**Work in Progress Report #3**

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

Server Hit Detection: Managed to get server to move around and stop at the back of the table (with a bit of an offset). After implementing an array of integers to suggest a coordinate system, There are less booleans and a better overall structure.

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

Multiple Guests: We got the guests to spawn every 5 seconds and they are all functional and able to drag but we still have two major bugs in that scratch. The one we are currently working on is being able to drag one guest at a time without them collecting when they overlap. The guests should be able to drag over each other and not affect each other. I have been looking at Ameer’s code as he solved that problem in a previous year.

//private boolean isMousedOver() {

vTouch = new Vector3(Gdx.input.getX(), Gdx.input.getY(), 0);

viewport.unproject(vTouch);

if (vTouch.x > (fX - 20) && vTouch.x < fX + (fW + 20)) {

if (vTouch.y > (fY - 20) && vTouch.y < fY + (fH + 20)) {

return true;

}

}

return false;

}

Right now our drag function is checking if the mouse is over the guest and that is what is causing it to collect the other guests.

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

N/A

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

Server Hit Detection: The scratch uses an array of int’s the delta X and delta Y of the server (arnDx and arnDY). The array is listed in a way that the index’s will go a certain direction:

private int arnDx[] = {0, 2, 0, -2, 0}; //Array of speed  
 private int arnDy[] = {0, 0, -2, 0, 2};

Index zero is neutral, index one is east, index two is south, index three is west, and index 4 is north. Within the “walk” method, the first thing it does is set the objtable x-coordinate, y-coordinate, width and height. There’s a check at the start of the function checking if the rounded coordinate of the table is odd or even. This is necessary because the server x-coordinate and y-coordinate is always even since we add the value of 2 whenever moving. Thus, when the code checks if the table coordinate and server coordinate is equal, it will never equal if the table coordinate is always odd. To prevent that I check it in advance and add the value of 1 to make it even. After all this code, the server will go east, south, west or north according to its coordinates compared to the clicked table. It is checking if it’s above or below the table and to the left or right of the table; and according to that the server will move accordingly.

//Makes server move up according to coordinate location

if (fX + fW == fTX && fY + fH != fTY) {

System.*out*.println("UP 2");

fX += arnDx[4];

fY += arnDy[4];

setX(fX);

setY(fY);

}

Example: If the server is to the right of the table and the server does not equal to the table Y-coordinate, then she will move up.

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

Server Hit Detection: The generic concept tested out was hit detection. Hit detection was used to prevent the server from going through the table. And, it was used to detect if the server hit a certain coordinate which if it is true, then the server will turn and go around the table.

**Source any website/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>

<https://github.com/Ameer-Mushani/Sort>

<http://3ui.sgrondin.ca>

<https://libgdx.badlogicgames.com/ci/nightlies/docs/api/com/badlogic/gdx/graphics/glutils/ShapeRenderer.html>

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

Server Hit Detection: I learned they’re many ways to approach a concept, some ways easier than others. In this case, using arrays is especially helpful in limiting the usage of booleans.

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

Server Hit Detection: The server would align perfectly with the table within the scratch, but not within the main game frame. This is a problem since the scrGame and sctWaiter use the same server sprite. Though now the server stops at the side of the table, it is better than before.

**Peer Assessment:**

Sarah: 100

Maddie: 100

Daphne: 100