CodeRaider

An 8-bit side scroller parody

Connor Harber, Sean McConnell, Hayden Moritz, and Johnathan Tibbetts

Idea

- Make a retro 8-bit side scroller game from scratch that presented our programming skills while simultaneously providing a unique and intriguing experience.
- Create a parody of the University of Wyoming Computer Science department that includes the professors and various challenges students experience throughout their college career.
- Create the game in such a way that anyone can enjoy playing it

Steps

- Needed to establish what languages and frameworks our game was going to use, as well as how it was going to be presented (executable vs web app).
- Basic content our game was going to include (storyline, mechanics, etc.)
- Determine and create artstyle/ artwork.
- Design basic game engine to render the character and their interaction with the environment.
- Design and construct levels.

Steps cont.

- Implement enemies and obstacles.
- Implement health bar and item mechanics.
- Deploy the application.

Technologies

- JavaScript for back-end
  - Physics engine, player movement, game mechanics
- HTML for front-end
  - Rendering, UI/UX
- Aseprite for graphic design
  - Sprite design

Features

Basics:

- Professors as bosses (interaction based on classes)
- Full range of movement for the player (jump,left,right,etc.)
- Player/Boss/Minion health and interaction
- Environment collision

Stretch goals:

- Inventory for items
- Environment interaction (vending machines, merchants, etc.)
- Refined collision

Sprites

Player

Bosses

- Drawn in Aseprite sprite development program.
- Based on classic ‘Megaman’ representation
- Background and populated environment objects based on Engineering Building hallways

Gameplay

- The player will begin in a menu screen where they can start the game.
- Once started, the player will be dropped into the first level.
- Each level consists of multiple rooms that are filled with moving enemies that can either be killed or avoided.
- After all rooms are completed, the level ends with a boss. Once the boss is defeated, the level is complete and the player moves on to the next level.

Challenges

- Due to the way our levels are loaded (each level is a different HTML file), sharing information between levels (remaining lives) is not allowed by default.
    - Our solution is to use local storage (cache) to store information such as health and ammo so that all levels can access it.

- We were having issues with the rendering and drawing of our sprites and animations. Occasionally, the sprites would render poorly or with floating pixels.
    - Refactored code to account for all frames, since our code originally dropped frames.

Successes

- Full range of motion, and smoothness
- Accurate player tracing via viewport
- Level sequencing
- Across level health logging
- Music
- Interactive minions
- Level completion and death

Future Work

- Score board (and in-game score)
- Upload the game to a dedicated site so it can be played in the future.
- Make the resources readily available for others to modify and add onto the game.
- Add other departments and locations to the game
- Implement additional character options

Q/A