Interactive Game

Stanley Kemp & Ronny Menendez

Motivation

Our project is motivated by a desire to develop and strengthen programming skills through engineering an easy-to-use, browser-based game.

Technologies Used

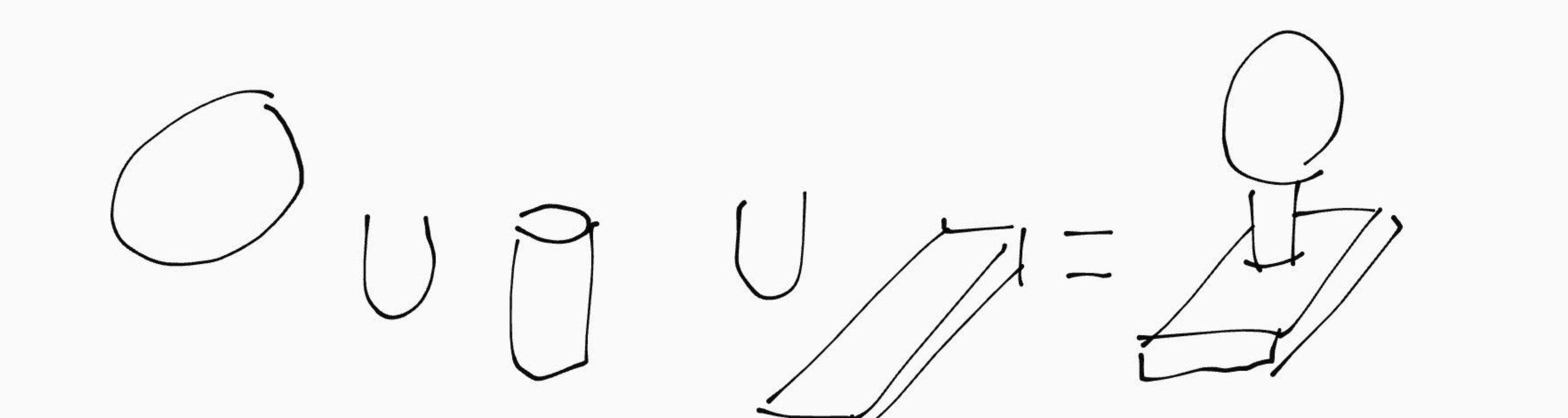
- HTML/CSS/JS (>=ES6)
    - 2D Canvas API
- HTTPS
- Three.js
- Browsers
    - Firefox
    - Chrome
- Text Editors/IDEs
    - Emacs
        - JS2-mode
        - org-mode
    - VS Code

Roadblocks

- Without following an individual tutorial for Three js, utilizing its library became an obstacle
- CORS
  - Launching a server to run the project locally
  - Discovering new packages that need to be installed via npm
- Transitioning over to an easier-to-implement project

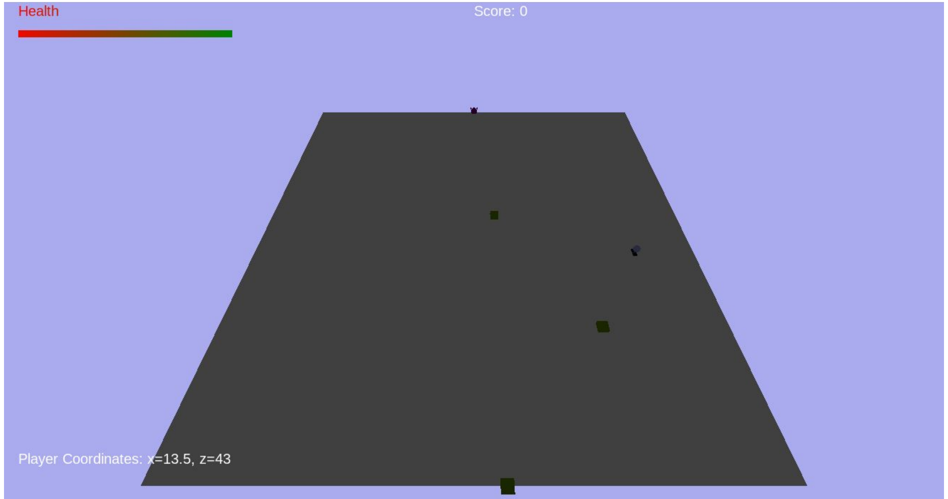
Creation of The Models

In order to get around CORS request issues, constructive solid geometry was used to create our assets so that it wouldn’t be necessary to load them from elsewhere, avoiding errors. As an example, this illustration shows the method to create the snowboarder/”skiier” that the player controls in the demo.



UI Creation

To serve as a UI, an HTML canvas was overlaid on which 2D graphics and text were drawn



Results

Next Steps

We should:

- Figure out why the game crashes occasionally
- Connect hit detection to the rest of the game
- Spawn new trees
- Add the monster