

Herb da Derp  
COMP 195 Senior Project

<https://github.com/comp195/senior-project-implementation-herbdaderp.git>

**Team Members**

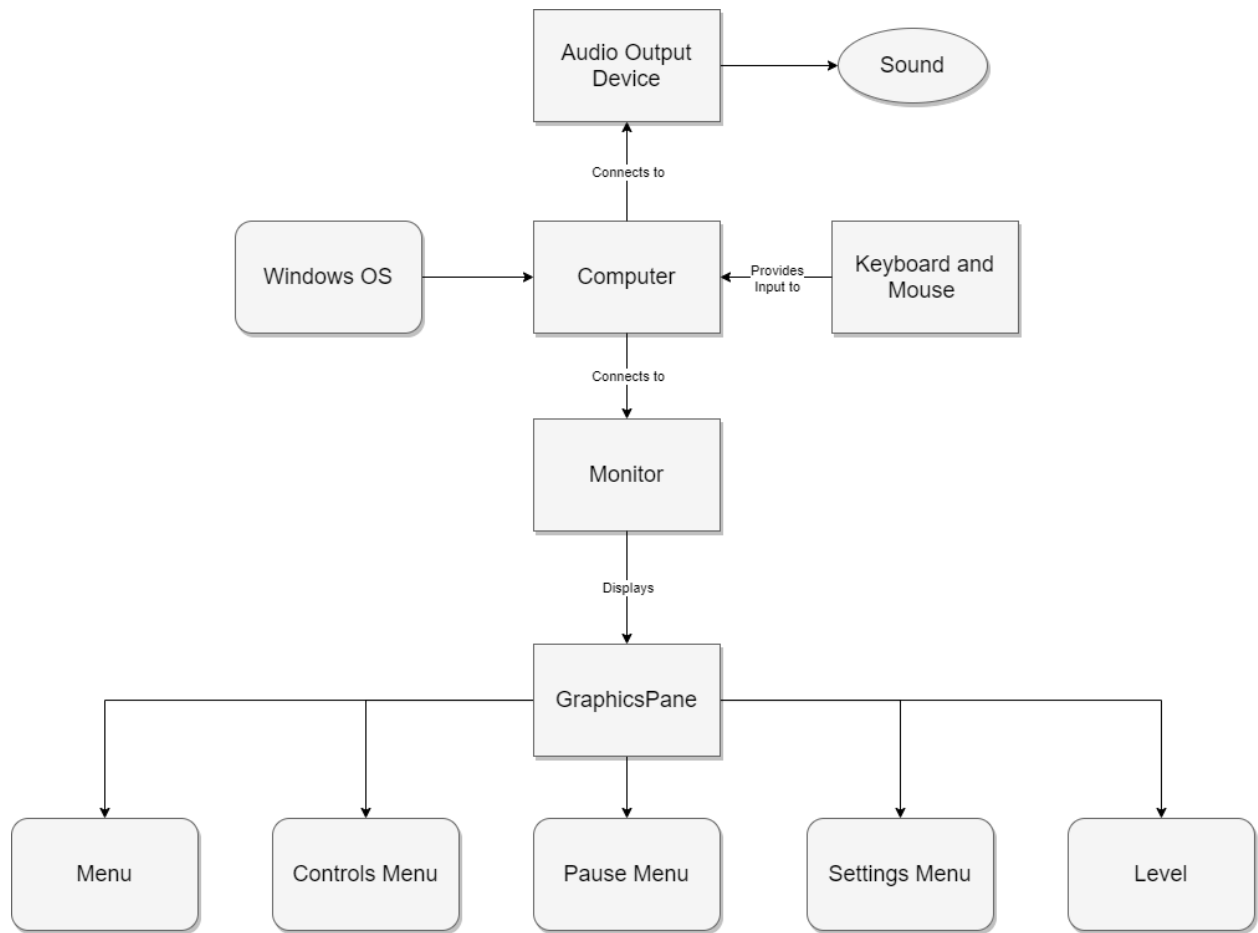
Jefferson Pelera (j\_pelera1@u.pacific.edu)

Trinity Belen (t\_belen@u.pacific.edu)

James Nicholas (j\_nicholas@u.pacific.edu)

September 26, 2021

## System Architecture



## Hardware, Software and System Requirements

### **Hardware Requirements**

Processor: 2.37 GHz

Memory: 1 GB

Graphics: 1Gb Video Memory, capable of OpenGL 3.0+ support (2.1 with ARB extensions acceptable)

Storage: 1 GB

### **Software Requirements**

Just download the game and run it.

### **System Requirements**

OS: Windows XP, Vista, 7, 8/8.1, 10

## **External Interfaces**

### **Monitor**

The monitor is where your screen is displayed to allow you to play the game, either part of the system (Laptop) or connected to the computer with HDMI, DVI, or Displayport.

### **Mouse**

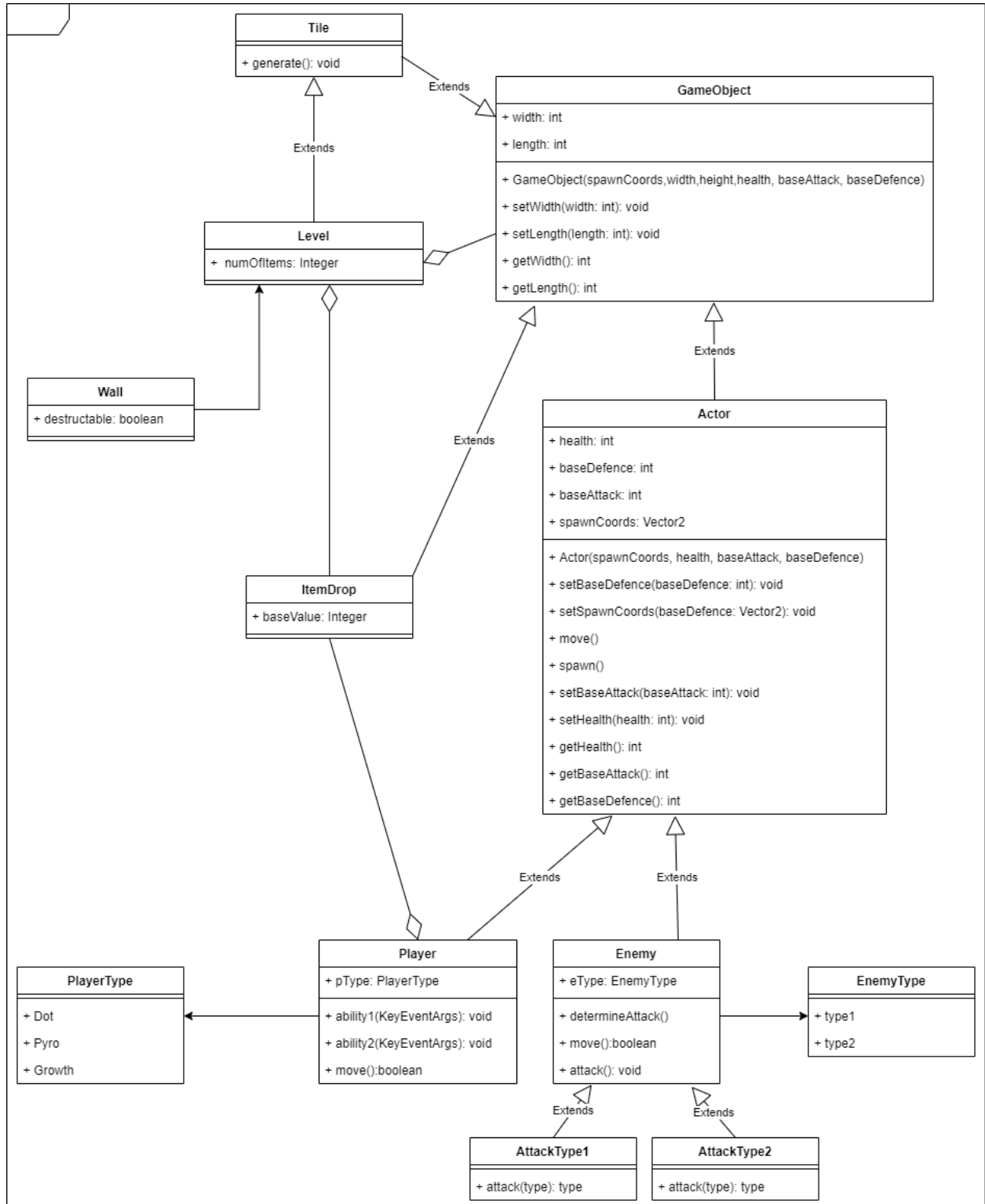
Used for primary navigation of the game menus and attacking in the game. Connected to the computer through USB or bluetooth.

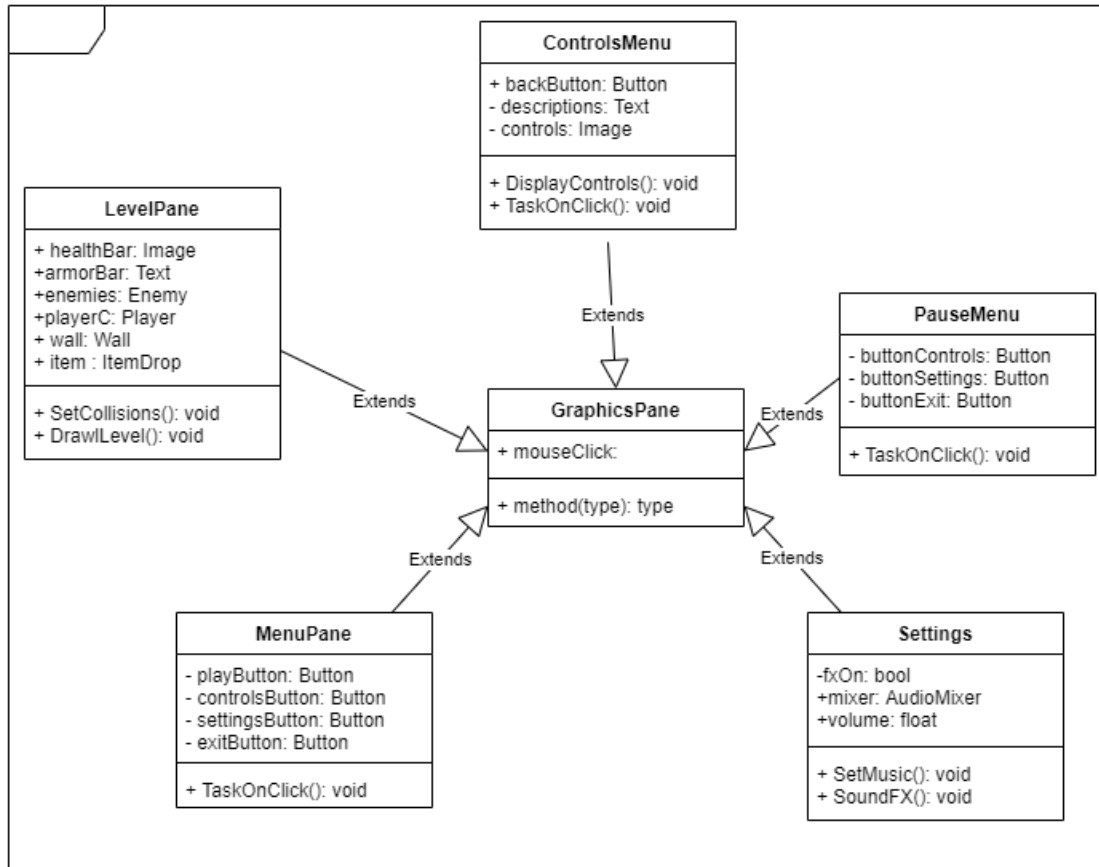
### **Keyboard**

Used to move throughout the game and use your abilities. Connected to the computer through USB.

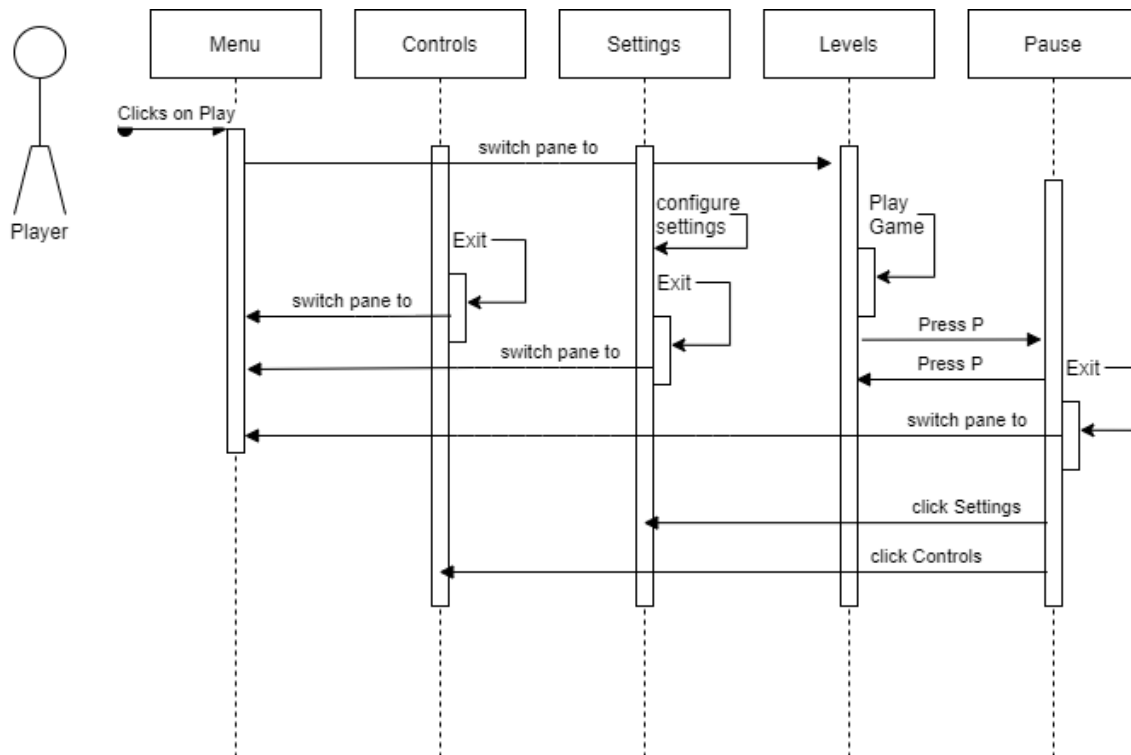
## Software Design

### Class Diagram & Specifications





## Interaction Diagrams

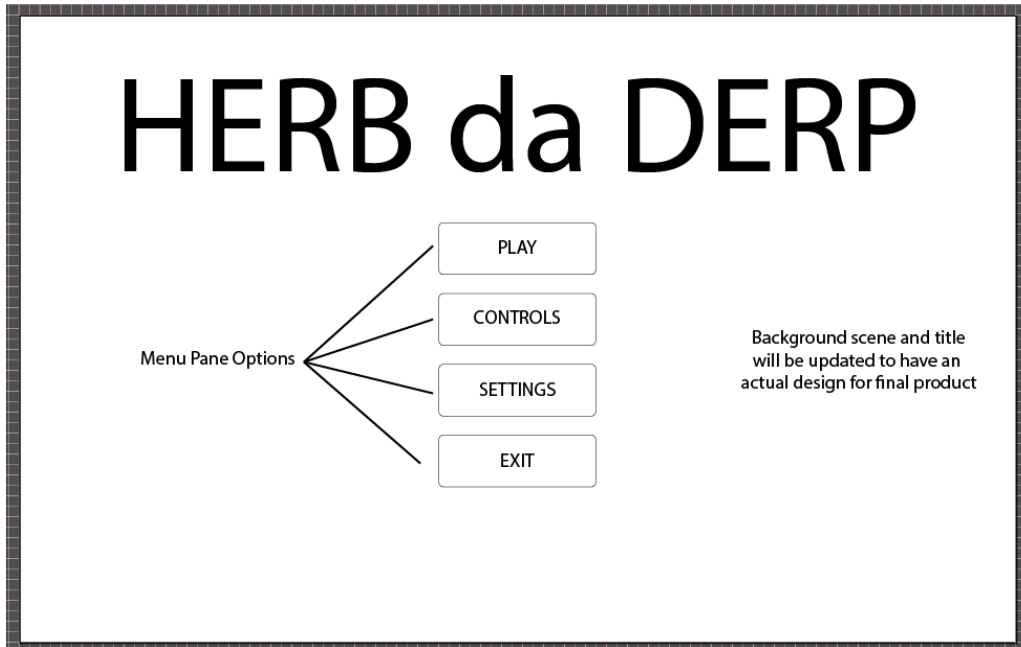


## **Design Considerations**

Some things we wanted to consider is that many elements of the game have the same attributes. For instance, both enemies and the player themselves have a base attack, defence, health, and spawn on the map. With that in mind they both extend from their parent class which would be Actor. Making use of the dynamic of parent and child classes will help with organization and help break things down to simpler parts.

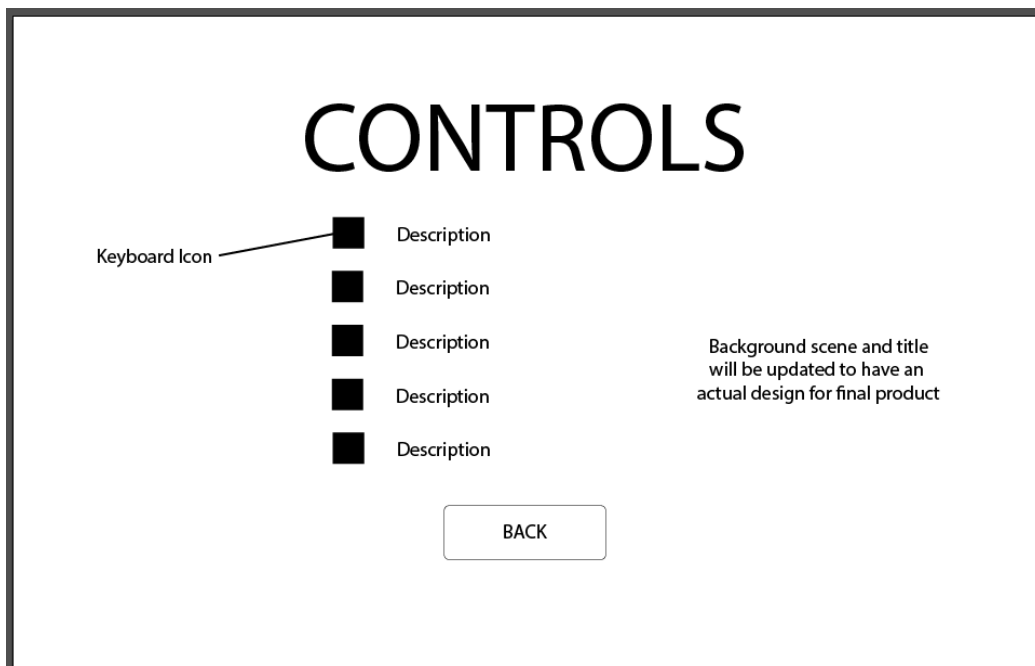
## User Interface Design

### Main Menu



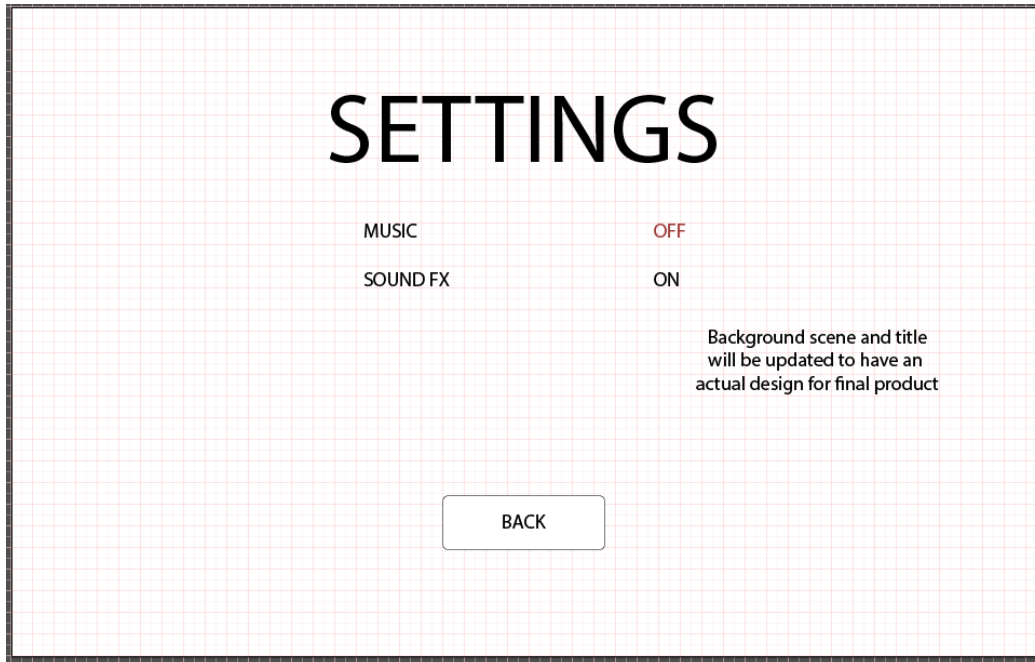
The menu that players will see once they start the game. There will be buttons for “Play,” “Controls,” “Settings,” “Exit.” Clicking on “Play” leads to the level being generated. Clicking “Settings” and “Controls” lead to the settings and controls menu respectively. Clicking on “Exit” will lead to the game’s window being closed.

### Controls



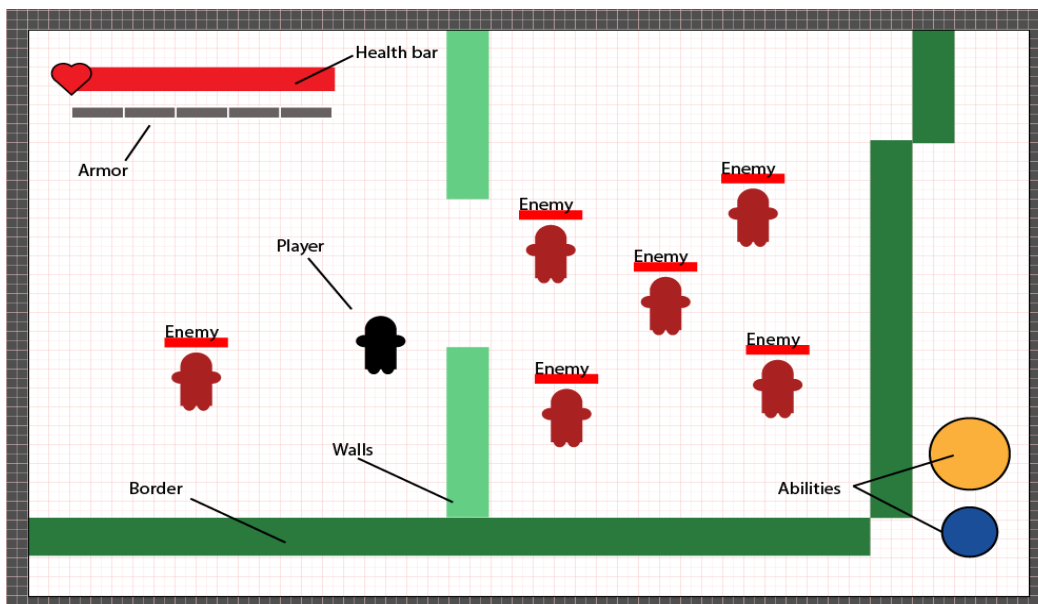
Once in the Controls Pane the user will have the option to view the controls for the character they will be playing. Afterwards, the user will click on the Back button to go back to the menu or the level depending on where they clicked on Control.

## Settings Menu



In the Settings Menu players will have the option to toggle on and off the game's Music and other audio related settings. Afterwards, the user will click on the Back button to go back to the menu or the level depending on where they clicked on Settings.

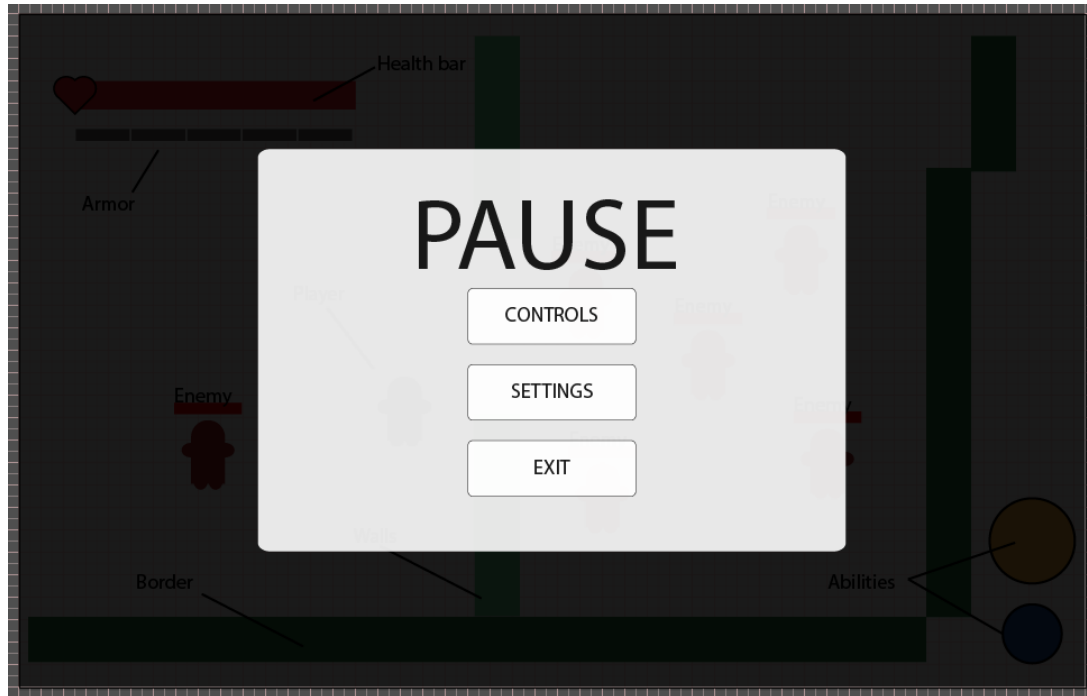
## Level UI





The UI for the Level, you can see the health bar for the enemy on top of the enemy sprite. You can see the player's sprite on the upper left corner. The player will use the controls described in the Controls menu to play the game.

## Pause Menu



After pressing the escape key, players will see the pause menu. The options are in the center. Clicking on "Controls" lead to the Controls menu, "Settings" lead to the Settings menu, and "Exit" leads to closing the game.

## **Glossary of Terms**

### **Ability**

The skills that the player can use to attack the enemies.

### **Armour**

An attribute assigned to the player and enemies that acts like a second health pool that will regenerate when out of combat for some period of time.

### **Cooldown**

Time it takes for a specific entity to perform an action.

### **Critical Chance**

Percent chance that a given attack scores a critical hit.

### **Critical Hit**

Every attack has a critical chance, if you crit then you do 2x damage.

### **Enemies**

Obstacles of the levels in which the player has to fight and defeat in order to progress through the levels.

### **HP (Health Points)**

An attribute assigned to the player character and enemies that allows them to continue functioning.

### **Level**

A series of stages or rooms that the player has to overcome in order to progress through the game.

### **PVE**

Player versus Environment, meaning that the player is not fighting against other players but against computer controlled enemies.

### **Procedurally Generated Levels**

Levels that are not hardcoded, but are randomly generated from the game's template.

### **Roguelike**

A subgenre of role-playing games that has a dungeon crawl through procedurally generated levels.

**Tactical Ability**

An ability that the hero has that can be used every few seconds to buff himself or do damage to the enemies.

**True Damage**

Damage that ignores defenses.

**Ultimate Ability**

A strong ability that can only be used if you have killed enough enemies to charge your ultimate meter.

**Ultimate Charge Bar**

A bar with x sections (at the moment it's 5) that fills up as you kill enemies. Once the bar is filled you will be able to use your ultimate ability.

## **References**

Handheld Android. (2017, Jan. 13). *Soul Knight (Android Gameplay)* [Video]. Youtube.

<https://www.youtube.com/watch?v=KQO2qsa5qWU>

The gameplay of Soul Knight (as shown in the video) is used as one of the inspirations for how the game will work as a roguelike.

Harris, J. (2011, Feb. 01). *Analysis: The Eight Rules Of Roguelike Design*. Gamasutra.

<https://www.gamedeveloper.com/pc/analysis-the-eight-rules-of-roguelike-design>

This article analyzes the critical design elements of successful roguelike games. We will use this as a guideline for the steps we are taking to create a roguelike game.

King, A. (2015, April 10). *The Key Design Elements of Roguelikes*. Envatotuts+.

<https://gamedevelopment.tutsplus.com/articles/the-key-design-elements-of-roguelikes--cms-23510>

The article defines what a roguelike game is and talks about key elements of roguelikes such as procedurally generated environments and the randomization of items. These elements are used as a reference to what elements we are going to program in.