

Mad-Libs

[illegible]

Play through

Open Mad-Libs with index.rb, make sure you have all all required files in the same folder. Game will load and give you a welcome screen with an enter to continue.

After this the game will give you options to enter up to 4 names, so we'll take 4 names to play the game today. Then it follows through with options 1/2/3/0, 1 and 2 select different stories, 3 gives you some information on how to play and 0 will quit the application. Once finished a story it will give you the option to continue with another story or exit the app.

Difficulties

GIT/GITHUB

The main difficulty we faced when creating this project was the use of git, using branches to your advantage and especially pulling/fetching. Especially for me when someone would create a new branch and I would try to upload another file it wouldn't let me push and I was so confused, until I figured out that I had to fetch the commits from Daniel and Huss. Creating branches was strange as well, we used them as a way to store the different features but didn't merge them into master until the very end (we had all features on an all features branch working instead of master). This assignment was great for getting us used to working with git and GitHub, and found out git is a must.

REQUIRE RELATIVE

Nands has taught us it is better to have separate files for certain functions, so we tried to stick with this by splitting up our features into files and combining them through requiring the file. At first, we forgot about having a separate file to call the objects and methods to run the code and tried to put it in each file which obviously didn't work. Thankfully nands noticed this and reminded us, after putting all the calling of objects and methods and a case statement into an index file and running that, the code pretty much worked perfectly. This is great as it keeps the code separate, easy to work on certain features without messing with other features and it's easier to read.

Favourite parts

-
- Our favourite part of this project would definitely be having a better understanding of git, once you get it, it's an amazing tool to have.
 - Using ascii words and colorise gem to create a much prettier looking game.
 - Putting in ridiculous inputs for a stupid but funny story.
 - Using Trello. It is so much easier working in a group project when you use Trello to split the load of work, which is always there to check up on and add stuff too!

Ethical/moral/legal issues

Minimal issues, the stories we have chosen are for all ages. Of course, depending on the words that are input by the user it might be for mature audience. But this completely depends on who is playing.

Possibly copyright from using stories picked from the internet.

Questions?