**Work in Progress Report**

**Papa John’s**

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

* Enemies changing colors based on health
  + Texture txSpriteSheet = new Texture("spritesheets/guck.png");

TextureRegion[][] tmp = TextureRegion.*split*(txSpriteSheet, txSpriteSheet.getWidth() / 6, txSpriteSheet.getHeight());

TextureRegion[] arGreenframes = new TextureRegion[2];

for (int i = 0; i < 2; i++) {

arGreenframes[i] = tmp[0][i];

}

TextureRegion[] arYellowframes = new TextureRegion[2];

arYellowframes[0] = tmp[0][4];

arYellowframes[1] = tmp[0][5];

TextureRegion[] arRedframes = new TextureRegion[2];

arRedframes[0] = tmp[0][2];

arRedframes[1] = tmp[0][3];

animation = new Animation[3];

animation[0] = new Animation<TextureRegion>(1 / 5f, arGreenframes);

animation[1] = new Animation<TextureRegion>(1 / 5f, arYellowframes);

animation[2] = new Animation<TextureRegion>(1 / 5f, arRedframes);

setRegion(animation[0].getKeyFrame(0));

if (getHealth() <= fMaxhealth / 3) {

setRegion(animation[2].getKeyFrame(fElapsedTime, true));

} else if (getHealth() <= fMaxhealth / 3 \* 2) {

setRegion(animation[1].getKeyFrame(fElapsedTime, true));

} else {

setRegion(animation[0].getKeyFrame(fElapsedTime, true));

}

* Cleaned up bullet code(now uses vector math instead of trig)
  + vVelocity = vTargetPos.sub(new Vector2(getX(), getY())).nor().scl(fSpeed).rotate(fSprayAngle);
* Fixed direction movement
  + nDir = nDirection;  
     switch (nDirection) {  
     case 0:  
     vVelocity = (new Vector2(0, fSpeed));  
     break;  
     case 1:  
     vVelocity = (new Vector2(fSpeed, fSpeed));  
     break;  
     case 2:  
     vVelocity = (new Vector2(fSpeed, 0));  
     break;  
     case 3:  
     vVelocity = (new Vector2(fSpeed, -fSpeed));  
     break;  
     case 4:  
     vVelocity = (new Vector2(0, -fSpeed));  
     break;  
     case 5:  
     vVelocity = (new Vector2(-fSpeed, -fSpeed));  
     break;  
     case 6:  
     vVelocity = (new Vector2(-fSpeed, 0));  
     break;  
     case 7:  
     vVelocity = (new Vector2(-fSpeed, fSpeed));  
     Break;

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

* Trying to get enemies to dodge bullets was very difficult so we eventually gave up on it to work on other features

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

Added multiplayer even though we never intended to.

No longer going to make enemies dodge bullets

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

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

Online multiplayer

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

https://www.youtube.com/watch?v=uIPAaDslhPM&list=PLZm85UZQLd2Qh6r7jxBKPuB4hl-Xw5uZT

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

The server is made in Javascript using the Node.JS library. Communication between the server and the client is done using Web Sockets with the Socket.IO library.

The connectSocket() function connects the websocket to the server.

The configSocketEvents() function tells the socket object what to do when it receives various events from the server.

In the update() function, the client will send the player’s position to the server which allows the server to update their position on all other connected players’ screens.

On the server side, the server’s socket will broadcast different events such as player connection. The server has an array which holds data of all connected players such as their ID and position. The server also listens for the “playerMoved” event which updates the position of the player in the array and sends the updated position to all other connected players.

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

Right now, multiplayer isn’t completely functional (only player movement works, stuff like shooting doesn’t yet) so it hasn’t been implemented.

**Peer Assessment:**

Ethan: 100

John: 100

Danny: 100