**Work in Progress Report**

Major developments/breakthroughs(reference specific code please):

* Bullet hit detection
  + for (Bullet b : player.getBullets()) {

if (collisionHandler.findDistance(new Vector2(b.getX(), b.getY()), new Vector2(player.getX(), player.getY())) >= player.getGun().getRange()) {

player.getBullets().removeValue(b, true);

}

}

// Update enemy location and check for collisions

for (Enemy e : arEnemies) {

float fEStartX = e.getX();

float fEStartY = e.getY();

e.update(Gdx.*graphics*.getDeltaTime());

if (collisionHandler.isCollidingWithMap(e.getHitbox(), 2)) {

e.setPosition(fEStartX, fEStartY);

}

if (collisionHandler.findDistance(new Vector2(player.getX(), player.getY()), new Vector2(e.getX(), e.getY())) <= e.getRange()) {

e.setPlayerInRange(true);

e.setTargetPos(player.getX(), player.getY());

} else {

e.setPlayerInRange(false);

}

for (Bullet b : player.getBullets()) {

if (collisionHandler.isCollidingWithMap(b.getBoundingRectangle(), 2)) {

player.getBullets().removeValue(b, true);

}

if (collisionHandler.isSpriteColliding(b.getBoundingRectangle(), e.getHitbox())) {

player.getBullets().removeValue(b, true);

e.setHealth(-10);

}

}

for (Bullet b : e.getBullets()) {

if (b.getVelocity() == Vector2.*Zero*) {

b.setTargetPos(player.getX(), player.getY(), e.getGun().getSpray());

}

if (collisionHandler.isCollidingWithMap(b.getBoundingRectangle(), 2)) {

e.getBullets().removeValue(b, true);

}

if (collisionHandler.isSpriteColliding(b.getBoundingRectangle(), player.getHitbox())) {

e.getBullets().removeValue(b, true);

player.setHealth(-10);

}

}

if (collisionHandler.isSpriteColliding(player.getHitbox(), e.getHitbox())) {

e.setPosition(fEStartX, fEStartY);

player.setPosition(fStartX, fStartY);

}

if (e.getHealth() <= 0) {

arEnemies.removeValue((Guck) e, true);

}

}

if (collisionHandler.isCollidingWithMap(player.getHitbox(), 2)) {

player.setPosition(fStartX, fStartY);

}

healthBar.setValue(player.getHealth());

if (player.getHealth() <= 0) {

game.updateState(1);

}

cam.position.set(player.getX() + player.getWidth() / 2, player.getY() + player.getHeight() / 2, 0); // Set camera location to player's

cam.update();

}

* Health bar
  + public class HealthBar extends ProgressBar {

private BitmapFont font;

public HealthBar(int nWidth, int nHeight, float fMaxHealth) {

super(0.0f, fMaxHealth, 1f, false, new ProgressBarStyle());

getStyle().background = UIUtils.*getColoredDrawable*(nWidth, nHeight, Color.*LIGHT\_GRAY*);

getStyle().knob = UIUtils.*getColoredDrawable*(0, nHeight, Color.*GREEN*);

getStyle().knobBefore = UIUtils.*getColoredDrawable*(nWidth, nHeight, Color.*GREEN*);

setWidth(nWidth);

setHeight(nHeight);

setValue(fMaxHealth);

setAnimateDuration(0.25f);

font = new BitmapFont();

font.setColor(Color.*BLACK*);

}

public String getHealthString() {

return (int) getValue() + "/" + (int) getMaxValue();

}

public Drawable pickColourBasedOffHealth(int nWidth) {

if (getValue() <= getMaxValue() / 10) {

return UIUtils.*getColoredDrawable*(nWidth, (int) getHeight(), Color.*RED*);

} else if (getValue() <= getMaxValue() / 2) {

return UIUtils.*getColoredDrawable*(nWidth, (int) getHeight(), Color.*YELLOW*);

}

return UIUtils.*getColoredDrawable*(nWidth, (int) getHeight(), Color.*GREEN*);

}

@Override

public void draw(Batch batch, float parentAlpha) {

getStyle().knob = pickColourBasedOffHealth(0);

getStyle().knobBefore = pickColourBasedOffHealth((int) getStyle().knobBefore.getMinWidth());

setStyle(getStyle());

super.draw(batch, parentAlpha);

font.draw(batch, getHealthString(), getX(), getY() + font.getLineHeight());

}

}

Major Challenges/setbacks( reference specific code please):

-I(John) was working on working on have the enemies dodge player bullets but the structure of our code was making it difficult. The code also didn't push properly and it was lost

Any modifications to your specifications/release schedule:

We added UI much earlier than we intended to. (Health Bar)

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

Describe the generic concept you needed to test out:

Health Bar scratch: testing out the player health bar.

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

http://www.coding-daddy.xyz/node/20

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

Learned to work with progress bars

The health bar is a progress bar that does down with the player’s health. It then draws text on the bar showing the player’s health over their max health. It’s later drawn using Scene2D in the screen’s render function.

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

No challenges integrating scratch

**Peer Assessment:**

Ethan: 100 John 100 Danny: 100