**Work in Progress Report 5**

Major developments/breakthroughs(reference specific code please):

- Added menu screen to start the game

-Added Options screen to select character and level to play

-Lock levels prior to completion of the previous level

- Implemented the “one giant entity list” idea, it’s kinda beautiful

Major Challenges/setbacks( reference specific code please):

* None

Any modifications to your specifications/release schedule:

* None

**Description of your scratch/test program:**

1. **ScreenScratch (Rueban)**

**Describe the generic concept you needed to test out:**

- Adding multiple screens to the project

- Allowing user to set the level they want to play and to give a choice in the selection of the player (we dont have a separate player skin for this yet)

**Source any web site/book that helped you with that concept:**

- Intothewoods group for the screen management code Github: https://github.com/spidermanchild/IntoTheWoodsMultScreens

**Describe the code and the lesson that you learned from it:**

- In the GameMain class, the main menu and options screen will be created first.

- From there, the user has to select the level they wish to play. Only the first level is unlocked, for the rest of the levels, the user has to complete each subsequent level (we don’t have a completely different level for this either, just the first level with minor changes)

- Array of boolean are used to determine whether the levels have been completed or not. A sprite is used to show the user that a level is locked and in the button listener, the array of booleans arbLevelUnlocked needs to be true for the listener to work. When the boolean is set true, the sprite stops being drawn and the if statement in the listener is triggered.

-Player and level choice

* Player clicks on the image button that depicts the player/level
* Button listener sets a string that will get passed into the game screen to load a level with the specifics set in the options menu
* The level is initialized in the GameMain class

**Describe any challenges that you enjoyed in integrating this scratch code into your major project:**

- AndroidStudio refactoring tools are really useful, sometimes they make life much easier, sometimes they crash your IDE, it’s really up to how they’re feeling that day

1. **InheritanceScratch (Kevin)**

**Describe the generic concept you needed to test out:**

- Using a giant entity (interface) list instead of having multiple lists

- Finding a way of adding entities while updating them without causing a ConcurrentModificationException

**Source any web site/book that helped you with that concept:**

- Source SDK 2013 for the general concept from “cliententitylist”, the implementation was much different but the overall idea is similar: <https://github.com/ValveSoftware/source-sdk-2013/blob/master/sp/src/game/client/cliententitylist.cpp>

- My own brain for the “entityBuffer” idea, damn im smrat

**Describe the code and the lesson that you learned from it:**

- Re-coded many of the classes to work nicer with inheritance and the entity list

- Added a “entityBuffer” in the GameScreen to allow entities being updated to add more entities to the array (granted it delays the spawn of the new entity by 1 tick) and bypass the ConcurrentModificationException

**Describe any challenges that you enjoyed in integrating this scratch code into your major project:**

- A few of our classes were still using old code that newer scratches improved on

**Peer Assessment:**

Don 100

Kevin 100

Rueban 100