**Work in Progress Report 2**

Links to GitHub:

Most of our code is here. Including scratches and releases:<https://github.com/AbhinavA10?tab=repositories>

Some other earlier scratches are here:<https://github.com/MattM7?tab=repositories>

There are also some other failed scratches,

and the animation and gravity scratches that are only in groupwork. We didn’t put them in github because we did not want to have failed scratches there.

Fourtress is latest stable version.

Major developments/breakthroughs(reference specific code please):

We were able to load add animations to our sprite hero, make him have non-infinite jumping using contact listeners, and also made him able to shoot. We also integrated the menu scratch, and spikes to our game. Our most prized development this WIP is the ability to remove the bullets once they hit a platform, and remove the tile that it hit. This is done by using an arraylist and a boolean bDead variable in the Bullet and DestructibleTile classes.

We also looked into having the cone lights (for the flashlight) run in SpriteExtended instead of the Lights Util. We saw no benefit in this, as it basically worked the same. Furthermore, it would take slightly more lines of code to make the lights render, and since they would have to be public to make it work, we left the lights in Lights under the util package. It seems more organized this way and still works the same way.

We have basic AI that can follow a Box2D body. There are slight issues with it not decelerating quickly enough. This can be seen only on Matthew’s GitHub.

Major Challenges/setbacks( reference specific code please):

A major challenge since the last WIP was removing the right tile/body based on the bullet collision. We tried in multiple scratches to create a box2d body, and when it hit a platform, remove both the box2d body corresponding to the bullet, and the one for the platform. At home, it worked to the extent that the bullet could be removed, but the coordinates for the hit platform were not returned correctly. At school however, the program crashed on the creation of the second bullet after the first one was removed. This scratch can be seen under the Failed\_notWorking folder. So, after a couple of days, we started again from scratch, this time including just Box2d and not the tiled map.

Any modifications to your specifications/release schedule:

We modified the enemy section of our release schedule to better accommodate for AI.

|  |  |  |
| --- | --- | --- |
| 4.0 | [Griffon](http://bakugan.wikia.com/wiki/Griffon) | Enemies that can do basic movements, and can be shot |
| 4.1 | [Infinity Dragonoid](http://bakugan.wikia.com/wiki/Infinity_Dragonoid) | The enemies fire back when close, and are animated |

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

Describe the generic concept you needed to test out:

Scratch name: Bullets

We needed to test the removal of bodies upon collision.

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

We used this link to get an idea of how and when to deal with removing bodies: http://box2d.org/forum/viewtopic.php?t=9724

And this one to reverse the class of objects in ContactListnerUtil: https://docs.oracle.com/javase/tutorial/essential/exceptions/catch.html

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

When the mouse is pressed, a dynamic box2d body is created at the mouse’s location by adding it to an arraylist of live bullets. Upon collision, destroy() in BulletScratch is called. In this function, it removes the bullet object from the live bullet arraylist using getUserData(), adds the bullet object to a dead bullet arraylist. During render(), it cycles through the dead bullet arraylist, destroying each corresponding body, and then removing it from the arraylist.

We learned that the simpler off we start from scratch, the better. Having less variables and functions is a lot cleaner.

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

There were no challenges integrating this scratch into the our major project. Instead of making two arraylists, one for live and one for dead bullets, we made a variable bDead in the major project. This made it a lot cleaner.

A challenge we did have when integrating a scratch, was the RemovableTile scratch as the correct coordinates for the tile was not being returned. After too long, we realized that this was because the position of the body for the tile was not being set; so, right now only rectangle destructible tiles can be removed as we have not yet tried setting the position of polyline objects.

**Peer Assessment:**

We both agree that we did equal work, so the mark should be weighted equally. 100:100