Project 1 Journal

Name: Alex Bowes

Student Number: C00287604

Make an entry using the template below ***for each working session*** *on the Project.*

Session #1

Date: 25/02/24

Approximate time spent (hrs): 3 hours

Reflection on work completed:

During this session I made the timer that counts down from 60 using the frame rate I used a float so it was displaying the decimal points after the 0 but I look online for a way to stop this from happening. I had to include iomanip this gives me access to new functions like fixed and std::setprecisouion it was quite easy it took around 15 minutes.

```
float FrameRateMiliseconds = 16.67; //60 fps in miliseconds
m_time -= FrameRateMiliseconds / 1000.0; // Convert miliseconds to seconds


if (m_time <= 0) {
    m_time = 0;
    m_game_over = true;
}

std::stringstream m_TimeDigits;
m_TimeDigits << "Time: " << std::fixed << std::setprecision(0) << m_time;
m_timer.setString(m_TimeDigits.str());
```

The second part of the session I started to make a shooting function just to get the bullets setup before making mouse aiming and shooting. I wanted to challenge myself by trying to using only yaml data but its quite tricky to work with because of the structure and layout of the .yaml file any missing spaces or syntax's would cause an expectation to be thrown here:

```cpp
int currentLevel = 1;

try {
    LevelLoader::load(currentLevel, m_level);
}
catch (std::exception& e) {
    std::cout << "Level Loading Failed" << std::endl;
    std::cout << e.what() << std::endl;
    throw e;
}
```

Here is my Projectiles Data in .Yaml:

```yaml
projectiles:
  - type: bullet
    position: {x: 850, y: 100}
    scale: {x: 0.5, y: 0.5}
    rotation: 0
  - type: bullet
    position: {x: 850, y: 100}
    scale: {x: 0.5, y: 0.5}
    rotation: 0
  - type: bullet
    position: {x: 850, y: 100}
    scale: {x: 0.5, y: 0.5}
    rotation: 0
  - type: bullet
    position: {x: 850, y: 100}
    scale: {x: 0.5, y: 0.5}
    rotation: 0
```

Link to first commit : added time and drawing bullets · alexbowes51/2D-GameplayProgramming@ba1ed84 (github.com)

Session #2

Date: 26/02/24

Approximate time spent (hrs): 2 hours

Reflection on work completed:

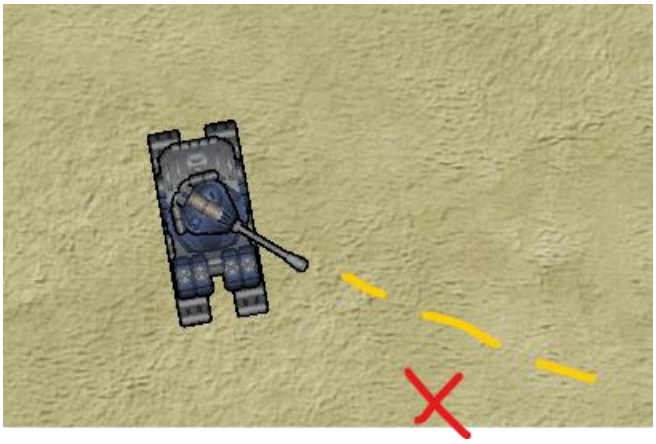
Session 2 I worked on getting the turret to move with my mouse position it was quite challenging but I got it to work its just slightly off but when I try to add degrees to the AngleDegrees variable it offsets the rotation of the turret . im a bit annoyed about not getting the mouse aiming perfect.

Here is the function responsible for the tanks turret rotation :

```
void Tank::tankAimSystem()
{
    sf::Vector2i mouseLocation = sf::Mouse::getPosition();
    // Calculate the direction vector from turrent to mouse
    sf::Vector2f direction = sf::Vector2f(mouseLocation.x, mouseLocation.y) - m_turret.getPosition();

    // Calculate the length of the direction vector
    float length = std::sqrt(direction.x * direction.x + direction.y * direction.y);

    // Calculate the angle between the turrent and the mouse cursor using atan2
    float AngleRadians = std::atan2(direction.y, direction.x);
    float AngleDegrees = AngleRadians * 180 / 3.14; // Convert radians to degrees

    m_turret.setRotation(AngleDegrees); // Adjust sprite orientation

    if (!m_Fire) {
        for (auto& projectile : m_ProjectileSprites) {
            projectile.setRotation(AngleDegrees);
        }
    }

}
```

Here is what it would look like in game :

(red X is the mouse location)



Link to second commit :

Session #3

Date: 26/02/24

Approximate time spent (hrs): 5 hours

Reflection on work completed:

This Session tried to make the move the mouse position when I clicked left mouse button . the problem was moving the bullet it would always follow my mouse and not the mouse click position that took me 2 hours . I ended up making 2 bools one for aiming and one for aiming , aiming is always true but fire is only true once you press the left mouse button and I added collision dectection for the bullets so when the bullet collidies with wnythinf fire with but = false and the bullet will be set back to the centre of the player turret .The logic and method was hard to wrap around but in the end im happy with the result. Then for the rest of the session I added the hit and miss ints to make it easier for me later to count up the score.

This is the way I move the bullets:

```
if (m_aiming && m_Fire) {
    for (auto& projectile : m_ProjectileSprites) {
        projectile.move(m_BulletVelocity);
    }
}
```

This is the function from session 2:

```
void Tank::tankAimSystem()
{
    sf::Vector2i mouseLocation = sf::Mouse::getPosition();
    // Calculate the direction vector from turrent to mouse
    sf::Vector2f direction = sf::Vector2f(mouseLocation.x, mouseLocation.y) - m_turret.getPosition();

    // Calculate the length of the direction vector
    float length = std::sqrt(direction.x * direction.x + direction.y * direction.y);

    // Calculate the angle between the turrent and the mouse cursor using atan2
    float AngleRadians = std::atan2(direction.y, direction.x);
    float AngleDegrees = AngleRadians * 180 / 3.14; // Convert radians to degrees

    m_turret.setRotation(AngleDegrees); // Adjust sprite orientation

    if (!m_Fire) {
        for (auto& projectile : m_ProjectileSprites) {
            projectile.setRotation(AngleDegrees);
        }
    }

}
```

Link to Third commit :

Session #4

Date: 26/02/24

Approximate time spent (hrs): 10+ hours

Reflection on work completed:

This session was very difficult for me because i made a error that would cause me to get stuck on enemys for a few days it was such a simple mistake and I know now to be very careful with yaml data But first I added so when the timer hits zero it showed the game over + added place holder hit and miss counters . The part the made me get stuck was the levelloader and yaml files it was the yaml data because I forgot to add a , between the x , y and it threw an expection every time I didn't spot the mistake until the last lab so for this session I commeted out the code. I put the counters into tank because it was easier to access the data of the tank I also added the accuracy calculator but I doesn't work well . I get the amount of shots and divide but the total shots hits but it always gave me a low number like 30% or 25%.

Game.cpp:

```
m_timer.setFont(m_arialFont);
m_timer.setCharacterSize(25U);
m_timer.setPosition(1150.0f, 25.0f);
m_timer.setFillColor(sf::Color::White);
m_timer.setString("Time : " + std::to_string(m_time));

m_GameOver.setFont(m_arialFont);
m_GameOver.setCharacterSize(100U);
m_GameOver.setPosition(400.0f, 350.0f);
m_GameOver.setFillColor(sf::Color::White);
m_GameOver.setString("GameOver");;
```

Tank.cpp:

```
m_Hits.setFont(m_arialFont);
m_Hits.setCharacterSize(25U);
m_Hits.setPosition(655.0f, 25.0f);
m_Hits.setFillColor(sf::Color::White);
m_Hits.setString("Hits : " + std::to_string(m_hits));


m_Shot.setFont(m_arialFont);
m_Shot.setCharacterSize(25U);
m_Shot.setPosition(355.0f, 25.0f);
m_Shot.setFillColor(sf::Color::White);
m_Shot.setString("Shots : " + std::to_string(m_shots));


m_Misses.setFont(m_arialFont);
m_Misses.setCharacterSize(25U);
m_Misses.setPosition(955.0f, 25.0f);
m_Misses.setFillColor(sf::Color::White);
m_Misses.setString("Misses : " + std::to_string(m_misses));
```

Where the , was missing:

```
enemies:
  - type: turret
    position: {x: 500 : 500, randomOffset: 300}
    rotation : 0
    scale: {x: 0.3, y: 0.4}
  - type: turret
    position: {x: 800,y: 800, randomOffset: 100}
    rotation : 0
    scale: {x: 0.4, y: 0.2}
  - type: turret
    position: {x: 400,y: 800, randomOffset: 400}
    rotation : 0
    scale: {x: 0.2, y: 0.3}
  - type: turret
    position: {x: 200,y: 300, randomOffset: 400}
    rotation : 0
    scale: {x: 0.4, y: 0.3}
  - type: static
    position: {x: 700,y: 300, randomOffset: 400}
    rotation : 0
    scale: {x: 0.4, y: 0.3}
  - type: static
    position: {x: 800,y: 300, randomOffset: 400}
    rotation : 0
    scale: {x: 0.4, y: 0.3}
```

Calculating the accuracy:

```
if (m_tank.m_shots > 0) {
     m_tank.m_accruacy = (static_cast<float>(m_tank.m_shots) * 100 / m_tank.m_hits);

    std::stringstream m_AccuracyDigits;
    m_AccuracyDigits << "Accuracy = " << std::fixed << std::setprecision(0) << m_tank.m_accruacy << " %";
    m_Accuracy.setString(m_AccuracyDigits.str());
}
else {
    m_Accuracy.setString("Accuracy = 0.00%");
}

if (m_tank.m_accruacy > 100) {
    m_Accuracy.setString("Accuracy = 100.00%");
}
```

I wanted it in a float to look fancy but I think the std:;fixed and setprecision messed with the calculations.

Session #5

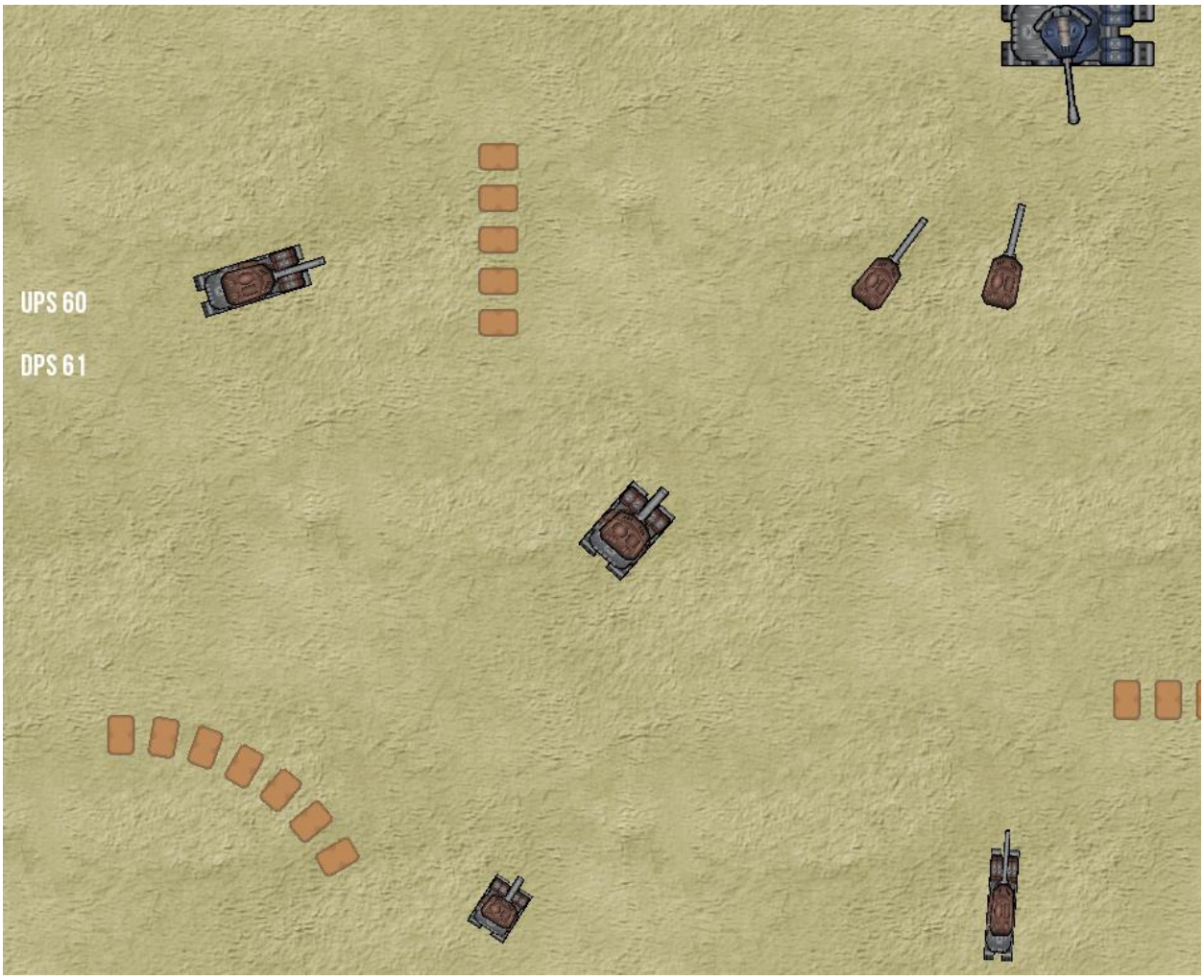
Date: 29/02/24

Approximate time spent (hrs): 8 hours

Reflection on work completed:

This session was a good one I found the error that had me stuck on loading the enemys data from the yaml file and then I set them up by drawing a tank for the turrets (full tanks) and static(turrets) by there yaml type.

```cpp
void Game::setupenemys()
    {
        sf::Texture& m_texture6 = m_holder["tankAtlas"];


        sf::IntRect enemybase(247, 0, 224, 116);
        sf::IntRect enemyturret(279, 114, 213, 96);

    for (auto const& enemyData : m_level.m_enemies) {
        if (enemyData.m_type == "turret") {
            sf::Sprite sprite1;
            sprite1.setTexture(m_texture6);
            sprite1.setTextureRect(enemybase);
            sprite1.setOrigin(100.5, 57);
            sprite1.setPosition(enemyData.m_position);
            sprite1.setRotation(enemyData.m_rotation);
            sprite1.setScale(enemyData.m_scale);
            m_EnemySprites.push_back(sprite1);


            sf::Sprite sprite2;
            sprite2.setTexture(m_texture6);
            sprite2.setTextureRect(enemyturret);
            sprite2.setOrigin(51, 45);
            sprite2.setPosition(enemyData.m_position);
            sprite2.setRotation(enemyData.m_rotation);
            sprite2.setScale(enemyData.m_scale);
            m_EnemySprites.push_back(sprite2);
        }
        if (enemyData.m_type == "static") {
            sf::Sprite sprite3;
            sprite3.setTexture(m_texture6);
            sprite3.setTextureRect(enemyturret);
            sprite3.setOrigin(51, 45);
            sprite3.setPosition(enemyData.m_position);
            sprite3.setRotation(enemyData.m_rotation);
            sprite3.setScale(enemyData.m_scale);
            m_EnemySprites.push_back(sprite3);

        }
```



I wanted to add a way of spawning them one by one but I didn't have time to get this done bit disappointed. I then added collisions and hit and misses to the right collisions. I then moved the text around to make it more nicer. I tried to get the enemys shooting but couldn't .

Final commit link :