35/35

Marissa Piscani   April 26, 2019
Milestone   Software Development

Abstract

For my semester long project, I have decided to create a Mad Libs game on Java for the users to play. The program will allow the user to choose a story to complete out of 4 stories. They then will be asked to input words of their choice for the different parts of speech. The story will be displayed back to them so that they can read. They can repeat if they desire.

Introduction

I was inspired to create a Mad Libs game because when I was a kid I completed Mad Libs all the time. I would look forward to completing the Mad Libs so that I could read the story that I have created. I ultimately chose to do this because I think it is fun, and that it is something everyone will enjoy. I will first start by outlining the requirements, then a detailed system description, literature survey, user manual, and lastly a conclusion.

Requirements

The user will begin by running the system. The system will run through each class when the classes are called. When the game.displayIntro is ran through, the program introduces the user to the game and explains how the game will be played. It also advises the user to have fun, and to be creative when inputting the different parts of speech. It will also ask the user for their name and their age. After gathering personal information, the user will be asked which story they want to go through. It goes through a brief description on what each story is about, so that the user has an idea on what they are choosing. Once the user chooses the story they want, game.replaceWords is ran through and the user is asked to replace the parts of speech for the corresponding story that they have chosen. After that, game.conclusion is ran through and within that the graphics,display is ran it reads back the story to the user. The story will show up line by line and each line will be in a different color. After that, it thanks

the player for playing the Mad Libs after the story it outputted and asks them if they want to play again. If they say yes, they will start all over. If no, the game ends.

Detailed System Description

The system works by first running the displayIntro and outputting an introduction. It then uses a scanner and prompts the user for information. Then it keeps it in an array of text files and asks the users to pick a story number. Then it passes the story it retrieves from the array into the replaceWords function which reads the file word by word and if the word matches a key word it will prompt the user to enter a specific type of word, which will then replace the word that was they key word. Then, it passes the finished story back to the main method which then passes it to the conclusion method which passes it to the graphics display story method. After the story it outputted, the thread goes back to the conclusion method which outputs a conclusion.

MadLibProject	graphics
-graphics: object	
+displayIntro(): void +replaceWords(fname: File): String +conclusion(result:String): void	+ANSI_RESET: String +ANSI_BLACK: String +ANSI_RED: String +ANSI_GREEN: String +ANSI_YELLOW: String +ANSI_BLUE: String +ANSI_PURPLE: String +ANSI_CYAN: String +ANSI_WHITE: String +displayStory(story: String): void

Player	StoryFile
-name: String -age: int	
+Player() +Player(name: String, Age: int) +getname(): String +setName(name:String): void +getAge(): int +setAge(age: int): void	+fileName: String +storyfile(fname: String) +getFileName(): String +setFileName(fileName: String): void

Literature Survey

I have looked up other mad lib programs that individuals have done in the past to get ideas on how to do the graphics for displaying the mad libs. The most helpful website that I came across was the stackoverflow.com, and it suggested that I use ANSI escape codes to use color in my input. Another helpful thing I came across when researching about Mad Libs was how to import a txt file from the computer so that the users can modify it.

User Manual

The user will enter their name and their age when prompted to do so. They will then read descriptions of the story and they input the story number that they want. After that, they will respond with words too fill in the story when prompted. They will then be given the story that they have created. If they wish, they will do another story by selecting yes.

Conclusion

Overall, all my goals are going to be accomplished by this program. There are some tasks that still need to be done to complete this program, such as making the text files for the users to choose from. Improvements can be more complete graphics once i get the system to run.