Nicholas Siconolfi 5/7/19

[CMPT\_220L\_203\_19S](https://ilearn.marist.edu/portal/site/27ad7ce2-ae28-4f6c-94e6-06eb4603a545/page-reset/4e468029-dd8e-46e9-912a-3b756571648f)

Professor Arias

Java Text-Based Role Playing Game

The Text-Based Role Playing game that I have developed as my final project has progressed very smoothly and is finally at its conclusion. This project has been interesting and fun to develop as I have been going through new strategies and ways to create and run the program as best as I see fit. The process of creating this program is very time consuming as well since much thought and effort must be included in order to avoid any types of mistakes and runs thoroughly without any problems as well. This project has been fun creating as well since I am a big fan of role playing games and creating one feels like I am able to control the outcomes and creating one feels like a dream, even though it is not as complex as the ones created by developers since it is only text-based, the effort is well shown.

Text-based games have been well documented since at least the 1960s, when [teleprinters](https://en.wikipedia.org/wiki/Teleprinter) were interlaced with [mainframe computers](https://en.wikipedia.org/wiki/Mainframe_computer) as a form of input, where the output was printed on paper. With that, notable titles were developed for those computers using the teleprinter in the 1960s and 1970s, and numerous more have been developed for video terminals since at least the mid-1970s, having reached their peak popularity in that decade

and the 1980s, and continued as early [online games](https://en.wikipedia.org/wiki/Online_games) into the mid-1990s. Although generally replaced in favor of video games that utilize non-textual graphics, text-based games continue to be written by independent developers. They have been the basis of instigating genres of video gaming, especially [adventure](https://en.wikipedia.org/wiki/Adventure_game) and [role-playing video games](https://en.wikipedia.org/wiki/Role-playing_video_game). Known as interactive fiction, text-based games combine the best of two worlds: books and interactivity. and they’re perfect for those who want to create interactive stories without pouring lots of time or money into art and sound assets.

The user manual for this program is nothing difficult such as a GUI or another application, it can run on Eclipse or the command prompts of any desktop as it is a simple Java program. It is difficult for me to add any other applications to it as it is beyond my knowledge of programming to do that yet, but it runs at a sufficient rate in a regular command prompt and does not need any necessary additions to it. Since it is only text-based, the need for a graphics interface is unnecessary since the user only has to type in simple commands for the program to run. Most role playing games in today’s generation are spent years in the making as there are many things to consider such as non-playable characters, the main character or the user, enemies, and even locations, take time to think about. Designs for each of these subjects are time consuming and difficult in order to get every specific detail correct. By looking at a more text based approach in creating the program, I felt that it would be easier for the user to run the program on just a command prompt or Java application such as Eclipse, so they are able to have easy access with no problems at all.

The beginning of the program contains the class TextBasedAdventure, which is the main class of the program, that allows all of the methods and other classes to incorporate from it. Methods that include the gameover method and those relating to the player, the enemy, the dice, and the combat. This is the most important class in the program as it is the source of it all and where everything is going to take place. The next class is Player, which describes the the attributes, the classes, the physical appearance, and the name of the player. The first thing the user is asked is what they want their name to be, and the user has the freedom to type in whatever they want to. The name could be their own name or whatever their heart’s content is. The next step in the Player class is to ask the user what type of class they would like to be, meaning if they would like to be a warrior, mage, or an archer, based on their personality and what they want their skill set to be. This class contains if statements such as if you put down warrior, it determines what type of adventure the user will have and how their experience differentiates from the other sets. They differentiate by the amount of health they contain and the amount of damage they do to enemies. I felt that this class was the most fun to make as it gives you the freedom to choose what your character wants to look like and imagining it how they do as well.

In Java programs, it is always interesting to see what the user inputs into the console interface and what the outcome would be like when the program runs its course. The best thing with programming is allowing the user the freedom to choose their own variables to create it by the developer and the user. Most of the variables in the program are developed by me in order to create the attributes such as the the playerhp, Level, and mana. The only ones the I had to modify for the user was the name and what type of role they are playing

as in the program such as warrior, mage, or archer. It is important to give the user a job when writing a program so they would not get bored by just reading something, but they feel as they have a part in it as well by putting their own creative responses.

The major class of this program is called Dice, meaning the user rolls a pair of dice that determine the user’s adventure and what their actions will be towards enemies or their class set. The rolls show what the user’s plan of attack or what the outcome of a situation is going to be. This class is also based on by the random import as the user “rolls” the dice and the outcome of a fight or situation is decided randomly based on the number of rolls that are made. Although this seems like a board game with the addition of a dice class, it allows the user to have multiple outcomes with the addition of rolls being returned at the end and the cycle keeps on going based on what numbers are obtained. There are different outcomes when a specific number is rolled that can change up the experience such as how the player goes into battle or if they can avoid it. This also affects combat sequences such as what the player will do in the situation and how it affects their overall process.

Another class that was newly added into the program is called Combat. This class consists of cases, with each of them with their own attributes such health, strength, and characteristics such as name and appearance. The Combat class is implemented into the main class and are placed throughout the program, whenever the user decides to fight and interact with them or not. It contains a while loop and cases in order for the user to determine how they will approach enemies and how they will choose the combat situations. The Combat class also contains if statements as well so they player is able to attack the enemy with either a solid weapon or with a magical skill. Since there is an if statement, the

situation is based on whether the user has enough energy or mana to implement their attacks and the program allows the user to see if they have enough or if they would need to choose a different strategy in dealing with enemies. Within the Combat class, it contains the statistics as well so many types of stats such as damage or health is converted as integers and doubles, that are defined in the beginning of the class. At the end of each combat situation of during it, it shows the health and the attack damage from both the user and the enemy known as the stats and shows the player if they level up with methods implemented into the class.

With the final production I was sadly unable to provide a Location class for the program despite the amount of time I had to do so. It was difficult trying to implement it into

the program as i was confused on some parts and many errors appeared in the process. Although it seemed easy doing a Location class since most of it is based on case statements and a few methods implemented, I had trouble comparing it with the rest of the classes and stretching it out with the rest of the program. Originally, the player was supposed to pick the location of where they wanted to go on the journey, but it was difficult to make more combat sequences for each location and had to pick more locations passed that. Instead of focusing of diverse places, I mainly placed the program in a singular location that I named myself, in order to let it run more smoothly. Another class I decided to change into as a method is the Enemy one. I found it easier to change it because it can go along with the rest of the classes in a smoother and easy rate instead of creating its own sequences. By doing so, moving it into the main class and the Combat class prove to be a sufficient feature of the program.

In its final stage, the program contains the necessary methods and classes it needed to run and for the user to enjoy the experience. It was a difficult process creating the program as there were numerous errors that needed to be corrected. Errors that were included in the program were as easy as spelling mistakes or just misplacement, but the most difficult ones included creating the classes and adding the methods within them to carry on the functions from the others. The most problematic one that I faced was the creation of the Dice class since I had to incorporate it into the others as well. I thought of keeping it as a method instead of a class to make the process easier, but it is important to modify it as a class for the project requirement and tinker around with the idea as well. Creating the enemystats() is difficult as well because it had to be included into the Combat class. Incorporating methods into different classes has always been an issue for me since understanding the concept can be quite difficult at times. We discussed in class the creation

of classes and how methods and variables are contained within in each of them, but the sharing of classes and methods are the only things I still get confused with to this day. Within the program, it took a good amount of time to devise solutions for each class, especially with the Dice class as well. Since the text-based adventure is mostly based on with the outcome of the dice rolls and the choices the user makes, it was difficult to combine them together to give the user the full experience of the game. I have tried my best to get the program working as efficient as it can get with all of my effort.

It important to understand some sort of background knowledge of what type of game this is as well since the user can be confused as to what this program is. The user is able to describe what type of character they are in the beginning and is immediately planted within

the game and are placed on a journey for the user to create themselves. Throughout my life I had a high interest for these types of games and always enjoyed playing them as I am able to create a secondary life and watch it unfold with the choices I create. It is one of my dreams to allow others to go through the same experience as I had and watch as a user goes through my creation. Although the program is far from perfect since it is not a fully fledged role playing game, it is a start in the right step and continue from this process to further expand my knowledge and do more programs such as this one in the future.