently and so these are outlined here for ease of reference and to serve as a pattern for creating maneuvers of their own.

One principle that underlies these maneuvers was selecting a wide range of Attributes. The more Attributes are involved in fighting, the more paths to an effective duelist there are and the more weaknesses there are that can be creatively exploited. In this way, no character is a total rube in a fight, and no one is an invincible machine.

Basic melee Attack:

- Fencing vs. Articulation

This is the basic action of attempting to inflict physical damage on an opponent with a weapon. The most common result of this Test is to trigger a Durability Save in the opponent with additional successes used to reduce the opponent's Target Number.

Other uses may be to trigger an Action Point Save using Locomotion, describing the attack as being a flurry designed confuse and overwhelm them. Or to trigger a Locomotion Save described as driving the opponent back (perhaps into danger). Also, instead of using all of the additional successes for "damage" (i.e. reducing the opponent's save TN), the attacker may use some of them to refresh his Action Point total. This could be described as being a probing attack. Instead of committing to a major effort (using a lot of successes for damage) the attacker settled for a less damaging attack that would improve his overall tactical positioning (more APs).

Finally, the successes can be rolled over into other actions entirely. Successes from one attack can be rolled over into the next described as a set-up attack followed by a strong thrust to the opening. Successes may be rolled over into the next defense roll; especially useful if fighting multiple opponents. Successes may be rolled over into an Athletics vs. Locomotion Test described as being an attack forcing the opponent back, leaving an opening for the robot to disengage and run away.

Basic Melee Defense:

- Fencing vs. Locomotion

This action covers pretty much every non exotic defensive technique a duelist would employ. The exact nature of the technique used is left up to the player narrative but substantial customization is available depending on how the successes are used.

The primary use of this test is to cancel successes from an opponent's attack. Any additional successes can be spent in a variety of ways. A common use is increase Action Point. Since Action Points are required to make an opposed roll, 1 additional success will replace the 1 spent to make the roll. A second additional success will provide the AP needed to pay for the right to buy initiative and take the next turn that one gets for winning a defensive roll.

Other uses would include adding the successes to the next defensive roll, perhaps managing to build up a stronger and stronger defense. Or successes can be rolled into a follow up attack roll after buying the initiative in a riposte like move. With the right narration an additional success might trigger a Locomotion Save in the attacker to see if the attacker was thrown off balance by his failed attack. One could also see adding successes from a defense roll into an allies defense roll as two fighters in "back-to-back" or "shoulder-to-shoulder" fashion offer cover to each other. Alternatively, the successes could roll into an ally's attack roll, as one defender binds up the opponent's weapon, while the other goes in for the kill.

Off Hand Weapon Use:

A staple of the swashbuckling genre is the use of a second weapon or other object. The off hand weapon can be used on the robot's turn by spending an Action Point, or as part of an opposed roll by spending a second Action Point. The Tests are made just like for a regular weapon only using the Off Hand Weapon Program instead of the Fencing Program. There are two advantages to using an off hand weapon in this way.

During the robot's turn, it can roll the off hand weapon Test first; but this does not count as his turn. He does not have to succeed on this test in order to win the right to Pay for Turn. If the opponent opposes the off hand Test and wins, the opponent may use those successes to bolster its defense against the main weapon (or for another purpose) but does not earn the right to Pay for Turn. Essentially the robot gets 2 turns in a row, automatically without the possibility of being interrupted between them.

The player does have to describe his actions with both rolls before making either and then spend any additional successes in a manner consistent to this. If the player describes using the off hand weapon to beat the opponent's weapon off line thereby opening up an attack for the main weapon, then the natural effect is to use any additional Successes to add to the main attack roll. The player could not change this to making an actual attack and triggering a Durability Penalty after the roll.

Secondly, if used to oppose an opponent's attack, the robot essentially gets two defense rolls. Additional successes from the first roll can be used to enhance the second roll, thereby allowing for a stronger defense. Narratively, the player may describe this as a classic crossed swords block, or it may simply represent that the presence of the second weapon made it more difficult for the attacker to find an opening, so that even though the narrative may suggest just a one blade parry, it was the employment of the second blade that made that parry easier.

Applying Damage:

The concept of damage in Robots & Rapiers is handled by reductions to the target's Durability Save. The more damaging an attack, the lower the Target Number, and hence the more likely that the target will suffer damage.

There are 4 potential modifiers for damage for any attack: 1) successes from the actual attack roll used to reduce the Target Number. 2) successes from *other* rolls used to reduce the Target Number. 3) free successes from the weapon itself used to reduce (or increase if negative) the Target Number 4) Armor worn by the target which increases the Target Number.

The Durability Save starts as a standard saving through against the target's current Durability Attribute. Additional successes from the roll which triggered the save can be used to reduce the TN of this save. This represents a particularly solid or well aimed blow being more damaging.

Successes from other rolls can be used to reduce the TN of this save. This other roll could be a variety of things. The most common would be a Fencing vs. Force roll representing a particularly powerful swing employing the robot's great strength. The attacker makes a Fencing vs. Force roll and uses the successes to reduce the target's Durability TN. He then spends an Action Point to take another turn to make the actual Fencing vs. Articulation attack roll to trigger the Save using the successes from that roll to reduce the Durability Save TN as well. Other uses might include successes saved from an "attack to cover" where the attacker already has the weapon pressed against a vulnerable area and if the target moves the attacker will make an attack roll bolstered by those saved successes.

Each weapon has its own Damage rating. For sonic rapiers (the most common dueling weapon) this rating is 0. Other weapons may do more or less damage than this. The damage rating serves to increase (negative damage modifier) or decrease (positive damage modifier) the Durability Target Number.

Finally, the Target's Armor is accounted for. If the target is wearing armor increase the Durability Save value by the amount of the armor bonus. If the target is equipped with the Armor Plate accessory, than a number of successes from the above methods used to decrease the Target Number, are ignored, unless the total successes used in this manner exceeds the target's Size in which case the Armor accessory is ignored and has no effect.

If the Target Number of the Saving Throw is increased to 10 or more, then total success is automatic, the blow cannot harm the robot (save for knock down). If the Target Number of the Saving Throw is reduced to 0 or less, then total failure is automatic and the robot suffers shut down.

Basic Ranged Actions

There are 4 basic ways of making a ranged attack in the game: pistols, muskets, cannon, and thrown weapons (typically grenades). The first 3 all use the Firearms Program with the appropriate specialty. Throwing weapons uses simply the Athletics Program or defaults to the Core Physical Program.

Firearms:

Firearms in Auvernais do not fire physical projectiles. Rather they fire small magnetic capsules of energy accelerated electro magnetically down the barrel length. The speed of these softly glowing balls of energy is slow enough to track with the naked eye. The capsule delivers an EM pulse which was safer to by standing guests than actual bullets. Thus, as with all EMP weapons, failed dice from a Durability Save do not result in Malfunctions, but rather temporary losses to the Durability Attribute. Pistols cause 1 point of Durability loss per failed die. Muskets cause 2. Cannon cause 3. Heavy Cannon cause 4 points of Durability loss per failed die. Grenades, also an EMP weapon, cause 2. Grenades and cannon shells also have a burst effect.

Loading a Firearm:

The historical weapons that Auvernais firearms were based on had to be reloaded by hand after every shot. Being energy weapons, firearms in Auvernais don't have to be loaded after every shot, but they do have a significant delay between shots as the capacitor recharges. This delay is measured in robot actions. It does not matter which robot takes the action or what the action is, it is a simple passage of time.

Each weapon has a power cell which must be replaced after a certain number of shots. With an appropriate Mechanics (Repair Robot) roll, these cells can be drained to recharge a robot's own capacitors, or can be charged from a robot's capacitor. Each shot takes a certain number of charge points. Each point of Power from a robot's capacitor converts to a number of charge points equal to the robot's Size x 10.

Basic Shooting Attack:

- Firearms vs. Perception for firearms
- Athletics vs. Perception for thrown weapons

Shooting a charged firearm requires selecting a target and determining the range to the target in yards. Divide that range by the Range increment of the weapon, dropping all fractions. This number is the added difficulty to the shot due to range assuming a standard robot size target. Increase this difficulty by 50% for targets less than 1/2 robot size. Decrease the difficulty by 25% for targets more than double robot size; and by 50% for targets more than 10x robot size.

Aiming:

- Firearms vs. Processor for fire arms, *Fine Motor Sensitive*
- Athletics vs. Processor for thrown weapons

Aiming involves taking an action and rolling successes from the aiming Test over into the next Test. The next Test can either be a shooting Test, or another aiming Test. Most other

Tests or Saving Throws cancel the effect of aiming and the roll over bonus dice are lost. The GM can make an exception to this, if he deems the other roll won't interrupt the aim. Multiple aiming Tests cannot be used to augment a single shooting Test directly (parallel), but a series of aiming Tests can augment each other daisy-chain fashion (serial) with the last test augmenting the shot. Each successive Aiming Test has a cumulative +1 Difficulty.

The Fine Motor Sensitive penalties and the Difficulty Penalty for successive Aiming Tests can be avoided if the weapon is braced.

Hitting Bystanders:

Robots in Auvernais were programmed not to take a shot if there is even a remote possibility of hitting a human guest. Since there haven't been any guests in Auvernais for decades, this fail safe has not been called upon recently. If a robot shoots a fire arm and misses, and there is a chance of the shot hitting someone or something it shouldn't have, use the following rules:

- Roll 1d10 + 1d10 for every point of Difficulty the shot had
- If any of these dice roll a 10, randomly determine the bystander who gets hit.
- If more than one rolls a 10, then use these additional "successes" to reduce the TN of the bystanders Durability Save.

If the attack used a weapon containing a burst radius:

- Roll 1d10 +1d10 for every point of Difficulty the shot had
- Sum the numbers on these dice to find the number of feet of deviation for the shot
- Roll 1d10 for the direction. 10 is too far. Read the other number clockwise around so that 5 is directly short.
- Apply the burst radius as normal from the new location.

Dodging Firearms:

- Athletics vs. Locomotion

Dodging bullets may not be realistic, but it sure is entertaining. The firearms of Auvernais shoot small glowing spheres that, while faster than a fastball pitch, are still able to be dodged by alert robots.

If dodging a burst effect weapon, assume each additional success moves the robot 5' closer to the edge of the blast radius.

Sample Maneuvers

A non exhaustive list of some of the staples of derring-do fighting for players and GMs to use as inspiration. Players are encouraged to come up with far more possibilities then are listed here, and even change the specifics of what is here to account for differing situations.

Mud in the Eye: Dust in the eye doesn't work so well on robots, but mud or other similar substance on the visual sensors can have similar effect. Roll Dirty Tricks vs. Articulation to trigger a Perception Save. The target may oppose with Athletics vs. Locomotion to dodge the slung mud.

Cloak Tricks: Cloaks, capes, or other material make effective off hand devices in the hands of a skilled fencer. They may be used to blind a target and force a Perception Save; distract a Target and force an Action Point Save using Perception; or entangle a blade or sword arm and force an Articulation Save. The Roll is usually Dirty Tricks or Fencing vs. Articulation. Since cloak use is so common, the standard Fencing Program is already designed to counter it and so they can be opposed by Fencing vs. Locomotion or Articulation.

Leg Sweep: Tripping ones opponent is a staple tactic. Robots may try it hand to hand with Wrestling vs. Locomotion. Pole arms are well suited for hooking maneuvers using Fencing (Pole arms) vs. Articulation. Success can trigger a Locomotion Save or an Action Point Save using Locomotion. Defense might involve Athletics vs. Locomotion or Size.

Corp-a-Corp: Fencing often gets physical with punches, knees, shoulder blocks, flying elbows and even head butts. Brawling vs. Force or even Dirty Tricks (eye gouges and the like) can then be used to roll successes over into a follow up action like an attack or other maneuver.

Ridicule: This could be anything from a scathing taunt or mocking mimicry (using Oration vs. Processor) to physical embarrassment such as being spanked with a sword (Fencing vs. Articulation), or dumped in the mud (Wrestling or Brawling vs. Force). This can trigger a Processor Save as the robot struggles to come up with an appropriate response based on its programming. Total Failure indicates that the robot becomes irrational or confused and receives a -3 dice to all future rolls while in this state. Total Success, however, indicates the target becomes enraged or determined and gets a +3 dice to all rolls in this scene against the source of the ridicule (This is the usual 6 point Attribute loss with the Persistent Modifier as it effects all rolls and not just Processor).

[Ed Note: These are just a few of the obvious ones I came up with. This list can certainly be expanded greatly. I'd encourage play testers to make note of some the better ones used in play to add to the list]

Weapons and Armor

[ed note: this will eventually go in an equipment chapter or some such but is appended here temporarily for reference. Weapon stats require extensive play testing and exist only as a way to differentiate different types of weapons. Other than the EMP nature of the pikes, I've considered treating weapons as pure flavor and not differentiating between them at all, but am not convinced of that idea]

Melee Weapons

Weapon Table:

	Reach	Difficulty	Damage		
Robot Fist	0		-2*		
Small Club	1		-2*		
Medium Club	2	+1	0		
Large Club	3	+2	+2		
Vibro Dagger	1		-1		
Sonic Rapier	3		0		
Vibro Cutlass	2	+1	+2		
Mono Sword	3	+2	+4		
Mono Great Sword	4	+3	+5**		
Energy Pike	4		0***		
Boarding Pike	3		-1***		

* These are pure impact weapons which, given the metallic nature of a robot's body, are not as effective at inflicting real injury. The target only suffers 1 Malfunction per failed Durability Save Die.
** The great sword is capable of tremendous physical damage. The target suffers 3 Malfunctions per failed Durability Save Die.
*** These EMP weapons, which means instead of causing Malfunctions, failed Durability Save Dice result in a loss to the Durability Attribute.

Reach refers to the length of the weapon and addresses the idea that it is harder to strike an opponent who is defending himself with a longer weapon. The combatant with the shorter weapon (i.e. lower Reach) has added difficulty to any attack roll involving striking the opponent (i.e. triggering a Durability Save) equal to the difference in Reach.

This penalty lasts until the combatant with the shorter weapon manages to close the range at which point the penalty reverses and it is the combatant with the longer weapon who suffers the penalty until the range opens. Closing and opening the range is accomplished by either scoring a hit (triggering a Durability Save) or by using successes from another roll equal to the Reach Difference in a Test for that purpose.

Difficulty is additional fixed Difficulty to all rolls made with the weapon that involve the weapon being employed aggressively or defensively in a fight (i.e. it may not apply to a roll involving

intimidation with the weapon or using the Fencing Program for another purpose during combat). It reflects a weapon being heavier or clumsier in design, especially as it relates to the weapons "speed". Lost successes due to the Difficulty represent having to struggle against the inertia of the weapon.

Damage: Since most weapons that have higher difficulty due to being heavier, also have a higher potential to inflict damage due to being heavier, such weapons also normally come with a position damage modifier. Since one source of damage is additional successes from an attack roll, applying Difficulty to an attack roll reduces not only the chance to hit, but also the damage from the attack. Therefore, these weapons have a Damage modifier that, if the attack manages to hit, allows the weapons to do more damage than they otherwise would.

The damage modifier is applied as an automatic reduction to the target's Durability Save TN. Negative damage modifiers increase the target's Durability Save TN.

Weapon Descriptions:

Robot Fist: The fist of a robot is, of course, mechanical in nature. Which means its effectiveness is about the same as being struck by an armored gauntlet. Of course, robot bodies are also mechanical in nature, and so aren't prone to pain, or contusions, or blows to sensitive areas. Blows with a fist are thus less effective at causing injury, and only inflict 1 Malfunction per failed Durability Save Die.

Clubs: These basically encompass any weapon that relies on physical impact for damage. As such they are not as effective as more exotic weapons against robots. Pistols can be used as small clubs, and muskets can be used as medium clubs.

Vibro Dagger and Vibro Cutlass: These weapons employ advanced technology to increase their effectiveness. The edge of these weapons are serrated at the molecular level (invisible to the eye) and an electro magnetic mechanism in the hilt causes the blade to vibrate thousands of times a minute like a microscopic reciprocating saw. This weapon is much more effective at inflicting physical damage on a robot than mere impact weapons.

Sonic Rapier: The sonic rapier operates under the same principle at the vibro weapons except the vibrations are set up in the long slender blade via sound waves rather than mechanical means. The special harmonics this sets up in the blade causes the extremely hard and sharp point of the weapon to essentially drill through the metallic casing of an opponent. Since this is the "standard" swashbuckling weapon, it is the default, and all other weapons are rated according to their effectiveness against it.

Mono Sword and Mono Great Sword: These weapons rely on the physical impact of the swing (also thrust for the Mono Sword) to inflict their damage. But unlike mere clubs the edges of these weapons have been sharpened down to the molecular level. Their edge is a mere 1 molecule wide. As such their true edges are typically recessed inside the hefty center of the blade for safety and only extended when needed. The weight of these blades makes them slower and more unwieldy than a rapier, but the impact of the mono edge is capable of

doing tremendous damage. In addition to a high damage rating, the Mono Great Sword also inflicts 3 Malfunctions per failed Durability Die instead of the normal 2.

Energy & Boarding Pike: The Energy Pike is about 6' long, the Boarding Pike about 4'. They are wielded more like a halberd than the long pike from the battle field. Its business end is topped by a fork shaped blade that kind of resembles a wide spear tip split down the middle. These blades are capacitors which release a sharp highly localized EMP pulse when in the proximity of a target. This pulse is designed to disable rather than damage and so causes no Malfunctions. Rather each failed die on a Durability Save results in the Temporary loss of 2 points of Durability (much the way Saving Throws normally operate for other attributes).

Missile Weapons

Firearms Table:

	Damage	Durability Loss	Delay	Charge per Shot	Shots per Cell	Burst Radius	Range (yds)	
Small Pistol	-1	1	1	1	2 (2)		2	
Pistol	0	1	2	1	8 (8)		5	
Heavy Pistol	0	2	4	2	4 (8)		5	
Carbine	0	2	3	2	6 (12)		10	
Musket	+1	2	4	2	6 (12)		15	
Heavy Musket	+1	3	6	3	4 (12)		15	
Light Cannon	+2	3	6	6	3 (18)	Small (0)	25	
Medium Cannon	+3	3	8	9	2 (18)	Medium (+1)	50	
Heavy Cannon	+4	4	12	18	1 (18)	Large (+2)	100	
Grenade	0	2	n/a	2	1 (2)	Small (-2)	5	

Damage: The damage modifier is applied as an automatic reduction to the target's Durability Save TN. Negative damage modifiers increase the target's Durability Save TN.

Durability Loss: Fire arms are EMP weapons, which means instead of failed dice on a Durability Save inflicting Malfunctions, they inflict a temporary loss to the Durability Attribute. The number of Durability points lost per failed die equals this Durability Loss.

Delay: Delay means the number of robot actions (typically individual Test rolls) that must occur after the weapon is discharged, before it can be discharged again. This represent the time it takes for the capacitor to recharge. A Mechanics vs. Processor roll can be made with successes used to reduce time as per the basic rules.

Charge per Shot: This is the number of charge points each shot uses. 1 Power Point converts to charge points at the rate of 1 : Robot's Size x10. A robot repair roll can be used to convert power back and forth between the robot's capacitor and the weapon's charge cell. "Living on Ammo" is a saying common to Sparked robots in Auvernais who, lacking proper recharge facilities, manage to keep going by draining charge from weapon cells.

Shots per Cell: There are 4 standard Cell sizes for weapons which can be used in any weapon that takes that size. The number in parenthesis in this column reflects that size rated in total charge points. Dividing the total charge by the Charge per Shot indicates the number of shots each cell can provide before needing recharged itself.

Burst Radius: Cannon and grenades have a burst radius which effects all of the robot's in a given area. If using successes on the attack roll to reduce the Target Number of the Durability Save they must use the x1/2 Multiple modifier if more than 1 target is inside the Burst radius. Small, Medium, and Large equate roughly to 10' / 20' / and 30' radius. The weapon does its full damage if within 1/2 of this radius. The weapon does reduced damage (as indicated by the number in parenthesis in this column) beyond this out to its maximum radius.

Range: Range is measured in yards. Each increment of range is a cumulative +1 Difficulty to the Test.

Weapon Descriptions:

Small Pistol: The small pistol is a small concealable weapon like a derringer or pepperbox, effective only at close range.

Pistol: The standard side arm in Auvernais, modeled off of an old flintlock

Heavy Pistol: Like a pistol only bigger

Carbine: A short light musket generally used from horseback, or as a sporting gun

Musket: The standard long arm in Auvernais, modeled off of an old flintlock

Heavy Musket: Like a musket only bigger an heavier. This weapon suffers a fixed 2 difficulty unless fired braced; typically from the long brace provided with the gun.

Light Cannon: a small cannon mounted as a swivel gun on ships or occasionally from a carriage. Also pulled behind horses on the battle field

Medium Cannon: a large cannon mounted in a wheeled carriage. Typically deployed on ships or as the main armament of a fort.

Heavy Cannon: monstrously large gun used for sieges.

Grenade: A cannonball sized sphere that is set with a timer and thrown. Emits a flash pulse of EM radiation to disable enemy robots.

Armor

[ed note: Like weapons, the values need to be heavily tested. Also, there is a difference between how worn armor is handled vs. the Armor Plate Accessory. This either needs to be justified or the better method used for both]

Armor Table:

Armor Type	**Bonus to Durability Save**			
Standard Tunic & Jerkin	1			
Heavy Jerkin	2			
Brigandine	3			
Cuirass	4			
Corslett	5			

Bonus to Durability Save: This value is added to the Durability Save vs. any weapon attack or other physical impact the GM judges would be aided by armor. The armor bonus is unaffected by EMP weapons.

Armor Description:

Brigandine: a leather jacket with metal plates riveted inside. Typically worn by on duty guardsman or musketeers in battle.

Cuirass: A breast and back plate typically worn for parade or by cavalry in battle

Corslett: Heavy armor consisting of a Cuirass plus helmet (head), Gorget (neck) Pouldrons (shoulders), and Tassets (thighs). Typically worn by pike men in battle.

Interpreting Player Actions

Translating the player's statement of intention into game mechanics can often be a challenging task for the GM. Robots &Rapiers does not offer lists of available moves or standard modifiers; instead the GM must decide how to mechanically portray a particular action by combining the basic core mechanics on an ad hoc basis.

Program and Attribute:

The most basic choice the GM will have to make is which Program serves as the number of dice to roll in a Test. The simplest situation comes when the robot possesses the Program in question. If not, or if there is no Program that covers the action, the next step is to evaluate whether the action is something that, in a general sense, a character of that type would reasonably be expected to have some basic ability in. If so, then the GM can use one of the 3 Core Programs as a "default". If this is not the case either, if the desired action is outside realm of what a typical individual of that background would likely know, then there is no appropriate Program or Attribute combination. The only means for the robot to accomplish this action is with the rules Sparks have for Inspired Rolls.

The Attribute choice offers similar challenge. Unlike the Core Programs which are permanently linked to the individual Programs, *any* of the Attributes can be combined with any of the Programs depending on the nature of the action in question. While the six Prime Attributes are the most common choices; Size, Anthropoid Class, and Perception are all widely useful.

Size can be used in any situation where being large and heavy is an advantage, such as Brawling or even Intimidation. It can also be used, with a little tweaking, for situations where Size is a disadvantage like squeezing through a tight spot, or crossing a rickety bridge. Simply determine the largest Size that can operate without difficulty and add 10. Then subtract the Size from this number to get a useable TN.

Anthropoid Class is a direct measure of how human the robot looks and sounds, but more importantly it serves as a proxy for social status and pecking order. As such it makes an effective Attribute to use in situations where a higher class robot has an advantage, or in situations where other games might use an attribute like charisma.

Perception is also an Attribute that can see a lot of use. This measures the raw sensor capacity of the robot. Actual knowledge on how to be alert and observant is covered by the Observation Program or, in lieu of that Program, by the Core Mental Program. Observation vs. Perception is a common test for spotting hidden things, but don't discount Observation vs. Processor as a roll for figuring out where the hidden things might be. Perception itself is an Attribute that actually must be assembled from the collection of 3 Sensors (more if equipped with Olfactory or Taste Accessories). The standard Perception Attribute is Visual + Auditory, but Visual x2 or Auditory x2 is appropriate for pure vision or hearing Tests, or Auditory + Tactile for feeling ones way through a pitch black room.

Opposed Rolls and Difficulty:

The next question a GM must decide is whether there is any opposing roll or added Difficulty working against the character. When using Extended Conflicts there are formal rules requiring spending Action Points for when another robot can make opposed rolls. When not using Extended Conflicts, it is up to the GM to decide if such is appropriate. Keep in mind that an opposed roll does not have to come from another robot. Any inanimate object that can be said to be working against the character may be a source of an opposed roll. Since inanimate objects are not generally given game stats, the GM will have to select dice and an appropriate TN.

Difficulty, as has been explained in the Basic Mechanics chapter, is nothing more than assuming a given number of successes from an opposed roll without having to actually make the roll. This can be quicker and easier for the GM to adjudicate. Assigning Difficulty is not an exact science, however, there are some simple guidelines to follow.

First, recognize that a minimal level of achievement is accomplished with just a single success and that single successes are pretty easy to come buy. It will be very rare that any robot will make a roll in a field it is reasonably competent in and not be able to score a single success. This is a completely reasonable state if the roll is not so much a question of whether the robot will succeed or not, but rather; given that the robot likely will succeed the roll is to determine by how much (and how many additional successes the player will have to spend). If, however, there should be a real measurable chance of failure involved, then there must be either an opponent, a proxy for an opponent, or an assigned Difficulty score.

Second, a handy formula to keep in mind is: # of dice x (TN/10); which is nothing more than a quick and dirty way of calculating an expected level of success for any roll. For instance rolling 8 dice vs. a TN of 6 gives just under 5 successes as an expected roll. What this means is that there is an equal chance of rolling less than 5 as there is of rolling more than 5. By setting the Difficulty to this expected level, the most likely outcome of the roll will be 0 net successes (failure), with an equal chance of getting more or less. By setting the Difficulty to 1 less than this expected level, you are making the most likely outcome be the minimal result of just 1 success.

Third, keep in mind, however, that the meat of the system relies on players generating an excess number of successes and then spending those successes on the various bonuses that power the system. Setting a difficulty too high, or opposing every roll, will choke those additional successes off rendering the system highly ineffective. Given the action oriented, high flying derring-do, associated with the swash buckling genre, the GM's primary concern when setting difficulty is to set enough of a challenge to make things interesting, but not so high as to rob the players of the ability to make their characters perform breath taking stunts and daring feats.

Compound Actions or Multiple Tests:

Frequently, what the player describes his character doing ultimately represents several distinct parts, each of which could be interpreted as a separate action. Robots & Rapiers provides two alternate ways of handling this.

Compound Actions combine all of the separate actions into a single roll. Each separate action requires its own success to "Accomplish Action" and any additional successes beyond these can be used to boost the effectiveness of any of the individual parts. Compound Actions are a great way to simply combine several small related parts into a single roll, and are useful for a variety of situations.

It should be noted, however, that requiring more than one "Accomplish Action" success, has essentially the same effect as increasing the Difficulty of the roll. In fact, the Compound Action concept is a great way to justify adding Difficulty to a roll and provides the added benefit of allowing "partial success" where some but not all of the individual pieces succeed.

Multiple Tests: Instead of combining all of the actions together, the GM may require separate rolls for them. In the Extended Conflict rules, this requires picking one of the actions to attempt first, and then paying Action Points to take another turn and attempt the next one. This usually requires winning the first action so that the robot has the right to purchase a consecutive turn. It is certainly possible, however, to envision several related actions that don't necessarily have to be right in a row to be effective.

While Compound Actions are generally quicker than making multiple tests, multiple tests are really the heart of the game system, especially when those tests are linked together in effectiveness enhancing combinations. Consider a situation where a robot is making a roll and being opposed by a robot of equivalent ability. If both are expected to roll about the same number of successes, than the most likely result is going to be failure (see the discussion on Difficulty above). However, this is by design. What the player will want to do in order to succeed is think of other actions he can perform that he can roll unopposed and then use successes from those rolls to add additional dice to the main roll. With enough additional dice pumping up the main roll, he can expect to not only succeed, but succeed dramatically.

This is where strings of exciting combinations get assembled ad hoc by the players in an attempt to build their effectiveness high enough to defeat their opponents.

Example: *Charles, the hot headed King's Guard, was stewing in his cups at the Tavern on the Green when he suddenly takes offense at a comment made by a hapless young Cardinal's Guardsman nearby. Charles's player announces that he is going to leap up from the table, attack the "villain" with his rapier while shouting deprivations about the man's mother and upbringing and something involving small farm animals.*

The GM now must parse out this statement of intent into actual game mechanics. There are essentially 4 actions the player has declared for Charles. 1) he needs to get up from the table, 2) he needs to draw his sword, 3) he's going to attack the Cardinal's man, and 4) he shouting insults.

The first 3 are easy, they can be done as a Compound Action. Does standing up and drawing a weapon really need *to be treated as a separate action? Well, the GM must decide; is it more difficult to make an effective attack if one has to stand and draw a weapon first, as compared to someone already in a fighting stance with weapon drawn? If so, the extra successes "wasted" on those two actions reflect the decline in effectiveness.*

The last one requires a little more judgment. It could *also be done as a Compound Action. Charles' player makes a single roll, probably Fighting vs. Articulation. 1 success, rises from the table, 1 draws the sword, 1 hurls the insults and triggers a Processor Save, and 1 lands the attack and triggers a Durability Save.*

Alternatively the insult roll could be done as a separate Oratory vs. Processor roll. This would increase the likelihood of both the attack being successful (1 fewer required success) and the likelihood of the insult being particularly effective (potentially a lot of successes to reduce the target's Save TN with). However it requires 2 actions, which using the Extended Conflict rules would cost an additional Action point. It would also require deciding which action to roll first and the first action would need to be successful before the second could be rolled.

Further, in a different circumstance with a different character, even the initial standing up and drawing the sword might be done with a separate roll. A player whose character has a high Locomotion Attribute might want an Athletics vs. Locomotion roll as their first Action, using 1 success to stand, 1 success to draw the sword and using the remainder to add additional successes to their next action to represent the advantage gained by speed and surprise of the attack

Using Saving Throws:

Any time a player indicates that he wants to do something to another robot is a sure sign that a Saving Throw is called for. If robot A is performing some action and the result of that action will be to have some effect on robot B that robot B really doesn't want, its time to roll a Saving Throw.

Saving Throws are triggered by the "Accomplish Action" success from Tests whose description involves such an effect. That single success is enough to cause the target to make the save. Additional successes on the Test can make the save more difficult (more likely to fail and result in bad things for the target). This is how damage from attacks is accounted for in Robots & Rapiers. One of the key reasons to string together a combination of several actions is to roll over successes into bonus dice until at last, in the final action, so many dice are rolled that enough successes are generated to make the Save almost guaranteed to fail.

The GM should require the player to be clear about exactly what result he'd like to come from such a save. This can include almost anything from inflicting actual physical damage, to having the robot run in fear, to throwing the enemy off a cliff; if its consistent with the overall narration of the action.

Non Combat Uses for Extended Conflicts

Combat is a common feature of both swashbuckling and role playing. It also provides a fairly well defined situation with easily understood parameters for purposes of illustrating rules. Hopefully players and GMs will see how the above rules can be expanded to cover a variety of situations in a game.

The four keys to using Extended Conflicts are:

- Robot to robot conflict, at least 1 on 1 and particularly in larger groups. There must be some form of stakes that the robots are competing for. In combat the stakes are obvious and easy to quantify, life and death as measured by damage and injury. This is why combat is such a common feature in many RPGs
- A complex situation where there are many variables, many possible approaches, and many possible responses. If the entire conflict can be boiled down to just a single die roll or two, than these rules become superfluous.
- Third, a need to regulate the pacing of the conflict. Who says or does what and when is the entire purpose of Action Points and Turn Order. They provide a means of limiting how many rolls a robot can collect roll-over bonuses from and when or if a robot can oppose another's Test. If the situation at hand has its own obvious pacing, such as perhaps an interview with the King, where simple logic and narration dictates these factors than an Extended Conflict is probably not necessary.
- Finally, the conflict must be dramatically interesting. Using the extended conflict rules will not take a mundane activity and suddenly make it interesting by making more of a game out of it. Instead it will take a mundane activity and drag it out until players become bored with it. If the situation is not dramatically grabbing, then the GM should be trying to get through it as quickly as possible (or better yet, avoid it entirely) not extending it.

Social Based Extended Conflicts

These could be a variety of situations, often centering on court or a ball or a garden party, or intermission at the theater, or other formal occasion. The key here is that there must be a source of conflict between the participants and there must be some form of stakes that they are competing for. If there are no stakes, there is no conflict nor a need for these rules. The scene is simply one of atmosphere and perhaps a couple of events. Stakes might include: Obtaining information from others wishing to conceal it. Currying favor with the right parties by drawing attention to oneself in a flattering light. Undermining the position of another robot with rumors and innuendo. Gaining the attention and affection of a lady who is also the subject of wooing from another (or multiple other) sources, and other possibilities.

The situation should be one where it is possible to measure the stakes and there is a potential for the effect to cut both ways, like damage in combat.

Changes to the above rules:

- Initial Turn Order is based on Processor instead of Perception. Ties are broken by Perception, Inspiration and d6 roll in that order.
- Ambushes and Surprise are more likely to be based on Intrigue and Oration than Conceal and Stealth. An Etiquette Save is used instead of a Perception Save to see how well the robot handles the shock.
- Base Action Points are determined by Anthropoid Class + Self Awareness instead of Processor + Self Awareness.
- The Fixed Target Number is determined by a comparison of Anthropoid Class to the mean. This mean may be set by the GM if the actual numbers in question are unknown (such as determining the actual average at a ball of several hundred).
    - Target Number = 5 if Class equals the mean
    - Target Number = 6 If Class is greater than the mean, 7 if 2x greater
    - Target Number = 4 if Class is less than the mean, 3 if 2x less.
- Instead of rolling Durability Saves for Malfunctions, the likely goal is to roll Anthropoid Saves for loss of Favor, or loss of an arbitrary score invented simply for tracking the progress of the conflict.

Fighting without the Extended Conflict Rules

The Extended Conflict rules are great for set piece battles, especially those involving a number of combatants and a full fledged fight. Sometimes, however, play will not call for such a full blown approach. On such occasions, fighting can be handled straight using the basic mechanic rules without resorting to Action Points and Turn Orders

All of the rules are in place, except that there is no resource that needs to be spent to make an opposed roll. Robots can continue to make Tests that augment each other in a combination fashion, as long as each Test succeeds. As soon as a robot fails a Test or "completes" a sequence, the GM selects another robot to take a turn based on Perception score or narration.

Detailed Table of Contents