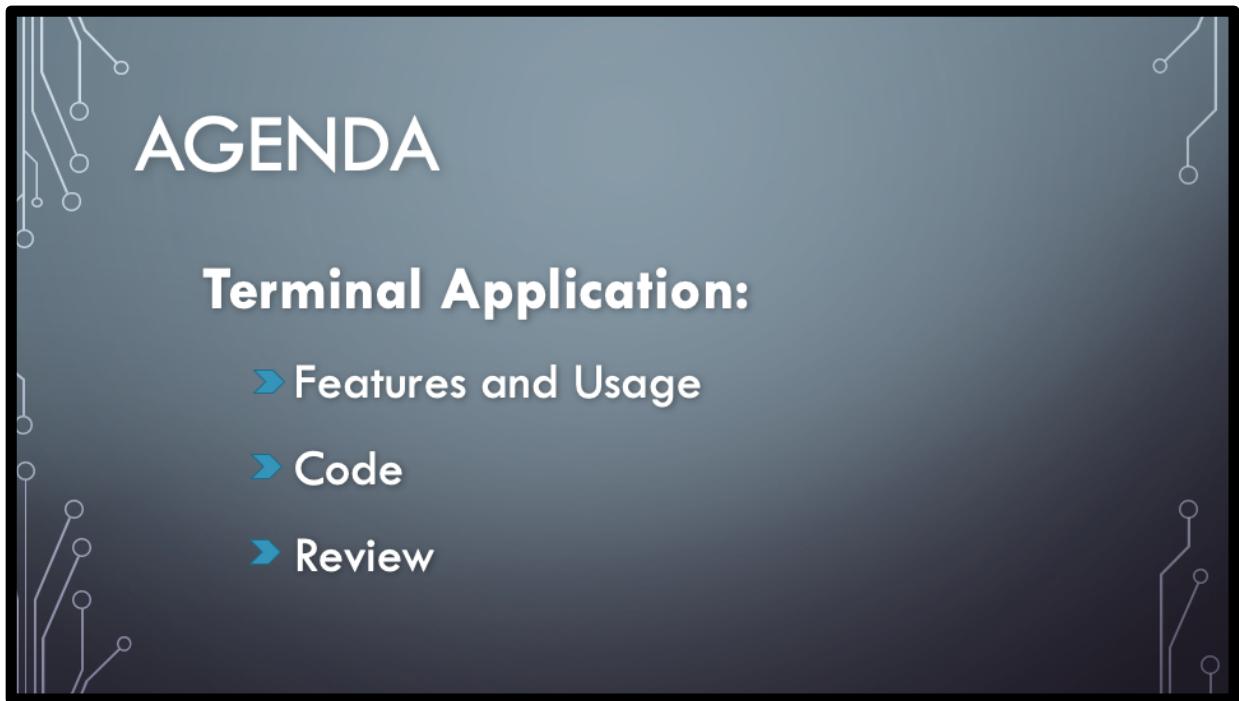




TERMINAL APPLICATION

WALDOW-THE-DEV

Hi gang,
I'm a Waldow, this is a presentation, and I built an app.



AGENDA

Terminal Application:

- Features and Usage
- Code
- Review

I'm going to cover these three topics today, so strap in.

OVERVIEW

Core Functionality:

- Star Wars themed
- ‘Scissors, Paper, Rock’ simulator
- ‘Sith-Lord, Jedi, Ewok’**
- Single Player Game vs AI

So what did I build?

Due to the fact I am a huge Star Wars nerd, I built a Star Wars themed ‘Scissors, Paper, Rock’ simulator, called:

‘Sith-Lord, Jedi, Ewok’

Which works as a single play game, with the player competing against the computer

FEATURES

Main Menu:

- Play Game
- View Leaderboard
- Read Rules
- Exit

Features:

From the Main Menu of the game, players can select one of 4 options:

- Play Game: This starts a new game of 'Sith-Lord, Jedi, Ewok'
- Read Rules: Displays the rules and scoring for the game
- View Leaderboard: Displays the top 10 scores for previously played games. This includes a save feature which persists data in a YAML file after the application is quit so players always have access to their previous high scores on their local machine
- Exit: Before exiting players are asked if they are sure they want to quit

SPRINKLES

- Coloured Text
- Ascii Header
- Prompt Menus
- Quote Generator
- Sound Bytes
- Round Tracker
- Player Name
- Options Sub-Menu
- Screen Clear
- Exit Check
- Sleep + DEMO mode

Here is a list of all the additional features I implemented to enhance the user experience while using the app or playing the game

REVIEW

Challenges:

- Lengthy Testing ➤ Require Loop Errors
- MVC ➤ Scope Creep

Challenges:

- Lengthy Testing:
 - I initially tried to build my app according to Test Driven Development principles, but this slowed my progress significantly as I wasn't familiar with this practice
 - I quickly abandoned this approach and just started building, which initially appeared to be saving a lot of time, but whenever I wanted to test a feature, I had to do it manually which ultimately would have taken longer than writing tests first
- MVC:
 - I also tried building the app in an MVC structure from my first line of code, but I began running into some serious trouble getting it to work and was wasting too much time with file structure and not building anything
 - Again I abandoned the 'best practice' for app development temporarily to just get my code working which meant most of my code was being written in the controller and nothing was being ported to any view files
- Require Loop Errors:
 - Once I had most of the basic functionality built I started moving all of the input/output tasks into view files. At first this was pretty simple as I could

cut and paste chunks of code into new files and just point to them with `require_relative`.

- What was challenging, however, was that I didn't know how to hand data back to the controller correctly so I just tried to point the view files back at the controller which ended up creating an infinite loop and crashing the app.
- I eventually realized I could just save the results of my View files to a local variable and then I was able to easily manipulate the data back in the controller
- Once I started refactoring my codebase I realized I didn't even need to do this ^ and came up with an even better solution
- Scope Creep:
 - Once I had the basic functionality built, I had buckets of time before the deadline, and was avoiding having to work on the documentation, so I started building tonnes of small features to enhance the user experience.
Note to self, don't do this again in future.



DEMO?

Now that's out of the way, who wants to see a demo?

APPENDIX

