

The Software Life Cycle Applied to the Development of the
Avnul RPG Character Creation Program

James Carlyle Rogers III

A Senior Thesis submitted in partial fulfillment
of the requirements for graduation
in the Honors Program
Liberty University
Spring 2013

Acceptance of Senior Honors Thesis

This Senior Honors Thesis is accepted in partial fulfillment of the requirements for graduation from the Honors Program of Liberty University.

Mark Shaneck, Ph.D.
Thesis Chair

Terry Metzgar, Ph.D.
Committee Member

Mark Harris, Ph.D.
Committee Member

James H. Nutter, D.A.
Honors Director

Date

Abstract

This project will endeavor to replicate the "cradle-to-grave" software engineering life cycle development process. This process takes a software project through many stages of existence, from requirements gathering to software design, coding, testing, and maintenance. In this thesis we apply this process to a character creation program, which will be constructed for a client to facilitate the design of his tabletop roleplaying game.

Over the course of developing this software, contact was maintained with the client, but difficulties were encountered in the form of shifting requirements as he refined and modified his as-yet unfinished game rules. Without regular communication, these changes would not have been adequately reflected in the final work. It was also observed that while testing is often an afterthought tacked onto the end of the development process, keeping it in mind during early stages of development makes its final implementation much easier. Due to technical difficulties in combination with the aforementioned shifting requirements, the software itself remains in development.

The Software Life Cycle Applied to the Development of the
Avnul RPG Character Creation Program

Project Overview

The purpose of this project is to develop a functional software system for the client, Tobias Mastgrave, who is a published fantasy author and is currently attempting to develop a tabletop roleplaying game (RPG) based on his work of fiction. The software system he requires is a character creation utility, designed to facilitate the process of developing valid roleplaying characters within his established framework of rules.

Mastgrave's fictional works are set on a continent called Avnul, which features a variety of races and cultural groups. Mastgrave borrowed heavily from East Asian and Middle-eastern cultures for the linguistics of his civilizations; the population of humans, for example, are known as the Longminjong. Other races include the saurian Saru and the Neshilim, who speak a language based on Hebrew pronunciation. Since Mastgrave is an avid scholar and hand-to-hand martial artist, his fictional world and its RPG derivative place heavy emphasis on unarmed combat and scholarly pursuits such as philosophy. While the current published volume set in Avnul is very much philosophical and theological in nature, Mastgrave's roleplaying game seeks to be much more action-oriented, offering players many chances for simulated combat and team-centric character development.

While Mastgrave's universe is intricately detailed, the variety of fictional facts does not translate directly into software. In order to create a program that adequately models both his fiction and his game rules, Mastgrave's needs for the software must be

translated into a list of usable requirements. Stated below in numerically ordered format are the various requirements for the system, gleaned from meetings with the client and from perusal of his proprietary documentation regarding the ruleset for his upcoming game. Accompanying this ordered list is a verbal summary of the requirements of the project.

Requirements List and Design Details

1. Basic System Requirements

1.1. **The system must be designed to run on current 64-bit Windows architecture**

(i.e., Vista, Windows 7, or Windows 8). This requirement will have very little to do with actual system design, but will affect the final compilation of the executable file.

1.2. **The system must utilize a graphical user interface (GUI).** A GUI will make the system more inviting for users than a text-based interface, and it will make it easier to handle invalid text input.

1.3. **The system must contain support for random number generation to simulate the use of multiple 10-sided dice.** This will necessitate a random number library function, and use of the mod operator (%) to ensure that only numbers 1 through 10 are used. Repeated calls to this function can be used to simulate the use of multiple dice. (According to the current version of the RPG, this function is not used; it is included as a utility for future iterations of the program to build upon.)

- 1.4. **The system must progress in a linear manner.** The character creation process has a series of defined steps, and they must be completed in order. The GUI of the program should progress in the same order.
- 1.5. **The system must enforce gameplay prerequisite rules.** Mastgrave's system uses a fairly intricate prerequisite system, where certain choices early in the character creation process affect choices that occur later on.
- 1.6. **The system must allow an overview of current character details at any time during the creation process.** Because the creation process is so lengthy, it will be necessary for users to see what they've already done without having to undo previous choices.
- 1.7. **The system must allow review of completed characters.** On paper, this concept is known as a "character sheet," wherein all useful data items for gameplay purposes are shown in concise form for review.
- 1.8. **The system must be able to save character data.** Roleplaying games can last for a very long time, so Mastgrave wants the system to be able to save character data for later reuse, so that the program doesn't need to be kept running as a reference for hours, days, or weeks.
- 1.9. **The system must be able to reload saved character data into a reviewable file.** After a character is fully created, it can be reloaded at any time and reviewed, according to requirement 1.6.
- 1.10. **The system must be modular.** Mastgrave intends to expand various features of the game as time progresses, including the addition of new races, skills,

abilities and an entire section for choosing items. The program must be designed to facilitate such additions at a later date.

2. Miscellaneous Data Requirements

- 2.1. **Floating Points.** Not to be confused with a floating-point data type, these are actually integral numbers. These points are used to purchase optional add-ons for characters at various points during creation. The value will be initialized to 20 and decremented appropriately by the program for every purchase made during creation.
 - 2.2. **Background Points.** These points are used in creation, and they start with a value of 0. They are then modified by the program during creation based on player choices. After the number of points reaches 0 again, no more background information can be created.
 - 2.3. **Attribute Points.** The player will start with 18 Attribute Points: these points will be modified by the game logic, and subtracted from during Attribute selection to keep gameplay balanced between different character types.
 - 2.4. **Virtue/Vice Points.** Every character has 10 of these to distribute: these points are spent to modify Virtues and Vices data during character creation.
 - 2.5. **Skill Points.** Each character is given 64 Skill Points, which are used to purchase additional ranks in Skills during character creation.
 - 2.6. **Ability Points.** Characters start with 10 of these by default, though more can be obtained by selecting a variety of options during character creation.
- ## 3. Character Data.
- Whereas the previous information is needed to *create* a character, the following actually *represent* the character.

- 3.1. **Concept.** The Concept is not used for any mechanical computation but is necessary for the purposes of roleplaying. It consists of a brief description of motivations and how the character approaches problems.
- 3.2. **Race.** While more races may be added in future versions, the current game only allows one race: the Longminjong, or humans. However, Mastgrave has plans to expand this option at a later date with other races from his work, so the program must be flexible to allow other races to be added.
- 3.3. **Name.** Every character must have a name for purposes of gameplay, though it will not affect character generation after it is entered by the user. There is no need to separately store different parts (e.g., first name, surname), since no computation is performed on the name of a character at any point.
- 3.4. **Age.** Every character has a numerical age (units are in years; Avnul years equate to the standard Earth years.) This will affect certain statistic information later in the creation process.
 - 3.4.1. Depending upon age value and race, one of the following categories will be assigned; depending on age category, certain changes will be encountered later in the creation process.
 - 3.4.1.1. **Not Valid** if Human age is less than 5 years.
 - 3.4.1.2. **Child** if Human age is from 5-10 years.
 - 3.4.1.3. **Adolescent** if Human age is from 11-16 years.
 - 3.4.1.4. **Adult** if Human age is from 17-35 years.
 - 3.4.1.5. **Middle Age** if Human age is from 36-50 years.
 - 3.4.1.6. **Elder** if Human age is from 51-65 years.

3.4.1.7. **Ancient** if Human age is greater than 65 years.

3.5. **Gender.** Only two genders are valid, male and female. Mastgrave’s world is not home to any androgynous or hermaphroditic races.

3.6. **Background.** This determines a character’s historical skillset. While the system is designed to have one background for a character, it is also possible to purchase multiple backgrounds for characters that have gone through several careers over the course of their lives.

3.6.1. Characters may possess a number of background options. Listed first is a name, followed by a cost, which is to be subtracted from the running total of Background Points during creation. Listed last is an indicator of specialization, which determines how a background matches up to the character that possesses it.

- | | | | |
|-----------|---------------------------------|-----------|------------------------------|
| 3.6.1.1. | Alchemist, cost 4, Academics | 3.6.1.15. | Engineer, cost 2, Expert |
| 3.6.1.2. | Economist, cost 4, Academics | 3.6.1.16. | Linguist, cost 1, Expert |
| 3.6.1.3. | Priest, cost 5, Academics | 3.6.1.17. | Sorcerer, cost 3, Expert |
| 3.6.1.4. | Scholar, cost 1, Academics | 3.6.1.18. | Investigator, cost 2, Social |
| 3.6.1.5. | Strategist, cost 5, Academics | 3.6.1.19. | Noble, cost 3, Social |
| 3.6.1.6. | Sahnsing, cost 5, Academics | 3.6.1.20. | Officer, cost 3, Social |
| 3.6.1.7. | Merchant, cost 3, Business | 3.6.1.21. | Socialite, cost 2, Social |
| 3.6.1.8. | Craftsman, cost 1, Business | 3.6.1.22. | Spymaster, cost 4, Social |
| 3.6.1.9. | Artist, cost 1, Business | 3.6.1.23. | Caravanner, cost 1, Travel |
| 3.6.1.10. | Crime Lord, cost 5, Business | 3.6.1.24. | Explorer, cost 1, Travel |
| 3.6.1.11. | Administrator, cost 2, Business | 3.6.1.25. | Mendicant, cost 2, Travel |
| 3.6.1.12. | Combat Engineer, cost 2, Expert | 3.6.1.26. | Ranger, cost 2, Travel |
| 3.6.1.13. | Combat Medic, cost 3, Expert | 3.6.1.27. | Thief, cost 1, Travel |
| 3.6.1.14. | Doctor, cost 3, Expert | 3.6.1.28. | Marksman, cost 2, War |

- 3.6.1.29. Martial Artist, cost 3, War 3.6.1.31. Soldier, cost 2, War
 3.6.1.30. Mercenary, cost 2, War

3.7. **Bloodline or House.** Similarly to Background data, these choices will affect later options during creation, but only one of them can be chosen. (Bloodlines are reserved for non-human races, and are included here only for completion to facilitate later modification. Bloodlines will not be visible to users in this version of the program, though the Longminjong Houses will be.)

3.7.1. The following list summarizes the Bloodline and House options required by the program. Listed first is the House or Bloodline name, followed by its cost, which is to be subtracted from Floating Points during creation if selected.

3.7.1.1.	Mei, 1	3.7.1.11.	Panja, 1
3.7.1.2.	Tuzi Dao, 1	3.7.1.12.	N'yuta, 1
3.7.1.3.	Jitai, 2	3.7.1.13.	Giragita, 2
3.7.1.4.	Juren, 2	3.7.1.14.	Khola, 2
3.7.1.5.	Wan Guen, 3	3.7.1.15.	Joda, 3
3.7.1.6.	Yingye Guo, 3	3.7.1.16.	Vicara, 3
3.7.1.7.	Wan Hu, 4	3.7.1.17.	Jahara, 4
3.7.1.8.	Yinying Kuishi, 4	3.7.1.18.	Na'e Sarira, 4
3.7.1.9.	Wuyi Zhu, 5	3.7.1.19.	Ajagara, 5
3.7.1.10.	Yaoren Bi, 5	3.7.1.20.	Naga, 5

3.8. **Dedication.** Dedications represent a character's trade focus and are represented as contracts and businesses. It is possible for characters to have more than one Dedication, and as a result of earlier choices made in character creation, some Dedications will not be available.

3.8.1. Dedication options are as follows. Each will need to be available for selection within the program.

3.8.1.1.	Fa de Kong	3.8.1.24.	Fa de Di Magic
3.8.1.2.	Fa de Shui	3.8.1.25.	Fa de Huo Magic
3.8.1.3.	Fa de Di	3.8.1.26.	Fa de Zhong Magic
3.8.1.4.	Fa de Huo	3.8.1.27.	Fa de Mu Magic
3.8.1.5.	Fa de Zhong	3.8.1.28.	Fa de Shu Magic
3.8.1.6.	Fa de Mu	3.8.1.29.	Fa de Dong Magic
3.8.1.7.	Fa de Shu	3.8.1.30.	Fa de Yingzi Magic
3.8.1.8.	Fa de Dong Yu	3.8.1.31.	Shu Kui Magic
3.8.1.9.	Fa de Yingzi	3.8.1.32.	Gun Magic
3.8.1.10.	Houshi and Yingshi	3.8.1.33.	Veraughn Magic
3.8.1.11.	Langshi and Sheshi	3.8.1.34.	Gojin Magic
3.8.1.12.	Shenlingshi	3.8.1.35.	Yaojin Academy
3.8.1.13.	Zhishi	3.8.1.36.	Hall of Herbs
3.8.1.14.	Danshi	3.8.1.37.	Knowing Wall
3.8.1.15.	Chenshi	3.8.1.38.	Square of Words
3.8.1.16.	Maodao	3.8.1.39.	Luntan
3.8.1.17.	Shaqishi	3.8.1.40.	Jin Wei's Academy
3.8.1.18.	Chanshi	3.8.1.41.	Stone Keep
3.8.1.19.	Yinyoushi	3.8.1.42.	The Wall
3.8.1.20.	Ogo'Mukaima	3.8.1.43.	Lang Shi Hao's Military Academy
3.8.1.21.	Longshi	3.8.1.44.	Stone of Tongues
3.8.1.22.	Fa de Kong Magic	3.8.1.45.	Mercenary Company
3.8.1.23.	Fa de Shui Magic	3.8.1.46.	Peddler

3.9. **Attributes.** These data points represent characters' fitness in given areas and are divided into three groups to represent physical capabilities, affinity for social

interaction, and prowess of the mind. The character creation logic will determine the stats' starting values and maxima.

3.9.1. Physical Group

3.9.1.1. **Strength**

3.9.1.2. **Dexterity**

3.9.1.3. **Endurance**

3.9.2. Social Group

3.9.2.1. **Charisma**

3.9.2.2. **Appearance**

3.9.2.3. **Willpower**

3.9.3. Mental Group

3.9.3.1. **Intelligence**

3.9.3.2. **Awareness**

3.9.3.3. **Intuition**

3.10. **Virtues and Vices.** These vary by race, but each race has five fields that can range from 5 to -5. Each is initially 0, representing a character of neutral morality, though certain previous choices may modify these by default at the expense of some Virtue/Vice Points.

3.10.1. Human virtue/vice categories are:

3.10.1.1. **Courage/Cowardice**

3.10.1.2. **Loyalty/Betrayal**

3.10.1.3. **Modesty/Pride**

3.10.1.4. **Will/Weakness**

3.10.1.5. **Courtesy/Discourtesy**

3.11. **Skills.** There are 42 possible skills, each with numerical values ranging from 0 to 10, where 0 represents a character's complete inadequacy at a certain skill

while 10 represents complete mastery. All will begin as 0, though many will be altered by the program depending on choices made previously, and the player will be able to exchange Skill Points for an increase in the values of any skills.

The skills available are as follows.

3.11.1.1.	Artillery	3.11.1.22.	Politics
3.11.1.2.	Marksmanship	3.11.1.23.	Engineering
3.11.1.3.	Martial Arts	3.11.1.24.	Sorcery
3.11.1.4.	Strategy	3.11.1.25.	Architecture
3.11.1.5.	Tactics	3.11.1.26.	Linguistics
3.11.1.6.	Mounted Combat	3.11.1.27.	Medicine
3.11.1.7.	Vehicle Combat	3.11.1.28.	Surgery
3.11.1.8.	Animal Husbandry	3.11.1.29.	Mercantile
3.11.1.9.	Vehicle Operation	3.11.1.30.	Crafting
3.11.1.10.	Foraging	3.11.1.31.	Bureaucracy
3.11.1.11.	Resistance	3.11.1.32.	Artisan
3.11.1.12.	Apothecary	3.11.1.33.	Shady Dealing
3.11.1.13.	Athletics	3.11.1.34.	Gambling
3.11.1.14.	Stealth	3.11.1.35.	Economics
3.11.1.15.	History	3.11.1.36.	Performance
3.11.1.16.	Mathematics	3.11.1.37.	Socializing
3.11.1.17.	Philosophy	3.11.1.38.	Larceny
3.11.1.18.	Religion	3.11.1.39.	Leadership
3.11.1.19.	Occult	3.11.1.40.	Gaming
3.11.1.20.	Literature	3.11.1.41.	Manipulation
3.11.1.21.	Science	3.11.1.42.	Investigation

3.12. Styles and Techniques of Martial Arts. Depending on skill selections,

additional points will be granted at runtime to spend on martial arts sub-skills.

These points will be handled by the program and not stored with character data.

All of these martial arts techniques can range in value from 0 to 10, with the same proficiency-based representation used by the Skill data.

3.12.1.1.	Flame	3.12.1.9.	Break
3.12.1.2.	River	3.12.1.10.	Grapple
3.12.1.3.	Stone	3.12.1.11.	Block
3.12.1.4.	Wind	3.12.1.12.	Tumbling
3.12.1.5.	Mind	3.12.1.13.	Acrobatic
3.12.1.6.	Strike	3.12.1.14.	Meditative
3.12.1.7.	Throw	3.12.1.15.	Esoteric
3.12.1.8.	Lock		

3.13. Methods and Arts of Magic. Similar to the martial arts sub-skills above, points will be allocated by the program at runtime depending on Skill choices to spend on these sub-skills, which also range in value from 0 to 10 based on character proficiency.

3.13.1.1.	Creation	3.13.1.9.	Emotion
3.13.1.2.	Mutation	3.13.1.10.	Creature
3.13.1.3.	Shaping	3.13.1.11.	Plant
3.13.1.4.	Destruction	3.13.1.12.	Spirit
3.13.1.5.	Control	3.13.1.13.	Image
3.13.1.6.	Perception	3.13.1.14.	Time
3.13.1.7.	Mental	3.13.1.15.	Fate
3.13.1.8.	Element	3.13.1.16.	Dream

3.14. **Subjects and Fields of Academics.** Just like the previous two sections, these 0-10 values will be assigned using points given to the user at runtime, based on their previous skill selections.

3.14.1.1.	Humanities	3.14.1.7.	Rhetoric
3.14.1.2.	Sciences	3.14.1.8.	Demagoguery
3.14.1.3.	Societies	3.14.1.9.	Knowledge
3.14.1.4.	Law	3.14.1.10.	Research
3.14.1.5.	Military	3.14.1.11.	Empathy
3.14.1.6.	Logic		

3.15. **Aspects and Executions of Command.** As with the last three sections, these sub-skills are 0-10 values and are assigned using points, given to the user at runtime based on their skill selections.

3.15.1.1.	Henchmen	3.15.1.6.	Warfare
3.15.1.2.	Business	3.15.1.7.	Trade
3.15.1.3.	Military	3.15.1.8.	Espionage
3.15.1.4.	Information	3.15.1.9.	Administration
3.15.1.5.	Logistics	3.15.1.10.	Production

3.16. **Abilities.** These represent certain unusual things a character can do, apart from more conventional activities governed by skills and sub-skills. Characters have no progression of proficiency when it comes to abilities: they either possess an ability, or they do not.

3.16.1. Ability options are as follows. Listed first is the name of the ability, followed by the cost. Though Mastgrave has yet to flesh out the prerequisite

system he wants, the system needs to be modular enough to support one at some point in the future.

3.16.1.1.	Acute Reflexes, 6	3.16.1.28.	Marksman, 4
3.16.1.2.	Closed Gate, 5	3.16.1.29.	Sniper, 6
3.16.1.3.	Giant, 6	3.16.1.30.	Leading the Target, 8
3.16.1.4.	Demon Tongue, 0	3.16.1.31.	Brawler, 1
3.16.1.5.	Diminutive, 2	3.16.1.32.	Jumping Attacks, 2
3.16.1.6.	Intuitive Thinker, 2	3.16.1.33.	Martial Expertise, 2
3.16.1.7.	Genius, 3	3.16.1.34.	Overcoming the Weakness, 2
3.16.1.8.	Open Gate, 4	3.16.1.35.	Mind Over Body, 3
3.16.1.9.	Physically Gifted, 8	3.16.1.36.	Unarmored Fighter, 5
3.16.1.10.	Prodigy, 3	3.16.1.37.	Blind-Fight, 6
3.16.1.11.	Silver Tongue, 4	3.16.1.38.	Improvised Weapons, 6
3.16.1.12.	Phenom, 3	3.16.1.39.	Warrior's Hatred, 6
3.16.1.13.	Unnatural Senses, 3	3.16.1.40.	Seeing Without Eyes, 11
3.16.1.14.	Will to Power, 2	3.16.1.41.	Faithful Friend, 6
3.16.1.15.	Quick Learner, 7	3.16.1.42.	Fleet of Hoof, 3
3.16.1.16.	Child of the Dragon, 10	3.16.1.43.	Trick Riding, 4
3.16.1.17.	Magistrate, 2	3.16.1.44.	Trusted Mount, 4
3.16.1.18.	Battle Priest, 2	3.16.1.45.	Cover, 3
3.16.1.19.	Demagogue, 2	3.16.1.46.	Mobile Attacks, 4
3.16.1.20.	Warrior Poet, 2	3.16.1.47.	Moving Target, 6
3.16.1.21.	General, 2	3.16.1.48.	Overrun, 4
3.16.1.22.	Provost, 2	3.16.1.49.	The Wishing Horse, 5
3.16.1.23.	Close Combat Munitions, 5	3.16.1.50.	Animal Cunning, 7
3.16.1.24.	Rangefinder, 5	3.16.1.51.	Veterinarian, 3
3.16.1.25.	Quick Reload, 8	3.16.1.52.	Animal Whisperer, 4
3.16.1.26.	Unintended Accuracy, 5	3.16.1.53.	Animal Trainer, 4
3.16.1.27.	Lucky Shot, 4	3.16.1.54.	Gearhead, 8

- | | | | |
|------------|---------------------------------|-------------|----------------------------|
| 3.16.1.55. | Tricky Maneuvers, 5 | 3.16.1.85. | Probability of Prophets, 7 |
| 3.16.1.56. | Maintenance, 3 | 3.16.1.86. | True Alchemist, 7 |
| 3.16.1.57. | Just a Little Give, 4 | 3.16.1.87. | Well Read, 5 |
| 3.16.1.58. | Pathfinding, 5 | 3.16.1.88. | The Tides of History, 7 |
| 3.16.1.59. | Tracking in Terrain, 5 | 3.16.1.89. | Above Your Station, 7 |
| 3.16.1.60. | Scavenge, 5 | 3.16.1.90. | Skilled Researcher, 5 |
| 3.16.1.61. | Preparation, 4 | 3.16.1.91. | Dogmatic Fervor, 6 |
| 3.16.1.62. | Navigation, 4 | 3.16.1.92. | Virtue of a King, 5 |
| 3.16.1.63. | Sense of Direction, 7 | 3.16.1.93. | Vice Versus, 5 |
| 3.16.1.64. | Iron Gut, 5 | 3.16.1.94. | Blood Magic, 5 |
| 3.16.1.65. | Ice in the Veins, 7 | 3.16.1.95. | Special Favor, 7 |
| 3.16.1.66. | Fiery Blood, 6 | 3.16.1.96. | Creature Expert, 5 |
| 3.16.1.67. | Paloa Lungs, 6 | 3.16.1.97. | Dreamcaster, 4 |
| 3.16.1.68. | Heavy User, 4 | 3.16.1.98. | Dream Lord, 10 |
| 3.16.1.69. | Overexposure, 5 | 3.16.1.99. | Dreamwalker, 5 |
| 3.16.1.70. | Secret Ingredient, 7 | 3.16.1.100. | Magical Expertise, 2 |
| 3.16.1.71. | Every Good Cook, 7 | 3.16.1.101. | Proficiency, 6 |
| 3.16.1.72. | Just a Druggist, 5 | 3.16.1.102. | Legal Manipulations, 5 |
| 3.16.1.73. | Dancing on a Twig, 7 | 3.16.1.103. | Less Than Honest, 5 |
| 3.16.1.74. | Feats of Incredible Strength, 6 | 3.16.1.104. | Picking a Target, 5 |
| 3.16.1.75. | Natural Swimmer, 5 | 3.16.1.105. | Focus, 5 |
| 3.16.1.76. | Freerunning, 4 | 3.16.1.106. | Bargainer, 5 |
| 3.16.1.77. | Oiled Leathers, 5 | 3.16.1.107. | Triage Care, 3 |
| 3.16.1.78. | Master of Disguise, 6 | 3.16.1.108. | Eye for Quality, 3 |
| 3.16.1.79. | Hidden Blades, 4 | 3.16.1.109. | Advertising, 5 |
| 3.16.1.80. | Secret Pockets, 2 | 3.16.1.110. | Lay of the Land, 4 |
| 3.16.1.81. | Academic Expertise, 2 | 3.16.1.111. | Business as Usual, 4 |
| 3.16.1.82. | Refined Mixture, 6 | 3.16.1.112. | Master of Item, 5 |
| 3.16.1.83. | A Head for Numbers, 4 | 3.16.1.113. | Master Craftsman, 9 |
| 3.16.1.84. | Mad Science, 5 | 3.16.1.114. | Invoking the Muse, 6 |

3.16.1.115.	Improvised Construction, 6	3.16.1.132.	A Player's Player, 8
3.16.1.116.	Greater Improvised Construction, 10	3.16.1.133.	Copyist, 6
3.16.1.117.	Hedging the Bet, 5	3.16.1.134.	Fence, 2
3.16.1.118.	Shaved Knuckle in the Hole, 4	3.16.1.135.	Light Touch, 5
3.16.1.119.	Rigging the Game, 7	3.16.1.136.	Swift Fingers, 5
3.16.1.120.	Inside Information, 7	3.16.1.137.	Bragging Rights, 4
3.16.1.121.	Friends in Low Places, 6	3.16.1.138.	Trash Talk, 0
3.16.1.122.	Protection, 7	3.16.1.139.	Command Experience, 2
3.16.1.123.	Major Players, 3	3.16.1.140.	Inspiring a Crowd, 4
3.16.1.124.	Subject Research, 6	3.16.1.141.	Mob Rule, 7
3.16.1.125.	Choosing the Bit, 3	3.16.1.142.	Riding the High, 3
3.16.1.126.	Crowd Pleaser, 4	3.16.1.143.	Delegation, 1
3.16.1.127.	Friends in High Places, 7	3.16.1.144.	Risk Management, 0
3.16.1.128.	Playing the Field, 6	3.16.1.145.	Proliferous Contacts, 6
3.16.1.129.	Manners for Any Table, 5	3.16.1.146.	Treasure Hound, 6
3.16.1.130.	The Trusted Servant, 4	3.16.1.147.	Knowing the Game, 8
3.16.1.131.	Snake-oil Salesman, 7	3.16.1.148.	The World at a Glance, 10

4. Functional Requirements

4.1. File Storage

4.1.1. The system will store character data with an “.avnul” file type.

4.1.2. “.avnul” files will be delimited by colons (:).

4.1.2.1. All text boxes will prohibit entry of colon character for that reason.

4.1.3. “.avnul” files will be written and read by the program using the same order of data with which the character class is implemented. (See Design Document section 1.2).

4.2. Startup

4.2.1. The system will prompt the user to see if they want to open a previously-created character for review.

4.2.1.1. If so, a Windows file dialog is opened to the program directory, set to search for “.avnul” files.

4.2.1.1.1. A selected file is then read into the system and instantiated as a character class, whose data is then printed in a concise sheet.

4.2.1.1.2. The user may then exit the program entirely, or return to the main startup screen (section 4.2.1).

4.2.1.2. If the user doesn't wish to open a file, proceed to character creation.

4.3. Character Creation

4.3.1. Throughout character creation, available Floating Points will be displayed, with an option on each screen to determine their current exchange value.

4.3.2. Throughout character creation, an option must be available to see an “up to this point” character sheet to review previous choices.

4.3.3. The program will first accept user input for their character's Concept, Race, Name, Age, and Gender.

4.3.3.1. Depending on Race and Age, it will automatically assign an Age Category.

4.3.4. Next, the program will display a list of possible Backgrounds for the user to select, with the possibility of spending Floating Points and Background Points on more than one Background.

- 4.3.5. The program will then display a list of valid Houses, from which the user may select one and only one, having its cost deducted from the Floating Point total. In future builds, this must discriminate by race; for now, it must show only Longminjong Houses and not Bloodlines.
- 4.3.6. The program will display a list of available Dedications, from which the user may purchase any number with available Floating Points before moving onward.
- 4.3.7. Next, the program will display all 9 Attributes' current values, with a running tally of available Attribute Points that the player may distribute to increase the values of the Attributes. The user must assign all available Attribute Points before progressing.
- 4.3.8. The program will then display the counter for the user's available Vice/Virtue Points, and allow them to be distributed to alter the displayed values of all five vices and virtues. The user must assign all Available Vice/Virtue Points before progressing.
- 4.3.9. All 42 Skills will be displayed. A counter will be displayed of all available Skill Points, which will be decremented as the user assigns them to increase the values of Skills. The user must assign all available Skill Points before progressing.
- 4.3.10. If requisite Skills are at sufficiently high values, display all Styles and Techniques of Martial Arts, and calculate the available Temporary Points that the user's Skill selections have earned to distribute among them. Prohibit the user from progressing until all Temporary Points are assigned.

- 4.3.11. If requisite Skills are at sufficiently high values, display all Methods and Arts of Magic, and calculate the available Temporary Points that the user's Skill selections have earned to distribute among them. Prohibit the user from progressing until all Temporary Points are assigned.
- 4.3.12. If requisite Skills are at sufficiently high values, display all Subjects and Fields of Academics, and calculate the available Temporary Points that the user's Skill selections have earned to distribute among them. Prohibit the user from progressing until all Temporary Points are assigned.
- 4.3.13. If requisite Skills are at sufficiently high values, display all Aspects and Executions of Command, and calculate the available Temporary Points that the user's Skill selections have earned to distribute among them. Prohibit the user from progressing until all Temporary Points are assigned.
- 4.3.14. The program will then display a list of Ability options and their costs, display the available Ability Points, and decrement from it appropriately for each Ability purchased. The program shall prohibit the purchase of any ability whose cost is greater than the currently available amount of Ability Points.
- 4.3.15. At this point, the character is complete. Print the data in character-sheet format for review.

4.4. Exit

- 4.4.1. Prompt the user if they wish to save the character they have created.

- 4.4.1.1. If the user does wish to save, form a filename based on the character's given Name value, and check to see if that file already exists.
 - 4.4.1.1.1. If the file does exist, the user must be prompted to cancel saving, or enter a new name for their character. If a new name is entered, return to section 4.4.1.1.
 - 4.4.1.1.2. If the file does not exist, create it and save all recorded character data for later loading.
- 4.4.1.2. If the user does not wish to save, simply exit the program and let collected data be deallocated.

Design Specifications

1. Data Storage

1.1. Program Point Counters and Option Storage

1.1.1. The variety of “points” defined in section 2 of the Requirements

Document, used by the program to limit certain character choices, will all be stored as int values within the logic of the program itself. Their initial values are determined by their function: Floating Points start at 20, Background Points at 0, Attribute Points at 18, Virtue/Vice Points at 10, Skill Points at 64, and Ability Points at 10. Sub-skills do not have a set starting point value; in these cases, a temporary int variable will be used and its starting value computed at runtime based on previous choices.

1.1.2. The program will maintain three structure types to represent Bloodlines, Abilities, and Backgrounds. The Bloodlines struct will store a string value for the name of the House or Bloodline, and an int value to represent its point cost. Similarly, the Abilities struct will store a string for the ability name and an int for its cost. The Background struct will contain a string value for the background’s name, an int value for the background’s point cost, and an enumeration representing its specialization. The specialization enumeration will include, in alphabetical order, “Academics,” “Business,” “Expert,” “Social,” “Travel,” and “War.”

1.1.3. The aforementioned structure types will be used to create several arrays, which are hard-coded to store viable options for Bloodlines, Abilities, and Backgrounds. Each will store its corresponding structure type, and will be

initialized to hold all the available values listed in Requirements Document sections 3.6.1, 3.7.1, and 3.16.1.

- 1.1.4. One more array of type string will store Dedications. Its indices will be hard-coded to correspond to the list of possibilities given in Requirements Document section 3.8.1.

1.2. Character Class Representation

As stated in the requirements specification, the system will implement a sizeable class structure to contain all of the individual points of data associated with a character. In addition to its basic data types, such as ints, strings, and Boolean values, this class will also contain a variety of enums and structs, which will be defined externally to the class for reference by both the main program and by the character class.

- 1.2.1. Strings are used to hold data that will not be used for any form of calculation, to give users maximum flexibility. The characters' Concept and Name values will be stored as strings.
- 1.2.2. Only a single Boolean value is used: gender is stored as a bool, with 0 representing male and 1 representing female.
- 1.2.3. Two enumerations are used within the character class representation, as well. Race is stored as an enumeration, with the value of 1 representing a Longminjong. This allows maximum flexibility to the class when Mastgrave decides to add new races to his game rules. The age "category" defined in Requirements Document section 3.4.1 is also stored as an enumeration, with 0 representing Not Valid, 1 representing Child, 2 representing Adolescent, 3

representing Adult, 4 representing Middle Age, 5 representing Elder and 6 representing Ancient.

1.2.4. Within the character class, most data points are of type unsigned int. Age is one such, and due to design constraints will only have coded restrictions against values below 5. Bloodlines and Dedications will be stored as unsigned ints that correspond to values given in Requirements Document sections 3.7.1 and 3.8.1, respectively, meaning that their respective maxima will be 20 and 46. Characters' 9 Attributes will be represented as unsigned ints as well, though these will not have hard-coded range limitations aside from the inherent inability to be negative. Finally, every single skill and sub-skill will be represented by an unsigned int variable with valid range from 0 to 10.

1.2.5. The virtue/vice data are represented as *signed* ints, with accepted ranges from -5 to 5.

1.2.6. Lastly, the character class employs a pair of linked lists. The first consists of pointers of type Background, defined above, and will contain constant-class references to elements of the main program's array of Background options. Similarly, the second contains constant pointers of type Ability, which reference the elements in the main program's array of Ability options.

1.3. Functional Design Elements

1.3.1. Random Number Generation

1.3.1.1. In order to imitate the function of ten-sided dice, as specified in the requirements, the program will feature a utility function called

DiceRoll. This function will return an integer variable. The function will not take any arguments, but will utilize the C# library's randomization function to generate a random number, which will be processed and returned as a value ranging from 1 through 10. In its current state, Mastgrave's character generation rules do not use this function. He does have plans, however, to incorporate its use in future iterations of the game, necessitating its inclusion.

1.3.1.2. In order to simulate multiple dice rolls, the program will call this function several times, the amount of which will be determined at runtime based on player choices and executed with a for-loop. Successive rolls will be stored into an array of integers, and then ordered with a library-derived sort function from greatest to least. Any logic that allows low dice rolls to be discarded will simply alter the usable array bounds.

1.3.2. User Interface Constraints

1.3.2.1. In order to enforce the order of character creation, but to prevent the interface from becoming crowded with previously entered data, the system must utilize a different method to allow users to see what they have already entered. Every window in the graphical user interface will therefore feature a button, which will print a simple "character sheet" in a pop-up window at any point during development. This button will appear consistently in the interface on every screen.

- 1.3.2.2. When selecting values for virtues and vices, due to the limited range of values, a system of sliders will be used. The sliders will default to a value of 0 at the center of their mobile track. From there, they can be clicked and dragged to left to decrease the values until they reach the -5 minimum on the left extremity, or dragged to right to increase the values until the +5 maximum value is reached.
- 1.3.2.3. The general interface will also feature a “next” button, but it will be marked as unusable and “grayed out” by default. It will be enabled only when sufficient choices are made on a given screen to make the character.
- 1.3.2.4. The general interface will also feature a numerical counter, which will display the number of remaining floating points. Below this counter, where it is appropriate, will be displayed a button called “Convert,” which will show a pop-up confirmation window when clicked. This window will display what the user will gain for spending a given number of floating points, and will offer a “yes” and “no” button. If the choice is declined, nothing will change and the pop-up will close, but if the choice is accepted, both the floating point counter and other aspects of the interface will be updated to reflect the changes.
- 1.3.2.5. Where it is applicable, similar “counter” windows will show up for the points used in selecting skills, abilities, and other character data.
- 1.3.2.6. For any options that can be selected from a list, such as skills or abilities, a list box will be used. The program will run through the

selection of values in a given struct (such as the one containing skill options), and it will evaluate which ones are viable choices for a particular character. Only valid choices will be displayed in the list box. From there, the user can scroll through the list and double-click choices to select them.

1.3.2.7. For string-based data entry, a simple text box or rich text box will be provided for short or lengthy data items, respectively. Due to design constraints in section 1.3.3, the colon character (:) will be automatically removed from text boxes before data is stored.

1.3.2.8. Boolean values and other more limited choices (such as gender or race) will use radiobuttons to limit the available entry possibilities.

1.3.2.9. Once the user is finished with the character creation process, a final screen will offer the opportunity to display the “character sheet” data for review, and a button will allow users to save their character.

1.3.2.10. When the save button is clicked, no file dialog opens, but a filename will be generated in accordance with section 1.3.3. If this file doesn't exist, it will be created and all data appropriately stored. If it does already exist, the user will be shown a pop-up message with a textbox and a prompt to either change their character's name or cancel the save attempt.

1.3.3. File Storage

1.3.3.1. The file type for this saving system will be declared as “.avnul”

- 1.3.3.2. Files will be delimited by colons (:), which will require all text boxes in the program to disallow use of that character.
- 1.3.3.3. When a file is saved, the “character” class will be analyzed data item by data item in a particular order, and each value (or range of values) will be printed in colon-delimited form into the .avnul file.
- 1.3.3.4. When a file is loaded, the reverse occurs; an object of class character is created, the values are read in order from the file, and all of the values are passed in appropriate order to a specific constructor.

Testing Plan

Testing is perhaps the most overlooked aspect of the software life cycle. It is usually placed at the end of the process, and because it does not actually create a product, it is often given a relatively low budget by management entities. However, testing is an important part of the development of high-quality software, as it both validates correct code and allows any flaws to be detected and repaired.

Software tests are commonly divided into two categories, based on how they are performed. So-called “black box” tests treat the underlying code as an unknown entity, metaphorically placing it into an opaque black box. In these tests, requirements are used to test the program’s output based on provided input. On the other hand, “white box” testing allows the software to be tested based on the source code itself. Both forms of testing are of value to programmers, and should be used in conjunction for maximum effect.

Listed below are a series of proposed test cases for the first build of the character generation software. Pending changes to requirements and final realization of the code, additional tests may be added, especially to the White Box category.

1. Black Box Tests

1.1. Sequence test

1.1.1. Activate the character creation process.

1.1.2. Using dummy data, proceed through the process to its end.

1.1.3. Verify that every part of the character creation process is present, and in the proper order.

1.2. Random Number Generator test

- 1.2.1. Create a separate program as a testing harness.
 - 1.2.2. Create an array of 100 random numbers.
 - 1.2.3. Analyze distribution of numbers to ensure equal chance of all possibilities
- 1.3. Character Sheet Overview test.
- 1.3.1. At every screen during character creation, activate the function that shows character information thus far.
 - 1.3.2. Make sure all information up to that point is printed correctly.
 - 1.3.3. The final screen should show a full set of character information.
- 1.4. Save and Reload test
- 1.4.1. Progress through all character creation screens.
 - 1.4.2. Opt to save a file.
 - 1.4.3. Verify that the filename format is correct.
 - 1.4.4. Close the program.
 - 1.4.5. Reopen the program.
 - 1.4.6. Attempt to load the saved file.
 - 1.4.7. Verify, via character sheet, that all aspects of the character are preserved.
- 1.5. Expendable Points test
- 1.5.1. During creation process, verify that Floating Points start at 20.
 - 1.5.2. Verify that no Floating Points can be spent if current amount is less than the amount required for a given purchase.
 - 1.5.3. Verify that Background Points are automatically allocated and cannot be spent when their number is less than a given cost.

1.5.4. Verify that Attribute Points start at 18, and cannot be spent once their value reaches 0.

1.5.5. Verify that Virtue/Vice Points are initialized to 10 and cannot be spent below their 0 minimum.

1.5.6. Verify that Skill Points start at 64, and cannot be spent when a cost exceeds their current number.

1.5.7. Verify that 10 Ability Points are granted, and they cannot be spent when the cost is greater than their current value.

2. White Box Tests

2.1. Age Category test

2.1.1. During character creation, test ages 0, 4, 5, 10, 11, 16, 17, 35, 36, 50, 51, 65, and 66.

2.1.2. Verify that the ages are bound into their proper categories according to Requirements 3.4.1.1 through 3.4.1.7.

2.2. Background test

2.2.1. Analyze the data file containing all Background options detailed in Requirements 3.6.1.1 through 3.6.1.31.

2.2.2. Verify that each Background name is properly spelled.

2.2.3. Verify that each Background's cost is appropriate to its name.

2.2.4. Verify that each Background's specialization is correct.

2.3. Bloodlines and Houses test

2.3.1. Analyze the data file containing all Bloodline and House options detailed in Requirements 3.7.1.1 through 3.7.1.20.

2.3.2. Verify that each Bloodline or House has its name correctly spelled.

2.3.3. Verify that each Bloodline or House is associated with the correct cost.

2.3.4. Verify (during execution) that no Bloodlines are available, since the only race in this build is Longminjong.

2.4. Dedication test

2.4.1. Analyze the data file containing all Dedication options detailed in Requirements 3.8.1.1 through 3.8.1.46.

2.4.2. Verify that each Dedication is present and properly spelled.

2.5. Virtues and Vices test

2.5.1. Verify during execution that no Virtue or Vice value can be lowered past a value of -5.

2.5.2. Verify that no Virtue or Vice can be raised past a value of 5.

2.6. Skills test

2.6.1. Verify during execution that no Skill value can be lowered past a value of 0.

2.6.2. Verify during execution that no Skill value can be raised past a value of 10.

2.7. Abilities test

2.7.1. Analyze the data file containing all Ability options detailed in Requirements 3.16.1.1 through 3.16.1.148.

2.7.2. Verify that each Ability is present and properly spelled.

2.7.3. Verify that each Ability is associated with the proper cost.

User Documentation

The creation of software culminates in its delivery, for unless a program is released to its users it remains useless and is therefore a wasted effort. Because programs of any degree of complexity can be intimidating to those not in a computing profession, it is common practice to include simplified documentation with a released program. This documentation serves as a user manual, providing simple instructions to guide users through the functionality of the new program. The following section provides such instructions, divided into sections based on the task at hand.

1. Introduction

1.1. Avnul RPG

Welcome to Avnul, the setting of the works of Tobias Mastgrave! This roleplaying game will allow you to place yourself into this world and follow the narrative of your Lore Master to weave a story into the fabric of Avnul's history. (For more details on how the game is played, please review the manuals provided with the game, or ask your nearest Lore Master.)

1.2. What this program does

Of course, to participate in a roleplaying session you must first create a character. It is possible to do this with pen and paper, but keeping track of all the details can be quite daunting. This program is designed to make character creation easier for you; just follow the steps in the program, filling in information as you go, and you'll end up with a character that is legal according to the rules of the game. You can even save your character and review his or her details any time you like. Read on to learn how you can get started.

2. Starting out

2.1. Launching the program

Simply double-click the “Avnul Character Generator” icon to launch the program. Once it loads, the main menu will appear on your screen.

2.2. Main menu

From the main menu, you have several options in the form of three buttons; you can load a character you’ve already saved, you can create a new character, or you can close the program.

2.3. Closing the program

Whenever you finish, simply click the “Close” button on the main menu or click the “X” in the upper-right corner. Be aware that any character information that you haven’t saved will be lost upon exit.

3. Loading a character file

3.1. Selecting the file

Click the “Load” button on the main menu to open a Windows file dialog.

Navigate your computer to find your character file, which will be saved with the character’s name followed by “.avnul.” Click the file you wish to load, and then click “Open.” The program will load your character.

3.2. Viewing your character

Once your character’s information is loaded, all of his or her information will be printed to the screen. You can use the scroll bar on the right side of the viewing window to pan over the information. (Due to the complexity of character creation, saved character data cannot be altered.)

3.3. Exit

Simply click “Return” to close your character’s data file and return to the main menu. You can also close the program with the “X” in the upper-right corner.

3.4. Errors

If an error message appears while loading a character, something is wrong with your character file. This may be because the program was closed while in the middle of saving it, or because it was tampered with and the program can no longer read it. It is recommended to start over and create a new character in this case.

4. Creating a new character

4.1. Starting the process

Click “Create” on the main menu to start making your own Avnul RPG character. The following steps will guide you through the process.

4.2. Viewing your progress

At any time during the character creation process, you can click a button that says “Review” to check on your progress. This button will open a new window that shows all the information you’ve entered about your character so far. You can pan through this data with a scroll bar, if it is too long to fit in the text box. Simply click “Close” when you’re ready to go back to editing your character.

4.3. Floating points

Throughout the creation process, you’ll notice a counter at the top of the screen. This counter tells you how many Floating Points you have at your disposal; you will always start with 20. Throughout the program, you will have the option of

spending these points with a “Convert” button when appropriate. Clicking this button will show you a window that indicates how many points you spend, and what you can gain for them. If you want to make the expenditure, click “Confirm.” Clicking “Cancel” instead will close the window and leave your Floating Points untouched. Spend these points as you will to make your character stronger and more unique, but spend carefully; you can’t get them back!

4.4. Entering basic information

The first screen you will encounter asks you to enter some basic details. First is a text box labeled “Name.” Here, enter a name for your character. It doesn’t matter what name you choose. Next, a you’ll see a pull-down menu called “Race.” Click the menu, then click the race you wish to play from the list (at the moment, only the Longminjong (or human) race is playable). There is also a “Gender” menu; click the menu, and select either “Male” or “Female” from the list to determine the sex of your character.

The last item on this screen is a large text box labeled “Concept.” Here, you should enter a brief summary of your character to help you remember who he or she is. Some details on the character’s personality, history, motivations, or other traits are good ideas for what to write here.

When you’ve made a selection in every field, the “Next” button at the lower right corner will be enabled. When you finish entering basic information, click it to proceed to the next step.

4.5. Selecting Backgrounds

This screen features another point counter in addition to your Floating Points;

this one tracks Background Points, which you use to buy backgrounds for your character. You actually start with 0 points; this is because your first background selection is free (everyone must have a background). However, if you wish to purchase an additional background, you can trade some of your Floating Points for background points. The more points you trade, the more points you can earn, which can help you purchase more expensive backgrounds.

The actual background selection is simple. A menu on the left contains every choice available to you, and lists its cost. When you find one that you want, click it, then click the “Buy” button. It will disappear from the menu on the left, and appear in a box on the right side to show that you’ve bought it. If you decide to undo a purchase, click the Background in the right box, then click “Remove” to get rid of it. If you spent any Background Points on it, they will be refunded.

For a list of what each background entails, please consult the rule book that came with your game, or ask your Lore Master.

When you’ve chosen at least one Background, the “Next” button at the lower right corner will be enabled. When you finish selecting Background information, click it to proceed to the next step.

4.6. Selecting Houses

This screen will show you a list of Houses available to you. Houses represent familial descent for your character; Longminjong Houses are family organizations that come with different traits to help your character along. You don’t have to choose a House, and if you do, you may only choose one. Because Houses are optional, their cost is taken out of your Floating Point total, so be

cautious with your selection.

As with the last screen, available options and their costs are shown in a menu on the left side, with the right side holding whichever option you choose. Before leaving this screen, you may “Return” a House selection to refund its cost.

Please consult your rule book or your Lore Master for more information about what benefits the Houses of Avnul may offer you.

The “Next” button at the lower right corner is already enabled. When you finish with the House selection process, click it to proceed to the next step.

4.7. Selecting Dedications

Dedications are also optional choices you can make, which represent important choices and commitments your character has made in his or her life. Each has its own benefits and its own drawbacks; a full explanation is available in your rule book. Unlike Houses, you can have any number of Dedications, but their cost is deducted directly from your Floating Points.

Selection of Dedications is identical to the previous two screens: click the one(s) you want from the left menu, and click “Buy” to move them into the right side.

Clicking purchased items and clicking “Return” will undo the choice and refund the points spend on those items.

The “Next” button at the lower right corner is enabled by default. When you finish choosing Dedications, click it to proceed to the next step.

4.8. Assigning Attributes

The next screen will again display a secondary counter, this time for Attribute Points. These 10 points are used to purchase increases to your characters

Attributes, which represent his or her underlying character traits with numerical values. This is probably the most critical part of character creation, as it is these numbers that allow you to play your character in the roleplaying game. For a summary of what each statistic provides, please consult your rule book.

To increase an Attribute, click the up arrow next to the Attribute you wish to change. Every increase will deduct points from your Attribute Points counter; clicking the down arrow next to a stat will remove a point that you've placed there and refund its cost, but will not lower a stat below its starting value. You can also opt to trade in some Floating Points for additional Attribute Points here, if you so choose.

When you've spent all of your Attribute Points, the "Next" button at the lower right corner will be enabled. When you finish perfecting your Attribute choices, click it to proceed to the next step.

4.9. Vices and Virtues

Every character is slightly different from a moral perspective. This set of traits is represented by a set of Virtue and Vice sliders, which default to 0. Negative values indicate that your character has a Vice in that area, while positive numbers indicate a Virtue. You have ten Virtue/Vice Points, shown at the top of your screen. Clicking the + button next to a slider will increase its value (more virtuous), while clicking the – button next to a slider will decrease its value (more morally flawed). Moving a slider toward either extreme will deduct from your Virtue/Vice Points, while moving it back toward 0 will refund points spent there already.

You must spend all of your Virtue/Vice Points to move on. You cannot use any of your Floating Points here, because all characters are assumed to be equally flawed and virtuous in different ways.

When you've spent all your Virtue/Vice Points, the "Next" button at the lower right corner will be enabled. Click it to proceed to the next step.

4.10. Assigning Skills and Subskills

The next set of screens will be accompanied by a "Skill Points" counter, which starts at 64 and can be increased by spending some of your Floating Points.

These points are used to increase your characters Skills and Subskills over the next few screens (the amount of subskill screens you encounter varies based on what Skills you select.)

Your Skill options are diverse, with 42 choices available, each of which starts at 0 and can be raised as high as 10. Depending on how you want to play your character, you can raise certain skills to a high proficiency, or spread your points out over a greater number of skills. It costs more points to raise a skill to a higher level, so you'll have to spend more points to reach level 10 than you did to reach level 9 in a skill.

To increase a skill, simply click the + button next to it. To reclaim the points you spent on it, click the – button instead. When you have spent all of your Skill Points, the "Next" button will be enabled. Depending on your skill choices, you may have the option to purchase Subskill points in Martial Arts, Magic, Academics, and Command. These subskills will show up on their own screens if you've earned access, and are purchased with their own sets of Skill Points.

(Again, you can augment these points by spending your Floating Points on each screen).

4.11. Selecting Abilities

Once you've purchased all the Skills and Subskills you can, you'll be shown a menu of Abilities and their costs. Abilities represent special, powerful tricks that your character has access to, and they greatly affect how you play the roleplaying game. You start with 10 Ability Points shown at the top of the screen, which can be augmented by spending Floating Points (note that this is the last chance you have to use your Floating Points, so if you have any left, go ahead and use them.)

To select an ability, scroll the menu until you find the one you want. Click it, and click "Buy" to move it to the right-hand box. As with your choices earlier in the process, selecting a purchased Ability and clicking "Remove" will refund any points spent on it.

Instead of a "Next" button, you'll notice that this screen features a "Finish" button. Whenever you're done assigning Abilities, click "Finish" to complete your character.

4.12. Reviewing and saving your character

Once you click "Finish," a text box will show up to let you see the character you just made in full detail. Use the scroll bar on the right side of the box to see all of his or her various attributes.

If you decide you like what your character turned out to be, click the "Save" button. The program will automatically make a file in its own directory. When

you want to open it up later, look for your character's name followed by ".avnul." Once the file is saved, the program will return to the main menu.

If, instead, you dislike your final character or have no need to save it, simply click "Exit Without Saving." Your character data will be destroyed and you will be returned to the main menu of the program.

References

- AC Arcana (2013). RPG Character Generator (Version 1.1) [Software]. Available from <https://play.google.com/store/apps/details?id=com.acarcana.RPGCharacterBackgroundGenerator&hl=en>
- Appcelerator (2013). Titanium Development Platform [Software]. Available from <http://www.appcelerator.com/platform/>
- Deitel, P.J. (2005). Visual C# 2005: How to program. Upper Saddle River: Prentice Hall.
- Eques Nobilis LLC (2013). Character Sheets and Campaign Tools for RPGs. Retrieved from <http://www.charactersheet.net/>
- Fadeyev, D. (2009). 12 Useful Techniques for Good User Interface Design. *Smashing Magazine*. Retrieved from <http://uxdesign.smashingmagazine.com/2009/01/19/12-useful-techniques-for-good-user-interface-design-in-web-applications/>
- Fire Emblem Planet (2008). Spriter's Resource: Online Tools. [Software] Available from <http://www.feplanet.net/sprites-tools>
- In Focus (2013). C# Corner: Windows Forms Programming with C#. Retrieved from <http://www.c-sharpcorner.com/1/76/windows-forms-c-sharp.aspx>
- Krug, S. (2005). *Don't make me think: A common sense approach to web usability*. San Francisco: New Riders Press.
- Mastgrave, T. (2011). *Among the Neshilim*. Available from <http://www.smashwords.com/books/view/71735>
- Minion Development Corp. (2012). RPG Sheets: Role-Playing Game Character Sheet Archive. Available from <http://www.rpgsheets.com/>

Object Management Group, Inc. (2013). UML® Resource Page. Retrieved from
<http://www.uml.org/>

PCGen (2012). PCGen: An open source RPG character generator (Version 6.00)
[Software]. Available from http://pcgen.sourceforge.net/01_overview.php

Pfleeger, S.L. & Atlee, J.M. (2009). *Software engineering: Theory and practice* (4th ed.).
Upper Saddle River: Prentice Hall.

Smith, J. Third Edition (3.5) D&D Character Generator (Version 4.8) [Software].
Available from <http://dnd3rd.sourceforge.net/>

Steve Jackson Games. GURPS, Fourth Edition. Retrieved from
<http://www.sjgames.com/gurps/lite/>

Stream Labs Television Computer Systems (2013). Character Generation Program Alpha
pro (Version 2.0) [Software]. Available from [http://www.stream-
labs.com/products/software/alpha_pro/](http://www.stream-labs.com/products/software/alpha_pro/)

Tsui, F. & Karam, O. (2009). *Essentials of software engineering*. (2nd ed.). Burlington:
Jones and Bartlett.

Wizards of the Coast (2012). d20 System Archive. Retrieved from
<http://www.wizards.com/default.asp?x=d20/article/srdarchive>

Wizards of the Coast (2013). Dungeons & Dragons Character Builder (Version 1.0)
[Software]. Available from
<https://www.wizards.com/dnd/tool.aspx?x=dnd/4new/tool/characterbuilder>

Woolever, K.R. (2007). *Writing for the technical professions*. London: Longman.