**Work in Progress Report**

**Major developments/breakthroughs(reference specific code please):**

Guest Sprite:

Hearts Object: The hearts are partially dependant off of the Guest. The hearts must follow the guest as well as change textures to display the Guests’ patience.

**Major Challenges/setbacks( reference specific code please):**

Viewport:

Viewports and cameras are used to help scale and resize a games’ coordinates. However the camera does not update to centre of the phone’s screen which is resulting in offset touch coordinates. It seems that

public void resize(int width, int height) {

viewport.update(width, height);

camera.position.set(nW / 2, nH / 2, 0);

}

Isn’t running or another cause. Unfortunately we haven’t found a solution yet.

Waiter pathways/movement:

This was a major setback because it is one of the main factors of the game that was required to be perfected and finished to code for other components. Such components would be coding the waiter taking the order, taking the order to the kitchen then back to the table when finished, and cleaning the table once customers leave. One of the major problems I experienced in the SctWaiter scratch was getting the waiter to go to the table upon one single click/touch to wanted said table. Firstly, I wanted the waiter to travel horizontally to the tables x-coordinate upon a single click; but, the waiter only moved 1 centimetre whenever the mouse clicked the table. This was solved by separating the code and making a int variable to check that the user did click the table, then nDirectionCheck =1 for horizontal .

if (objTable.isMouseOver() == true && Gdx.*input*.justTouched()) {

nDirectionCheck = 1;

}

Which also came to the problem of the waiter not stopping when the x-coordinate of the sprWaiter and tbl1 [table 1] aligned. Instead, sprWaiter continued to travel to the right side of the screen. This was solved by separating the boolean checking the x-coordinate and y-coordinate.

if (Math.*round*(fXWait) != Math.*round*(fXTab)){

return false;

} else {

return true;

}

}

.

**Any modifications to your specifications/release schedule:**

Release 1.1

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

“Scratch Guest”: Program tests the guests’ fundamental tasks such as walking down, waiting, indicating the loss of patiences, and more. Once you begin the scratch, the guest moves downwards and stops at the bottom of the screen. With 3 hearts above its’ head (indicating full level of satisfaction), it’ll begin to lose hearts the longer it waits in that position. The 3 hearts will recover if the guest is dragged to a table.

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

The guests’ timer. We needed to implement a timer that will later affect the guests’ hearts.

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

<https://stackoverflow.com/>

<https://libgdx.badlogicgames.com/ci/nightlies/docs/api/>

<https://github.com/DaphneLai/POLYGONE-Final/blob/master/core/src/gdx/objects/Shape.java>

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

These websites helped us utilize the most out of the Sprite class for our SprGuest. We managed to passover the guests’ X and Y coordinates, integrate only one timer instead of 5, and are on our way to program a pathway for the guest. For example:

public void walkDown() {

if (isWait == false) {

fY -= fDown + 10;

setY(fY);

if (fY <= 10) {

fDown = 0;

isWait = true;

canDrag = true;

}

}

Referencing off of code from other projects, we managed to make our character move under certain conditions.

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

Our code was very messy before. Public variables were used all the time and accessed over multiple screens when not necessary.

**Peer Assessment:**

Daphne - 100

Maddie - 100

Sarah - 100