**Work in Progress Report 1**

Due: Friday, Nov 20

Name: *Matthew Meade and Abhinav Agrahari*

Major developments/breakthroughs(reference specific code please):

We were able to integrate arrayList collision detection, gravity, scrolling, and multiple levels into our game. (version Clone Trooper) We were able to do this thanks to Matt’s “collisionWithArrayLists” and Abhi’s “scrolling\_AllDirections\_Work” (with the help of some downloadable source code, for the scrolling. The link is in the “List of Sources” document)

Major Challenges/setbacks( reference specific code please):

We looked into playing a video in processing (so we could play them, when the user tries to give up, as motivational messages), but the school computers do not have the required jdk version. You are able to take a look at the scratch code in “Video\_failed”.

We then looked into Box2D but, as you know, we decided to abandon it as the syntax was way over our heads. If we had another year or two… then maybe… just maybe we could’ve gotten it to work. All our Box2D scratch code is in “Box2D”. Note: all the code is copy pasted bit by bit while trying to understand it, from Daniel Shiffman’s Nature of Code website, so none of it is made by us.

Originally, Abhi was trying to get scrolling to work by writing a function that returned whether or not the “x” position of the player was in the center of the screen, and then depending on that the “camera” would or wouldn’t translate *n* number of pixels depending on the direction of the player. This method (of having the player move at times, and at other times, the camera) did not work when the player moved left. This scratch program can be seen in “scrolling\_rightWorks\_notLeft”. We later found some code (which is listed in the sources) that used the method of keeping the player in the center of the screen at all times if possible. This method worked much better, and we then went on to incorporate scrolling along the “y” axis, using the same method.

We spent two weeks attempting to get the main character (an image) to detect whether or not he had hit the objects ( images of boxes) in an arrayList with gravity working at the same time. We had collision detection in the arrayList, and that worked fine. As soon as we integrated the scratch into our main code everything stopped working the way we wanted it to, because the gravity in our main code seemed to overrule the collision detection. In “Boba Fett \ Don’t Work \ DontGiveUp\_abhi”, and “Boba Fett \ Don’t Work \ Ye\_Old\_Code”, you can see exactly what was happening: either the main character ignored the boxes, or “lagged” through them.

Any modifications to your specifications/release schedule:

In version 3.0, “Clone Trooper”, besides the 9 different levels, we added a timer that displays on your screen ( it basically tells you how long you have been playing).

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

Describe the generic concept you needed to test out:

We wanted to be able to detect collisions with objects in an arrayList. It was fine until there, but with gravity things started going haywire.

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

Besides Mr. Grondin’s website, no other website/book was used to help us with the concept of collision detection with an arrayList.

However, we did use a website to help with the scrolling concept: [http://www.hobbygamedev.com](http://www.hobbygamedev.com/int/platformer-game-source-in-processing/) (full link is in the list of sources document), and we used the downloadable source code.

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

The scratch program worked: the main character (a box) was able to detect when he had hit objects in a separate arrayList. By looping through the arrayList, we can check each object in the arrayList. We learnt that the architecture of a code is very important when we have a lot of code that may conflict with each other.

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

With the scratch,“collisionWithArraylists”, we had many problems integrating. This was due to the fact that this scratch did not include gravity; it just included collisions. When we went to integrate it into our major project, and added gravity, the main character instead of stopping at a wall did one (or more) of three things: just “lagged” through it (moved at a slower rate), not even care about the boxes when moving left/right, and most of the time jumped through them. You are able to see what was happening in “Boba Fett -> Don’t Work -> Ye\_Old\_Code”.

To solve this, Matt came up with another way to check for collisions and modified the main program using the concept of the scratch, but checked collisions twice instead of once (depending on which time it was, it set the “current” x/y coordinate of the character to the x/y coordinate of the character before it collided with something). There is another version of “collisionWithArraylists” which is called “collisionWithArraylists\_and\_gravity”, which includes gravity. We then integrated and it worked!!!