**Work in Progress Report 2**

Major developments/breakthroughs(reference specific code please):

-creating an enemy class and spawning it in in configurable quantities

-allowing collisions between the player and obstacles like spikes

-integrating touch support to player movement

-fixing foot sensor code to prevent sticking to ceiling when jumping

Major Challenges/setbacks( reference specific code please):

-figuring out hit testing between players and enemies, and between enemies. This is setting the category bits and mask bit values in the player and enemy class

Any modifications to your specifications/release schedule:

-no modifications

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

**1. (Rueban) EnemyScratch**

Describe the generic concept you needed to test out:

-spawning enemies that will track the player across the map and be able to cross obstacles like walls

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

Main Source: Don’s code for base player class

Array of Objects: <https://github.com/Mrgfhci/Drop/blob/master/core/src/com/mygdx/drop/Drop.java> line 58

MaskBits and filtering: <http://box2d.org/manual.html#_Toc258082970>

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

-the code for the enemy class is essentially identical to the player class with the exception of the move function which has been tweaked to facilitate automated movement instead of user inputted controls. The enemy tracks the player by constantly keeping track of the player’s coordinates and then setting its velocity in the direction of the player. If the enemy’s velocity in the x direction is 0, it will assume the path in front of it is obstructed and it will set a linear impulse in the positive y direction in order to clear the obstacle(walls). Currently the enemy is also able to cross the bridge in the ‘debugroom’ map. That is because the enemy is treating the side of the bridge as a wall and is jumping to clear the obstacle. I learned that the hit testing for environmental objects has to be fixed so as to not allow enemies and player to jump without ground beneath it. Using Don’s code has given me a better understanding of the way different functions are called and the frequency they are called. Any number of enemies can be spawned in depending on the preset bounds of the array of enemies. Through the population of enemies I learned that if two enemies are colliding, the collision between those enemies and the environment is disrupted. I used code from the Drop project to spawn multiple enemies.

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

-not integrated yet

**2. (Kevin) MovementScratch**

Describe the generic concept you needed to test out:

-integrating touch controls into movement of the player

-fixing foot sensor code to prevent sticking to ceiling when jumping

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

Box2D sensors: <http://www.iforce2d.net/b2dtut/sensors>

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

-the touch movement code is similar to the original keyboard input but checking for touch input instead of key input.

-for foot sensor code, the foot sensor itself is just a fixture and doesn’t require it’s own body, we just add it to the mainBody of the player. In the MovementTest file I changed the foot code to compare the colliding fixtures with player.footSensor (the foot sensor fixture). What we were doing before is checking the categoryBits of the fixtures, which may have worked if implemented correctly but was, unbeknownst to us, treating the body fixture as the sensor itself thus causing the player to stick to the ceiling while jumping.

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

-not integrated yet

With each WIP, you will be submitting EVERYTHING. Organization is key. When I go to the groupwork folder**, I should see your project submitted in the following format:**

YourLastName: Under this folder will be the following folders:

**Asana Specs**: Your Asana calendar will have a task that contains a github link to your project and scratches. Please add any comments within this task that can give me a better understanding, like : “It does not work.”

Even if you provided the link to the same project in a previous task from a previous month – go big – add it again.

**Documents**: It will hold all of your documents: journal, WIP, Specs, Release schedule, list of sources, and all the other documents that will be submitted in your final project.

**Releases**: There will be a folder for each release, with one folder CLEARLY telling me that it is the latest, stable release.

**Scratch**: There will be a folder for each scratch concept that you tested before you integrated it into your final project.

**Peer Assessment:**

Don 100

Kevin 100

Rueban 100