- Project Features List
    - Main Menu - Allows user to customize game experience with sound options, screen scaling options and resolution options, as well as to design character attributes
    - Game Environment - The game environment should include graphics for the level layouts and the character itself, including during different character movements and interactions.
    - Character Customization - The game will include various face masks for Common Sense to wear
    - Back End Game Storage - Back end character storage would allow the user to save the game state and their character stats / location for use at a later time. This will be hosted online.
    - Game Popups - These would include a respawn screen when the character dies, a beat level screen, and a beat game screen.
    - Character Interactions - These include moving, interacting with non-playable characters, picking up items, attacking, and interacting with the in-game environment.
    - Loading Procedures - These procedures would use the back end character storage to link the back end and front end while allowing the user to load a previous game state.
    - More features to be added...

- Requirements for Features
    - Main Menu
        - "As an experienced player, I want to be able to change character attributes and game options so that I can optimize my game experience."
        - Functional Requirements
            - The main menu contains a button so that it starts/loads the game
            - The main menu contains a button so that it moves to an options page of for the game
            - The main menu contains a button so that when clicked on it quits the game
        - Non-functional Requirements
            - The main menu displays an interface where all the functional requirements of the main menu can be accessed
    - Game Environment
        - "As an experienced player, I want to be able to see my game environment through graphics so that I can play the game."
        - Functional Requirements
            - The back end game storage loads level layouts and the front end technology stack displays graphics for the level
            - The character sprite changes depending on what the player is doing (moving, interacting, etc.)
        - Non-functional Requirements

            - The game environment loads in under 5-10 seconds
            - More than two levels (not including boss)
    - Back End Game Storage
        - “As an experienced player, I want to be able to save my game state so that I can load my progress in the future.”
        - Functional Requirements
            - The game saves a previous game state as directed by the user, including the current level and any customized character traits, in the back end technology stack
            - Back end has to use server (can not be local)
        - Non-functional Requirements
            - When the player chooses to save their game, the game saves necessary data in under 5 seconds
            - Provide multiple saves for a single player
    - Loading Procedures
        - “As an experienced player, I want to be able to load a previous game state so that I can continue playing with my previous progress.”
        - Functional Requirements
            - The game loads a previous game state as directed by the user through the integration between the back end and front end technology stacks
        - Non-functional Requirements
            - If there is no previous file to load, this should not be possible
            - The procedures load necessary data in under 5 seconds
            - The procedures load a history of saves.
    - Game Popups
        - “As an experienced player, I want to be able to see graphics displaying respawn options and level completion so that I can understand how much progress I have made in the game.”
        - Functional Requirements
            - The game popups display a graphic when the player dies and gives the option for respawning or exiting the game
            - The game popups display a graphic when a player beats a level
            - The game popups display a graphic when a player finishes the game
        - Non-functional Requirements
            - The game popups display the FPS of the game.
    - Character Interactions
        - “As an experienced player, I want to be able to move my character so that I can play the game.”
        - Functional Requirements
            - The player should be able to move their character using the keyboard.
            - The player should be able to interact with the game environment.

- Non-functional Requirements
  - A player is able to interact with other players.

Individual Contributions:

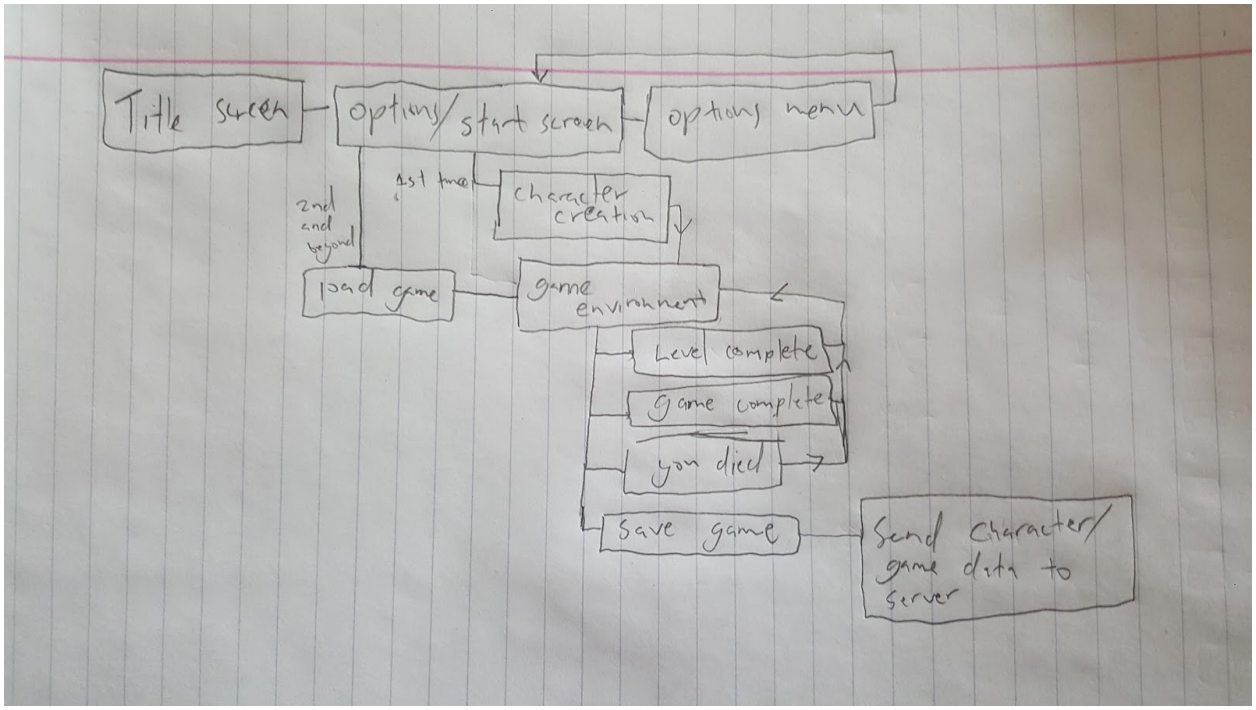
Since we are still in early development, as well as the uncertainty of if our project is even viable, we have worked mostly as a collective unit. Each of us added our own ideas and thoughts to the game design as well as the plan on Jira thus far. Outside of our team meetings, we have all been doing research on the engine in which we want our game to be designed on. We researched and played around in everything from Unity to GoDot, and we eventually landed on Oxygine. This will allow us to seamlessly work in C++ and HTML. In Jira, we each added several user stories to the backlog and later organized them into sprints and epics. Each of us brainstormed several ideas for the story behind our game and as a group, we decided to have the game based around 2020. After this, we hopped into a video call and began to think about art style, basic movement mechanics, and discussed the concept of a time based leaderboard. This will add a competitive element and motivate the player to be as efficient as possible.

- Mac made some very basic sketches for the game interface and played around in Unity at the start of the project since it's the most standard of all game engines. After seeing that it was mostly based for 3D and was C# based, we decided against it.
- Dan took it upon himself to begin diving into the story and lore of the game. This includes everything from what bosses, what backgrounds, and stories that the game would portray.
- Sam F worked on title and character design. He did some research outside of the necessary to find extra knowledge on game design and the linking of back end to the front end.
- Sam B and Dom focused on creating our roadmap and user stories. By planning ahead, they thought that we would be more efficient and focused overall. The time constraints add an element that will push us to get things done in a reasonable time, rather than saving it all for the last minute.
- Matt created some wireframes and basic images for the games UI, as well as researched game engines. He eventually found oxygine, a C++, web based engine which we eventually chose as our weapon of choice.

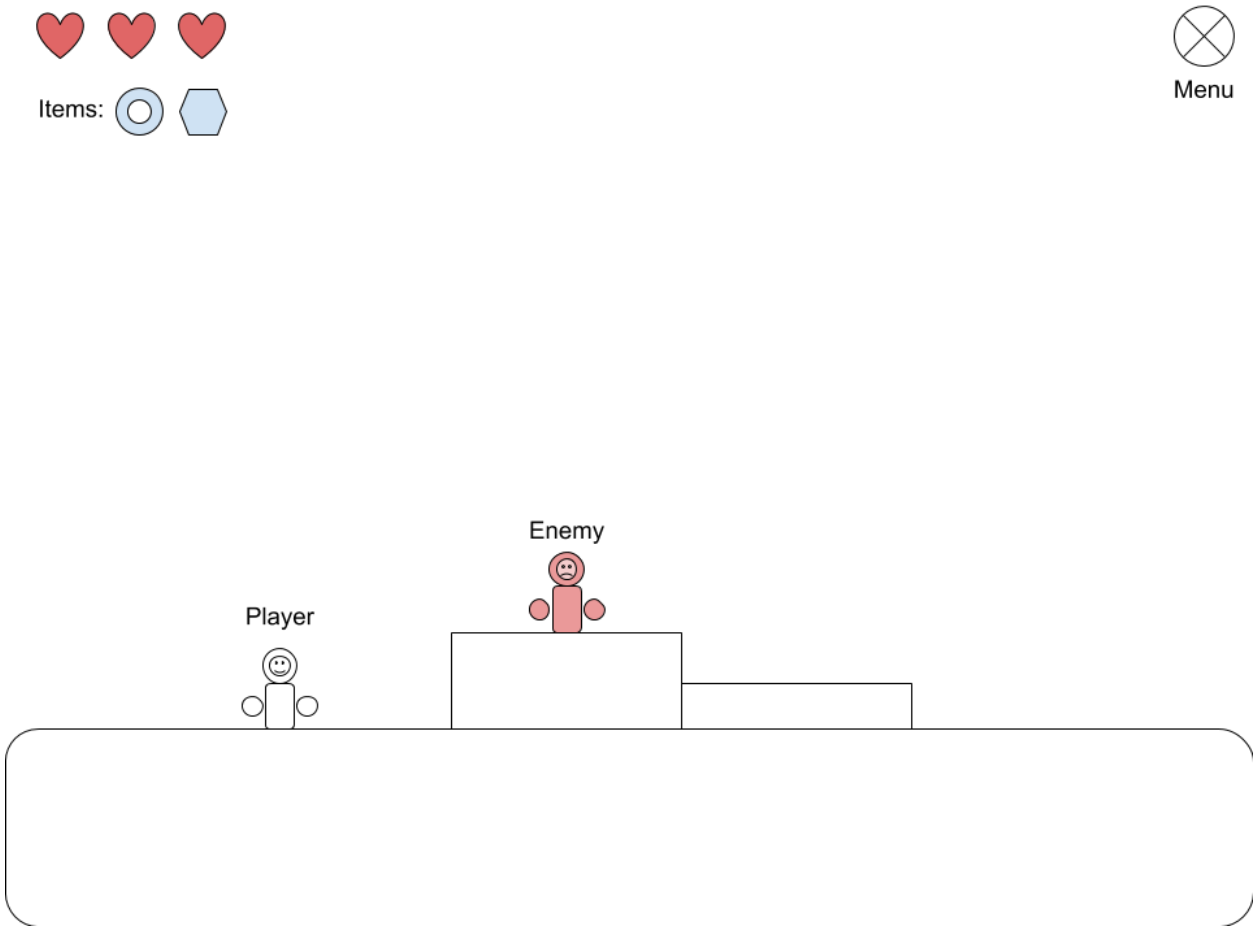
Project management page:
https://teamprofanity.atlassian.net/jira/software/projects/T23/boards/1/backlog

Note: Each feature is integrated into an epic on our roadmap.

Game Flowchart



Level Wireframe



Main Menu Wireframe

New Game

Load Game

Customization

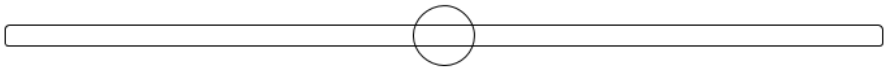
Settings

Credits

Settings Wireframe

Settings

Volume



Fullscreen

Extra Artwork