**Final Project for Java Programming**

**By Robin G. Blaine**

**Synopsis**

This is a GUI program for creating and editing a character in the open-source, tabletop roleplaying game, Open Legend. It allows the user to enter information about their character and calculates and tracks resources for character creation (such as the number of available attribute points) and ensures that all choices are legal under the Open Legend rules.

**Code Example**

The following is a sample of the code for handling the button to decrease the value of a selected attribute. (The attribute in this case is Agility.)

// create handlers for increase/decrease attribute buttons

btDecreaseAttribute.setOnAction(e -> {

int value;

String currentAttributePoints;

String maximumAttributePoints;

if (rbAgility.isSelected()) {

value = characterSheet.getAgility();

if (value == 0) {

taDescription.setText("You cannot lower an attribute below 0...");

} else {

value -= 1;

characterSheet.setAgility(value);

currentAttributePoints = Integer.toString(characterSheet.getAttributePointsCurrent());

maximumAttributePoints = Integer.toString(characterSheet.getAttributePointsMaximum());

tfAttributePoints.setText(currentAttributePoints + " / " + maximumAttributePoints);

tfAgilityDice.setText(attributeDice[value]);

tfGuardDefense.setText(Integer.toString(characterSheet.getGuardDefense()));

taDescription.setText(AGILITY\_DESCRIPTION);

if (value == 0) {

tfAgilityCost.setText("");

tfAgilityValue.setText("-");

} else {

tfAgilityCost.setText(Integer.toString(attributeCost[value]));

tfAgilityValue.setText(Integer.toString(value));

}

}

**Motivation**

My motivations for this project go beyond merely completing a course assignment. I am an avid RPG gamer and was a backer of the Open Legend KickStarter. Established gaming systems (such as Dungeons & Dragons) have numerous character editor programs (and other resources) available. Open Legend, however, is new and does not have the same sorts of resources available. This program will help to fill that void.

**Installation**

Installation of the program only requires that the user run the OpenLegendCharacterEditor.java file.

**API Reference**

The upper portion of the display allows the user to type in the character’s and the player’s names.

The left side of the display allows the user to select which attribute to adjust via radio buttons and to adjust the selected attribute up or down via two buttons at the bottom of the attribute display. The number of current and maximum attribute points is displayed next to the buttons.

The program will not allow the user to increase any attribute beyond the starting maximum of 5 nor reduce any attribute below 0. An attribute can only be increased if sufficient attribute points remain to do so.

In the bottom-left corner of the display is a message box. This is a text area that displays information about the selected attribute as well as error messages if the user attempts an illegal adjustment of an attribute.

Next to the message box, is a text area that allows the user to type in a description of the character and a short biography.

The right side allows the user to enter a filename of an image to display that represents the character. The image file must be in the same folder as the program.

Below the image display are several text boxes displaying the character’s secondary attributes which are calculated from the primary attributes on the left side.

In the center of the display is a checklist for selecting feats (not fully implemented) and a text area to display the list of selected feats.

Below that are two text areas for selecting and displaying the character’s equipment (not fully implemented).

**Tests**

The attribute interface on the left-hand side is complete and works as intended. It is a simple matter to test this by selecting and increasing / decreasing attribute values using the “Increase +” and “Decrease –” buttons at the bottom.

**Contributors**

Individuals who wish to contribute to the project are free to do so. The program is incomplete and requires a great deal of work in the area of feats (entering individual feats including: the name and description of the feat, any prerequisites, and the feat’s effects) and equipment.

The implementation of the character image handler could use refinement, as well.