RPG Character Creator

Ethan Buchanan and Steven Garcia-Alamilla

Abstract

The RPG Character Creator is a tool that lets users create their own characters for RPG games, from Tabletop RPG enthusiasts to casual gamers. The project is close to a finished state and allows the user create a character from a predefined set of classes, races, backgrounds, and skills. The program allows you to save your characters so you can come back and look at your created character. It also allows you to edit and delete previously created characters.

1. Introduction

Character creation is a very important part of any RPG (Role Playing Game) experience. The RPG Character Creator is designed to provide an easy way to plan out future RPG journeys, whether you are an experienced RPG veteran or new to the RPG genre, the RPG Character Creator will provide a fun and engaging way to customize certain aspects of your character.While it is not necessary to plan out your character in advance, doing so allows stronger, more focused characters that can make your experience playing through the game better. While we can't provide a RPG character creator that covers every RPG system, we do hope our more generalized system allows the user to create a character that covers many different games. The tool can also be used not just for tabletop games, but can also be used to design a character for video games and even can be used for planning characters for something such as a novel.

1.1. Background

If you are unfamiliar with the RPG (Role-Playing Game) genre, there are a few things you should know. Firstly, RPGs allow you to play the role of a fictional character in a narrative-driven story. You are then given the ability to create and customize you character choosing who and what you want to play as. Some different attributes that you might be able to customize would include:

- Name, Biography, Age, Gender and Birthday
- Appearance of your character through AI generated portraits or import your own image
- Predefined Classes
- Predefined Races
- Predefined Backgrounds
- Choose your character's stats and skills

1.2. Impacts

While this project might not be impactful to everyone,we hope that the RPG Character Creator will have a major impact with people who enjoy playing RPG games. It would allow for users to plan out their character for their next RPG journey, which can enhance their overall experience.

1.3. Challenges

A few challenges that we anticipate through the course of this project include:

- Limitations of WPF
- Creating a GUI that makes it easy for the user to navigate through.
- Being able to finish the project on time.

A way we would overcome these challenges is by following the stages in the waterfall software development life cycle (SDLC). The SDLC makes sure that as we progress through the different phases of this project we keep in mind what has to be accomplished in order to move forward to the next stage.

As we wrap of development of the program outside of stretch goals, some challenges that came up when developing the application was updating the UI dynamically, getting finished in time, and not letting the project get too overwhelming. As the first demo day approached, we had little of the project done. We got a reasonable amount done for it, but before that deadline was set, we had problems getting started. There was always that question of "Where do we even start?", but we did some research of WPF applications and that allowed us to get a somewhat working UI out for the demo. A lot of this project did not have a lot of hard parts but many different small parts that required some work. We should of sat down and came up with a list of what races, classes, and backgrounds and had all of the information required for each done before it came time to implement them, but we ended up just winging it.

Use Case ID	Use Case Name	Primary Actor	Complexity	Priority
1	Create New Character	User	Med	1
2	Fill out Character Bio/Information	User	Low	2
3	Choose Character Portrait	User	Hard	3
4	Choose Character Class	User	Low	2
5	Choose Character Race	User	Low	2
6	Choose Character Background	User	Med	2
7	Choose Character Skills	User	Med	3
8	Choose Character Abilities	User	Med	3
9	Choose Character Alignment	User	Low	2
10	Finish Character on Overview Screen	User	Med	1
11	Edit Character and Delete Character	User	Med	4

TABLE 1. RPG CHARACTER CREATOR USE CASE TABLE

2. Scope

The completion of the project would mean that the user can create multiple RPG characters by customizing the character's Name, Portrait, Class, Race, Background, Stats, Skills and alignment. The RPG Character Creator would show the list of characters that have been created allowing for the creation/deletion of characters and even modifying characters after they have been created. A few stretch goals we have include:

- Different rule sets for different types of RPGs (example: Fallout/ DnD 5e)
- Allow for the user to toggle a DM mode which allows them to add custom skills, abilities, races, and classes to the character creator allowing for more customization of characters.

The implementation is done for everything outside of the designated stretch goals. There is still areas of improvement that need to be made for the base program, the main program mostly works bug-free.

2.1. Requirements

We want to make sure that our project meets the user's needs, so we looked at similar RPG character creator websites and looked at a variety of different video game character creators to understand what is required to create a fulfilling character creator that does not make the user want for more customization. We found that users prefer something that's easy to use, responsive, and gives the user many different options. Based on what we found users preferred, we created these requirements for our application.

2.1.1. Functional.

- The user should be able to create a new character.
- The user should be able to customize their character's stats, skills, abilities, choose from a large list of classes, choose from a wide variety of different races, and their character's background
- The user should be able to manage their characters by viewing, editing, and deleting them

2.1.2. Non-Functional.

- The application should be easy to use allowing the user to easily navigate through the GUI making it a more pleasant experience.
- Users will be able to use RPG Character creator without it feeling choppy or unresponsive.

2.2. Use Cases

Use Case Number: 1
Use Case Name: Create New Character
Description: The user starts the process for creating a new character.They will click on the "Create New Character" button this will take User to the character creator menu starting with the character information screen.
Basic flow for the process:

1) User navigates to the "Create new Character" button
2) User left-clicks on the button.
3) User is taken the character creation screen.

Termination Outcome: The user is now in the process of creating a new character. The user begins on the character info screen.
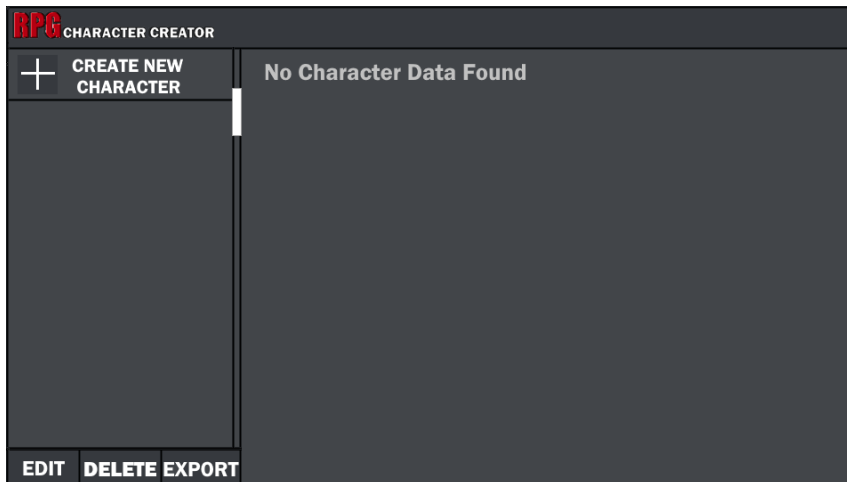
Use Case Number: 2

Figure 1. Use Case 1: What a user would see with no characters created.

Use Case Name: Fill Out Character Bio/Information
      Description: The user is now underway with creating their character. They have been taken to the first tab of character creation: the character information screen where they will fill in the Character's bio and basic information like name, age, gender, etc.
Basic flow for the process:
      1) User is now on Character Bio screen
      2) User can fill in each field for character info.
      3) User can fill in character's back story.
Termination Outcome: To leave this screen the user can click on the tabs at the top of the screen to go to a different category of character creation.

Figure 2. Use Case 2: What a user would see with no character info screen.

Use Case Number: 3
Use Case Name: Choose Character Portrait.
      Description: Once the user has reached this screen, they can choose a portrait that will represent their character.
Basic flow for the process:
      1) User is now on Character Portrait screen
      2) User can choose from an already collected group of portraits.
      3) User can import their own portrait for their character.

Termination Outcome: To leave this screen the user can click on the tabs at the top of the screen to go to a different category of character creation.

Use Case Number: 4

Use Case Name: Choose Character Class.

Description: Once the user has reached this screen, they can choose a class for their character.

Basic flow for the process:

1) User is now on Character Class screen
2) User can scroll up and down the list to see what class they want.
3) By clicking on a class it will select what class they want to be.
4) Clicking on a class also shows a description of what each class is as well as the class's primary and secondary stats, class skills, and class abilities.

Termination Outcome: To leave this screen the user can click on the tabs at the top of the screen to go to a different category of character creation.

Figure 3. Use Case 4: What a user would see on the Character Class screen

Use Case Number: 5

Use Case Name: Choose Character Race.

Description: Once the user has reached this screen, they can choose a race for their character.

Basic flow for the process:

1) User is now on Character Race screen
2) User can scroll up and down the list to see what race they want.
3) By clicking on a race it will select what race they want to be.
4) Clicking on a race also shows a description of what each race is as well as the race's traits.

Termination Outcome: To leave this screen the user can click on the tabs at the top of the screen to go to a different category of character creation.
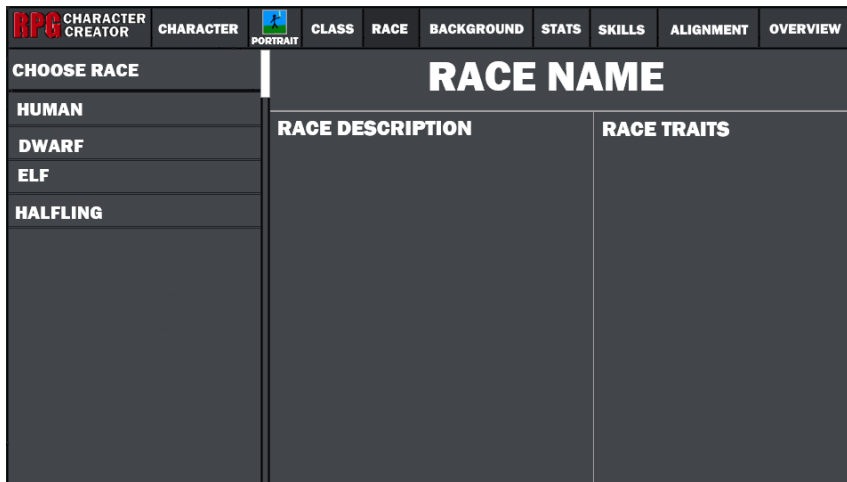
Figure 4. Use Case 5: What a user would see on the Character Race screen

Use Case Number: 6

Use Case Name: Choose Character Background.

Description: Once the user has reached this screen, they can choose a background for their character.

Basic flow for the process:

1) User is now on Character Background screen
2) User can scroll up and down the list to see what background they want.
3) By clicking on a background it will select what background they want to be.
4) Clicking on a race also shows a description of what each background is as well as what skills the background provides.

Termination Outcome: To leave this screen the user can click on the tabs at the top of the screen to go to a different category of character creation.

Use Case Number: 7

Use Case Name: Choose Character Skills.

Description: Once the user has reached this screen, they can choose what skills they want their character to have. There is universal skills and class specific skills.

Basic flow for the process:

1) User is now on Character Skills screen
2) User can scroll up and down the list to see what skills they want.
3) By clicking on a skill it will select what skills they want to add to their character, they can add multiple skills.
4) Clicking on a skill also shows a description of what each skill entails.

Termination Outcome: To leave this screen the user can click on the tabs at the top of the screen to go to a different category of character creation.

Use Case Number: 8

Use Case Name: Choose Character Abilities.

Description: Once the user has reached this screen, they can choose what abilities they want their character to have. There is universal abilities and class/race specific abilities.

Basic flow for the process:

1) User is now on Character Abilities screen
2) User can scroll up and down the list to see what abilities they want.
3) By clicking on a skill it will select what abilities they want to add to their character, they can add multiple abilities.
4) Clicking on a abilities also shows a description of what each abilities entails.

Termination Outcome: To leave this screen the user can click on the tabs at the top of the screen to go to a different category of character creation.

Use Case Number: 9

Use Case Name: Choose Character Alignment.

Description: Once the user has reached this screen, they can choose what abilities they want their character to have. There is universal abilities and class/race specific abilities.

Basic flow for the process:

    1) User is now on Character Alignment screen

    2) User can mouse over each alignment which will give them a description of what kind of character falls in to each alignment.

    3) Clicking on a alignment choose that alignment for their character.

Termination Outcome: To leave this screen the user can click on the tabs at the top of the screen to go to a different category of character creation. At this point if the user went in order, they would go to the Overview tab.

Use Case Number: 10

Use Case Name: Finish Character on Overview Screen

Description: Once the user has reached this screen, they are on the final section. This screen gives an overview of what they selected and they can finally add the character to the system.

Basic flow for the process:

    1) User is now on Character Overview screen.

    2) User can now see all of what they have chosen for their character.

    3) Clicking on "Finish Character Creation" will add the character to the main menu.

Termination Outcome: Once the user clicks the "Finish Character Creation" button they are taken to the main menu, exiting the character creation screen.

Figure 5. Use Case 10: What a user would see on the Character Overview screen

Use Case Number: 11

Use Case Name: Edit Character and delete character

Description: If there are characters in the system then the user can go back and edit their characters allowing them to go back through the character creation screen.

Basic flow for the process:

    1) User is on Main Menu screen

    2) User can click on character they want to select in left tab.

    3) User can then navigate to bottom left of screen and press the Edit button or delete button

    4) This will then take them back to the character creation menu or deletes the selected character.

Termination Outcome: Once the user has edited the character to how they want they will go back to the Overview screen and press "Finish character Creation" button again.

Use Case Number: 13

Use Case Name: Export Character

Description: If there are characters in the system that the user wants to export to a file then they can.

Basic flow for the process:

    1) User is on Main Menu screen

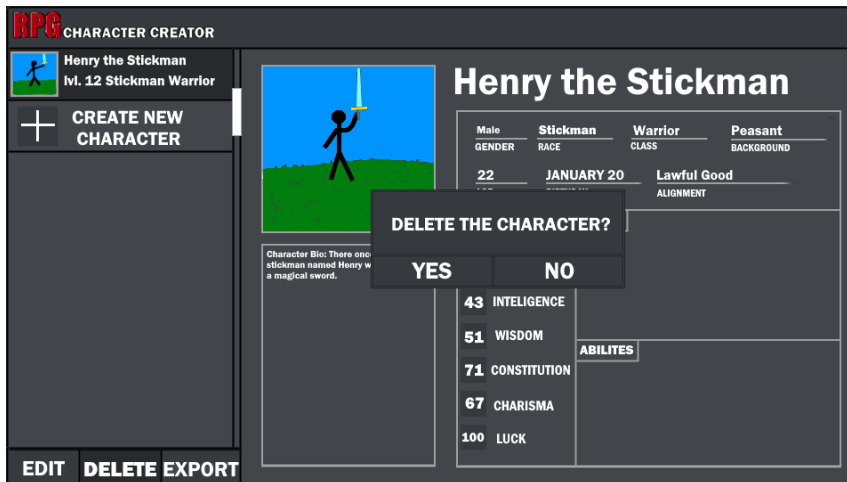   2) User can click on character they want to select in left tab.

Figure 6. Use Case 12: What a user would see on when deleting a character

3) User can then navigate to bottom left of screen and press the export button.

4) A menu showing where they want to export the character file to will show up and they choose where.

Termination Outcome: Once the user has chosen the location then, they character sheet should be saved to their computer.

2.3. Interface Mockups

Screenshots of what the final UI looks like. Once the program is started the user is met with an empty screen, with the only thing showing up is a message saying "No Character Data Found" and the create character button. If the user has started the program before then the saved characters should appear. Note, characters will only save if they have been finished, by selecting the finish character button on the Overview screen and the user exits the program through the X in the top right corner. Also, to get back to the main menu, the user needs to press on the RPG Character Creator logo.
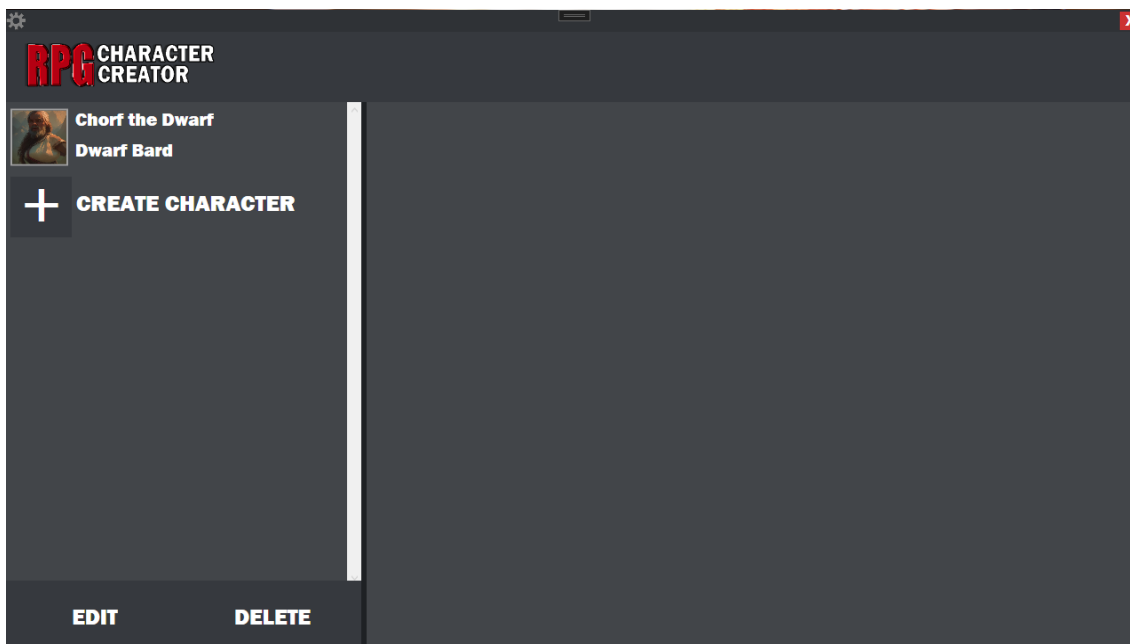
Figure 7. Main Screen, nothing selected

Figure 8. Main Screen, character selected

The user can select their characters on this page by clicking the button. Selected characters can be edited or deleted. Pressing the edit button take the user back through character creation.

Figure 9. Character Bio screen

The user inputs the character info into the text boxes. To continue character creation press any of the other tabs at the top of the screen. The user can come back to the character tab later if they need to.
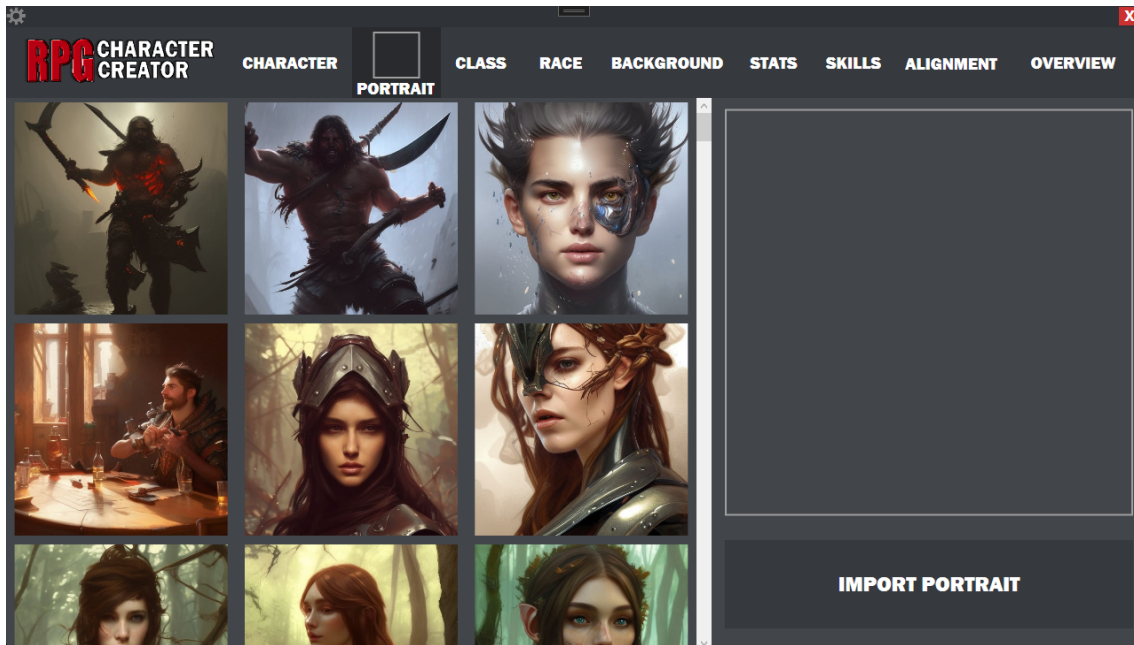
Figure 10. Portrait Screen

The user can select a portrait from the ones provided or import their own image by pressing the import portrait button.
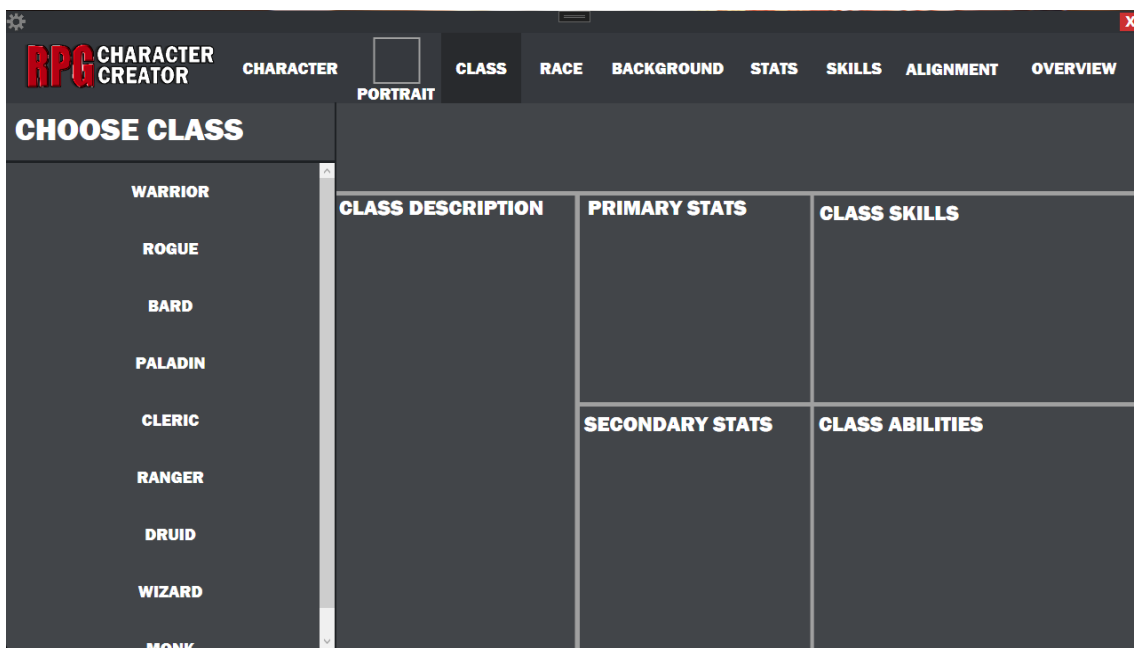
Figure 11. Class Screen

The user can select their class.
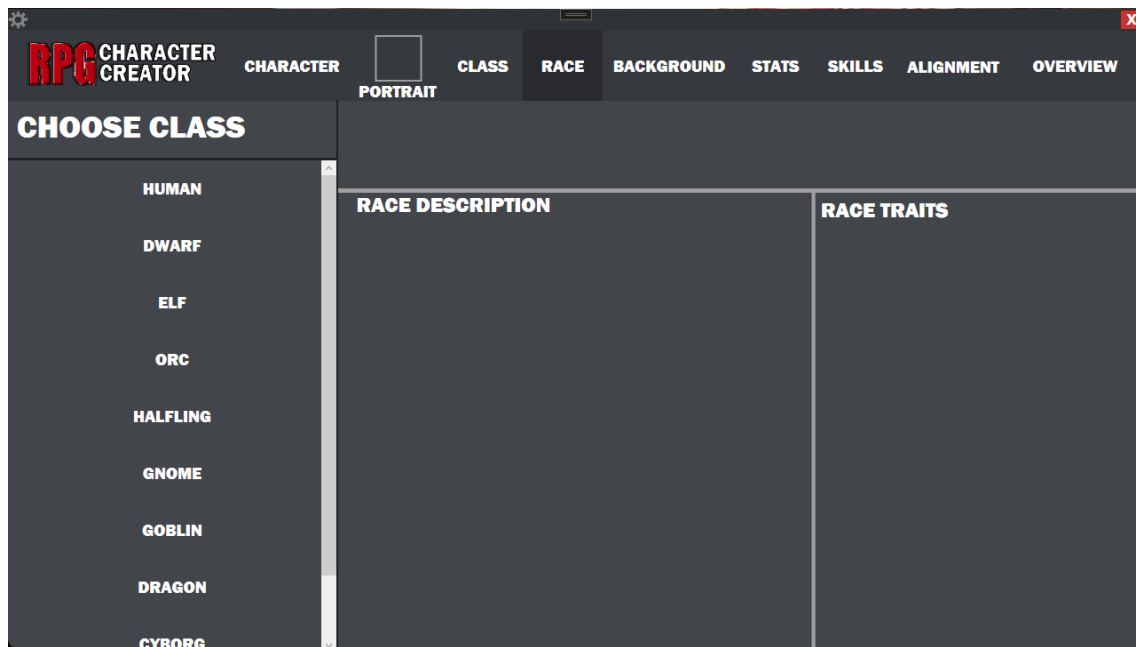
Figure 12. Race Screen

The user can select their race.
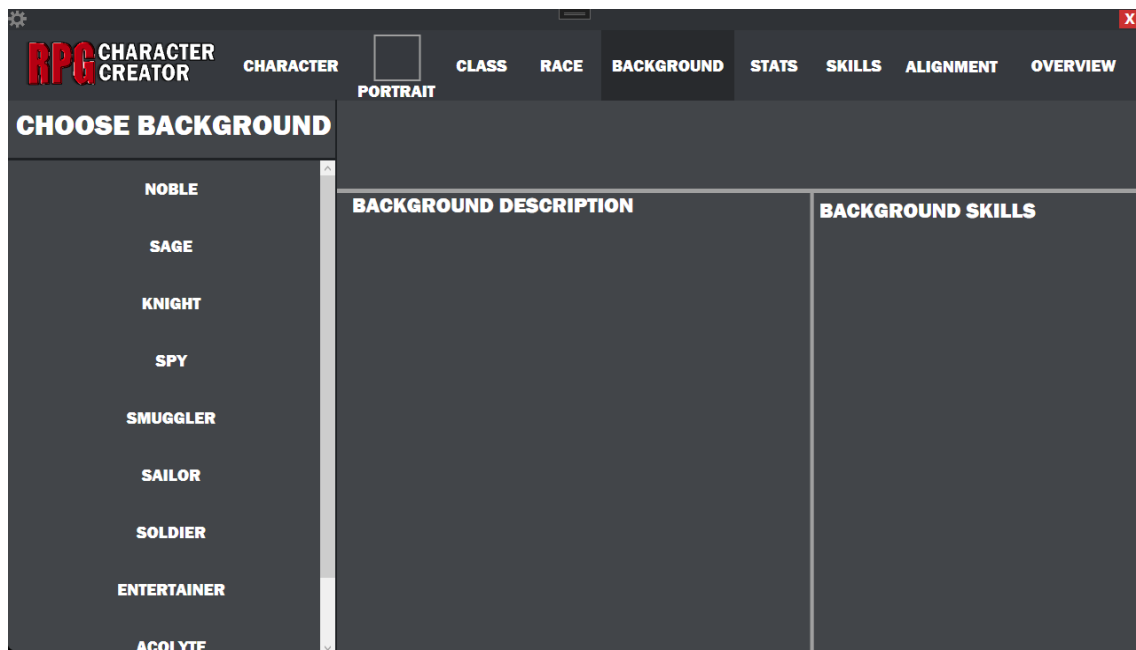
Figure 13. Background Screen

The user can select their background.

Figure 14. Stats Screen

The user can increase or decrease their stats.

Figure 15. Skills Screen

The skills from background are inherited, and the user can pick from the skills in the left panel.
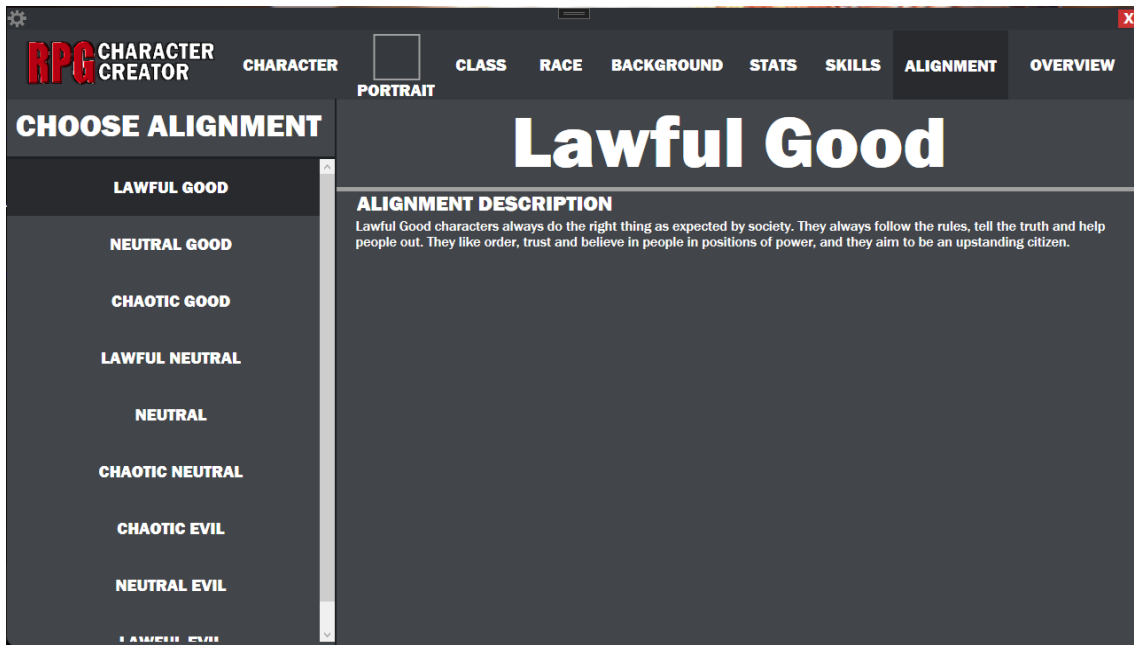
Figure 16. Alignment Screen

The user can choose their character's alignment.
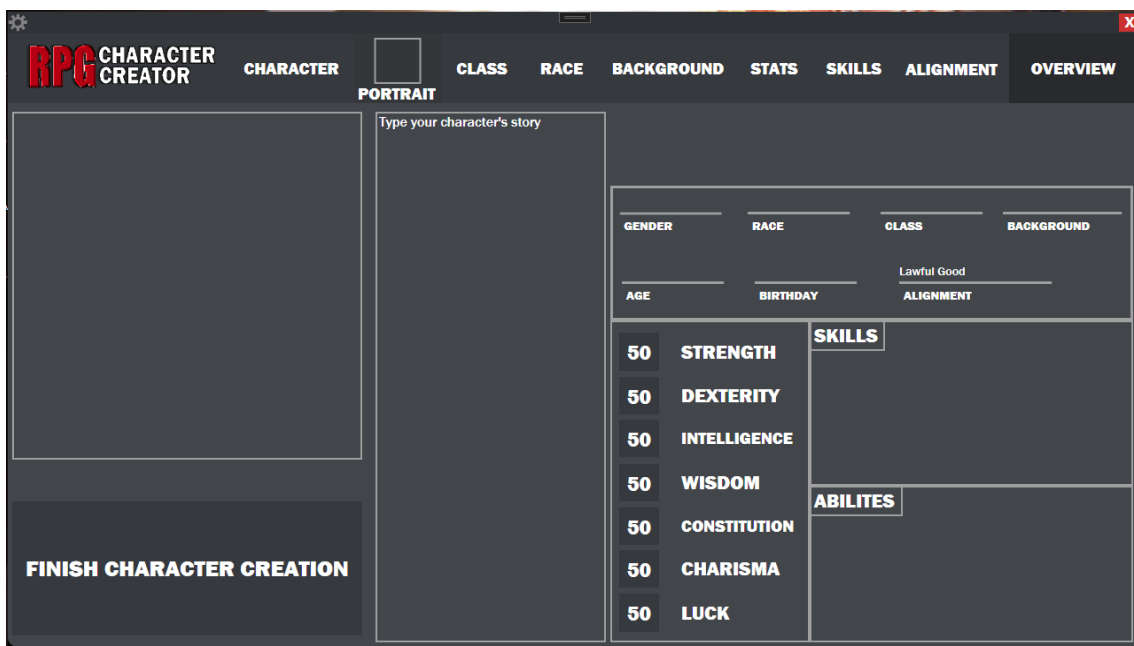
Figure 17. Overview Screen

The user can look over their character and then finish it by pressing the Finish Character Creation Button. This adds the character to the system and they can save it by exiting the program by pressing the x in the top right corner.

3. Project Timeline

Project Proposal Draft (February 04, 2023) - First, we brainstormed ideas for a project that would be worth pursuing, and then we drafted a project proposal. This proposal included a project name, an abstract, an introduction, and a scope.

Finally, we created a Github repository and uploaded the proposal to it.

Project Proposal Update (February 28, 2023) - The update to the project proposal involved identifying the use cases for the project, as well as defining its functional and non-functional requirements. In addition, several interface mockups were created to show our vision for the project.

GUI Day (March 9th, 2023) - Showcased our interface mockups to the class and gathered feedback on potential additions or modifications to our interface mockups.

Coding of the actual project began during these two periods beginning with the implementation of a basic UI and building upon it leading up to demo day.

Demo Day (April 6th, 2023) - Presented the current status of our project to our classmates and Dr. Ericson.

Project Update (April 10th, 2023) - We created a timeline to showcase the progression we have made so far, which involved adding project structure and outlining UML diagrams to project documentation.

Penultimate Writeup(April 21, 2023) - We plan to revisit our work and make necessary updates to our documentation to ensure that it is accurate and up-to-date by the specified time.

Presentation Draft (April 21, 2023) - We plan on finishing a draft for our presentation slides.

Final presentation(April 29, 2023) - We plan on finishing our project and giving our final presentation to the class.

Final Written Report (April 28, 203) - We plan on having our final written report that will contain all needed information to show what we have accomplished.

4. Project Structure

First, we needed to determine our target audience. In our case, we decided that this RPG character creator would be geared towards everyone, including both experienced and beginner RPG players. We wanted to make the process as smooth as possible for the user, so we decided on having an easy-to-navigate user interface. We designed the RPG character creator with several sections, including Character which includes the basic character information such as name, age, birthday, and gender. There is also a portrait section where users can choose from pre-selected images or upload their own. The race, background, and alignment section allows users to choose from multiple options, and their information is filled in accordingly. There is also a stats and skills section and an overview page that lets users review their character's details before finalizing the creation process.

In order to stay organized throughout the development stage, we have decided to implement the builder design pattern in our RPG character creator application. By using this pattern, we can easily create complex objects (Characters in our case) by building them step by step. The builder pattern offers the advantage of creating multiple types of objects using the same interface, which will be helpful for our application since users can create a variety of RPG characters.We also implemented the abstract factory design pattern which will be used to create different color themes for the application, easily allowing user to switch between different themes without affecting the functionality or structure of the application.

In addition to the design patterns, we have also implemented the Model-View-ViewModel (MVVM) architecture in our RPG character creator application. MVVM helps to cleanly separate the GUI and program logic allowing for better maintainability and testability.

4.1. UML Outline

The following UML is an Factory Design pattern see Figure 19

The BackgroundFactory is an abstract class that defines a factory method, 'CreateBackground()' , which returns a Background object.

The LightTheme and DarkTheme classes are concrete implementations of the BackgroundFactory class. Each class provides a different implementation of the CreateBackground() method, which creates a Background object that has a different color theme.

The Background class is a product of the BackgroundFactory class. It defines a set of properties and methods that allow clients to interact with the background object, such as getting and setting the background color.

Clients of the BackgroundFactory pattern use the factory method to create a Background object without having to know the implementation details of the factory or the product.
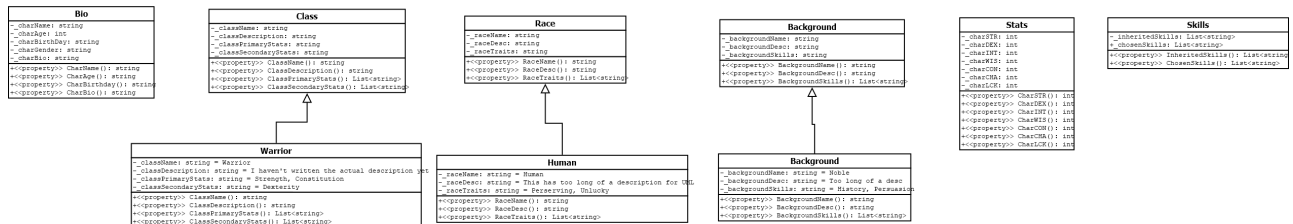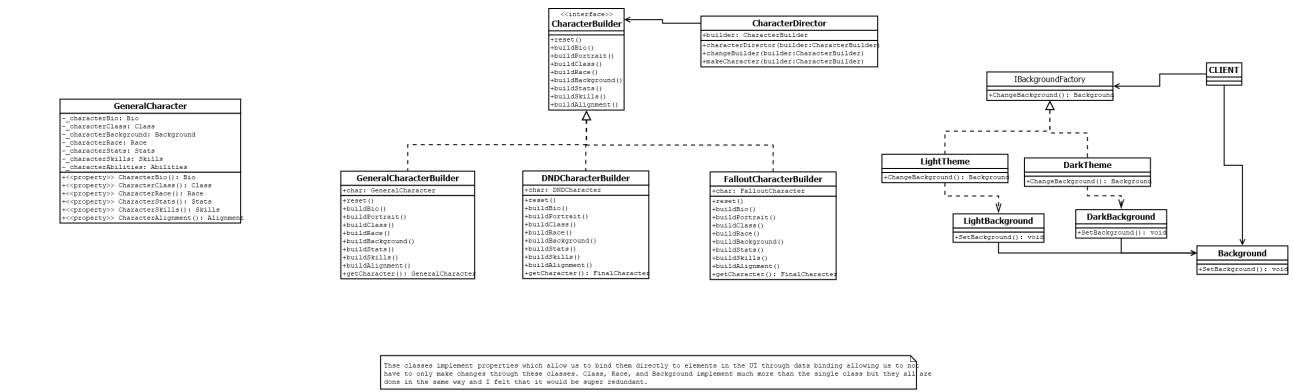
Figure 18. The entire UML for the project
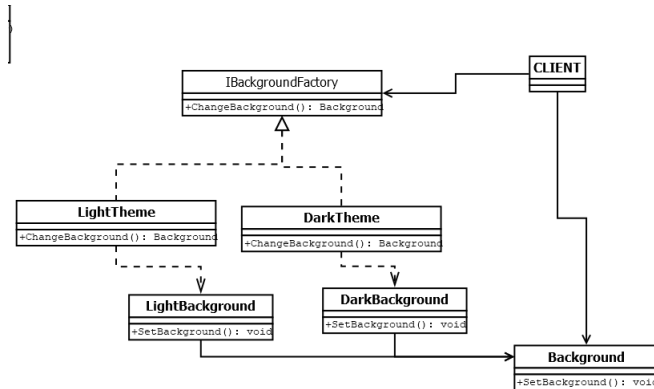
Figure 19. Possible implementation for a factory method that allows for us to easily change the theme of the UI.

The following UML is a Builder Design pattern see Figure 20.

The CharacterBuilder interface specifies the steps for building a character.

The GeneralCharacterBuilder class implements this interface and is responsible for building the character.

The CharacterDirector class instructs the GeneralCharacterBuilder on the steps to building a character. The GeneralCharacter object is the complex object, and the CharacterBuilder interface defines the steps required to construct it.

The GeneralCharacterBuilder class implements these steps in its methods.

The CharacterDirector class acts as a director, which looks over the construction process and can create a specific type of character by using a specific implementation of the CharacterBuilder interface.

4.2. Design Patterns Used

We used the builder design pattern which allows us to build a character all in one step. We first gather all the information need to create a character then call the builder through the director which creates a character with all the information that we need for the character. It can also create empty characters which the user can go back and edit. To implement different themes for the UI we implemented a factory method which allows us to easily create different themes. All of the UI elements

<<interface>>
CharacterBuilder

+reset()
+buildBio()
+buildPortrait()
+buildClass()
+buildRace()
+buildBackground()
+buildStats()
+buildSkills()
+buildAlignment()

CharacterDirector

+builder: CharacterBuilder

+characterDirector(builder:CharacterBuilder)
+changeBuilder(builder:CharacterBuilder)
+makeCharacter(builder:CharacterBuilder)

GeneralCharacter

-_characterBio: Bio
-_characterClass: Class
-_characterBackground: Background
-_characterRace: Race
-_characterStats: Stats
-_characterSkills: Skills
-_characterAbilities: Abilities

+<<property>> CharacterBio(): Bio
+<<property>> CharacterClass(): Class
+<<property>> CharacterRace(): Race
+<<property>> CharacterStats(): Stats
+<<property>> CharacterSkills(): Skills
+<<property>> CharacterAlignment(): Alignment

GeneralCharacterBuilder

+char: GeneralCharacter

+reset()
+buildBio()
+buildPortrait()
+buildClass()
+buildRace()
+buildBackground()
+buildStats()
+buildSkills()
+buildAlignment()
+getCharacter(): GeneralCharacter

DNDCharacterBuilder

+char: DNDCharacter

+reset()
+buildBio()
+buildPortrait()
+buildClass()
+buildRace()
+buildBackground()
+buildStats()
+buildSkills()
+buildAlignment()
+getCharacter(): FinalCharacter

FalloutCharacterBuilder

+char: FalloutCharacter

+reset()
+buildBio()
+buildPortrait()
+buildClass()
+buildRace()
+buildBackground()
+buildStats()
+buildSkills()
+buildAlignment()
+getCharacter(): FinalCharacter

Figure 20. The Builder of the project, the generalCharacterBuilder creates a general character and through the director builds a generalCharacter with each of its variables being assigned through the builder.

These classes implement properties which allow us to bind them directly to elements in the UI through data binding allowing us to not have to only make changes through these classes. Class, Race, and Background implement much more than the single class but they all are done in the same way and I felt that it would be super redundant.

Bio

-_charName: string
-_charAge: int
-_charBirthDay: string
-_charGender: string
-_charBio: string

+<<property>> CharName(): string
+<<property>> CharAge(): string
+<<property>> CharBirthday(): string
+<<property>> CharBio(): string

Class

-_className: string
-_classDescription: string
-_classPrimaryStats: string
-_classSecondaryStats: string

+<<property>> ClassName(): string
+<<property>> ClassDescription(): string
+<<property>> ClassPrimaryStats(): List<string>
+<<property>> ClassSecondaryStats(): List<string>

Race

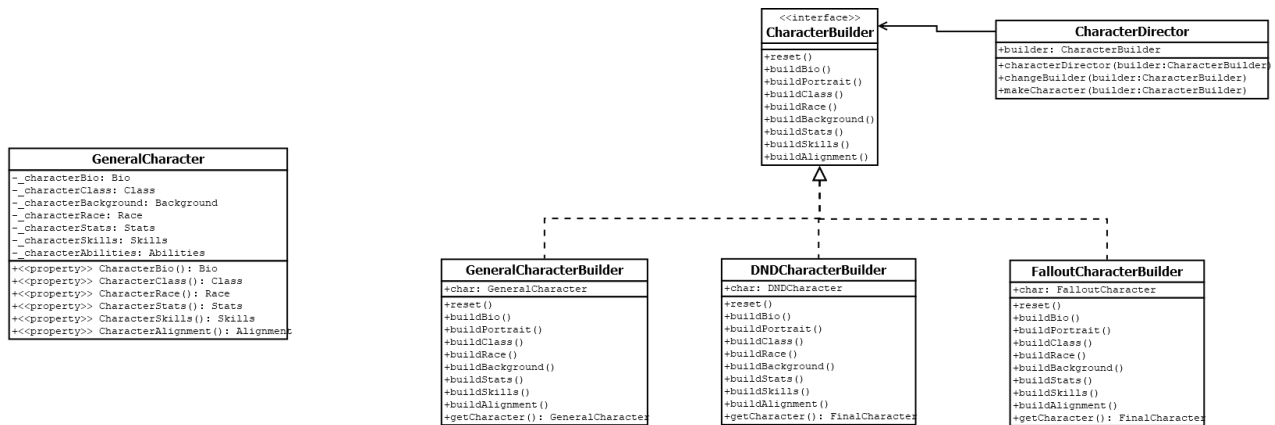-_raceName: string
-_raceDesc: string
-_raceTraits: string

+<<property>> RaceName(): string
+<<property>> RaceDesc(): string
+<<property>> RaceTraits(): List<string>

Background

-_backgroundName: string
-_backgroundDesc: string
-_backgroundSkills: string

+<<property>> BackgroundName(): string
+<<property>> BackgroundDesc(): string
+<<property>> BackgroundSkills(): List<string>

Stats

-_charSTR: int
-_charDEX: int
-_charINT: int
-_charWIS: int
-_charCON: int
-_charCHA: int
-_charLCK: int

+<<property>> CharSTR(): int
+<<property>> CharDEX(): int
+<<property>> CharINT(): int
+<<property>> CharWIS(): int
+<<property>> CharCON(): int
+<<property>> CharCHA(): int
+<<property>> CharLCK(): int

Skills

-_inheritedSkills: List<string>
-_chosenSkills: List<string>

+<<property>> InheritedSkills(): List<string>
+<<property>> ChosenSkills(): List<string>

Warrior

-_className: string = Warrior
-_classDescription: string = I haven't written the actual description yet
-_classPrimaryStats: string = Strength, Constitution
-_classSecondaryStats: string = Dexterity

+<<property>> ClassName(): string
+<<property>> ClassDescription(): string
+<<property>> ClassPrimaryStats(): List<string>
+<<property>> ClassSecondaryStats(): List<string>

Human

-_raceName: string = Human
-_raceDesc: string = This has too long of a description for UML
-_raceTraits: string = Perserving, Unlucky

+<<property>> RaceName(): string
+<<property>> RaceDesc(): string
+<<property>> RaceTraits(): List<string>

Background

-_backgroundName: string = Noble
-_backgroundDesc: string = Too long of a desc
-_backgroundSkills: string = History, Persuasion

+<<property>> BackgroundName(): string
+<<property>> BackgroundDesc(): string
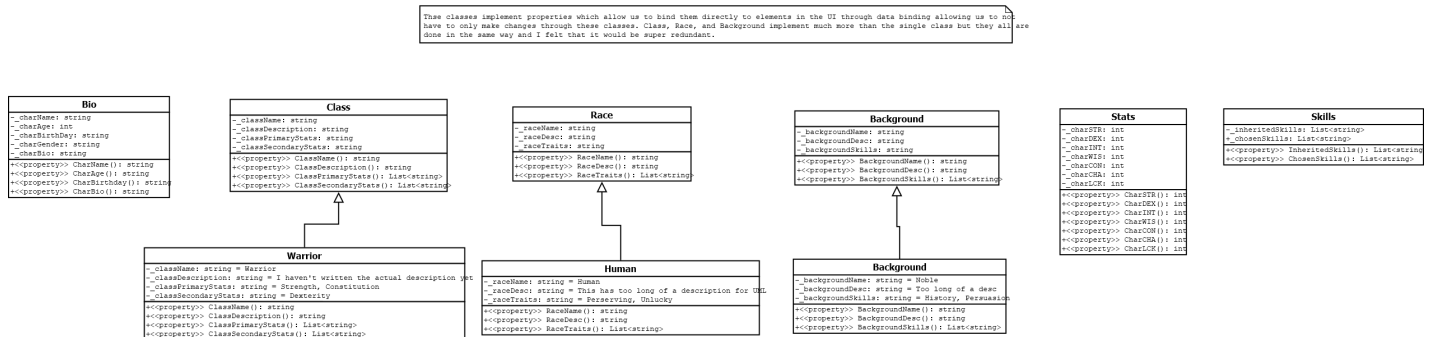+<<property>> BackgroundSkills(): List<string>

Figure 21. The generalCharacter is made of these classes, these classes store information and allow for us to easily update the UI through the use of data binding and properties.

are bound to different variables and the factory method assigns these variables different color values. This allows for us to add many different themes and if we need to, edit already create themes.

5. Results

Outside of the stretch goals, I think that the project ended up where we wanted it to when we first came up with the idea. Right now its scope is somewhat limited but for the most part it covers the most commonly used characteristics that define traditional RPG characters. The framework is there for being able to implement a means of creating custom classes, races, backgrounds, etc. which would greatly increase the applicability of the application. While it is done in the sense that it works like we first planned it to, there is still parts of the project that need some "love" added to them. Quality of life(QoL) features such search bars, sorting, tool tips are definitely missing from the project. Overall, we are happy with what we were able to accomplish with one of first major projects.

5.1. Future Work

In the future, like mentioned in the results section, there is a need for some QoL features and some love to go towards the more neglected parts of the project. While the implementation of different RPG systems would require a major rework of how the current UI is implemented, adding a DM mode would not. This DM mode would allow for the user them self to cover any areas of character creation that we might of missed allowing for them to get their character exactly how they want them.

References

[1] H. Kopka and P. W. Daly, *A Guide to LATEX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.