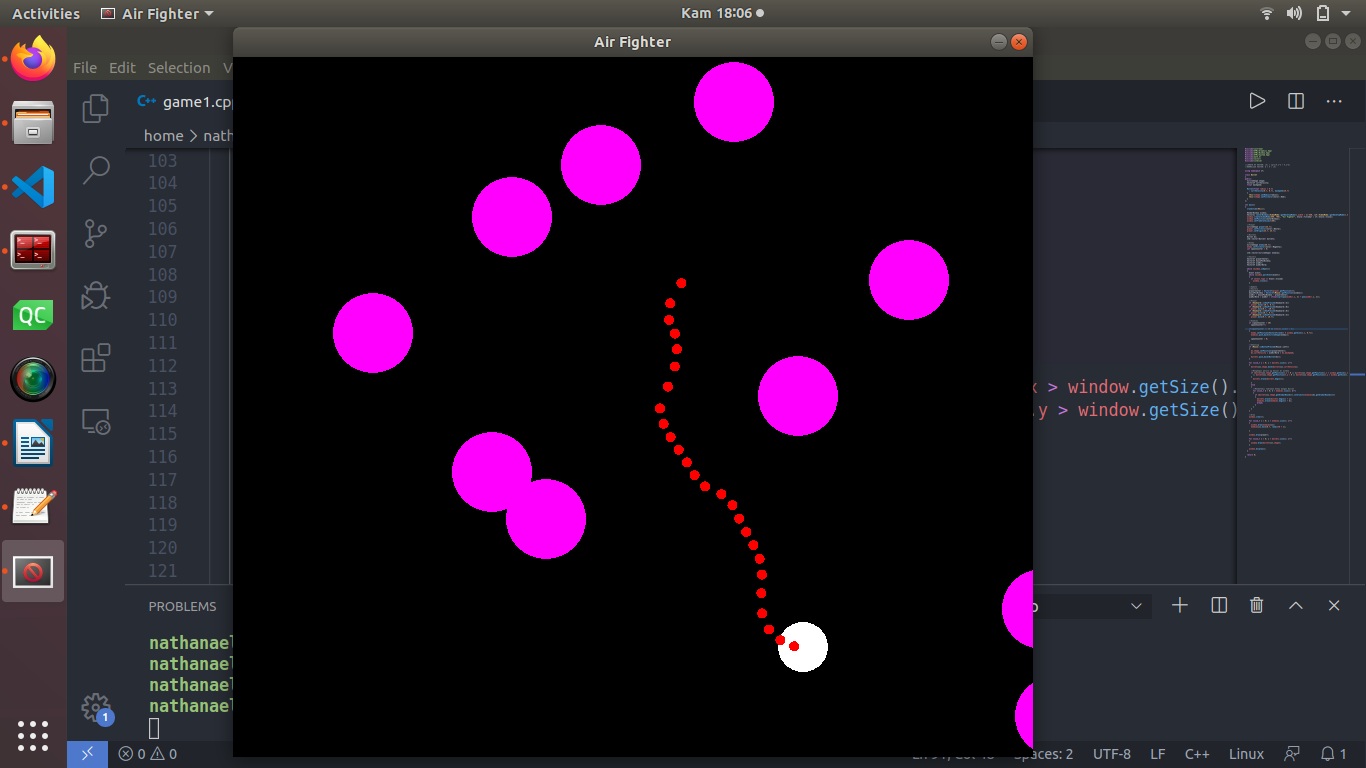
Nama : Nathanael Hutama Harsono

NRP : 07211940000044

Tanggal : 26/3/2020

**1. Hasil :**

****

**2. Program :**

#include<iostream>

#include<SFML/Graphics.hpp>

#include<math.h>

#include<vector>

#include<cstdlib>

using namespace sf;

class Bullet

{

public:

CircleShape shape;

Vector2f currVelocity;

float maxSpeed;

Bullet(float radius = 5.f)

: currVelocity(0.f, 0.f), maxSpeed(15.f)

{

shape.setRadius(radius);

shape.setFillColor(Color::Red);

}

};

int main()

{

srand(time(NULL));

RenderWindow window;

Vector2i centerWindow((VideoMode::getDesktopMode().width / 2)-450, (sf::VideoMode::getDesktopMode().height / 2)-390);

window.create(VideoMode(800, 700), "Air Fighter", Style::Titlebar | sf::Style::Close);

window.setPosition(centerWindow);

window.setFramerateLimit(60);

//Player

CircleShape player(25.f);

player.setFillColor(Color::White);

player.setOrigin(25.f, 25.f);

//Bullets

Bullet b1;

std::vector<Bullet> bullets;

//Enemy

CircleShape enemy(40.f);

enemy.setFillColor(Color::Magenta);

int spawnCounter = 0;

std::vector<CircleShape> enemies;

//Vectors

Vector2f playerCenter;

Vector2f mousePosWindow;

Vector2f aimDir;

Vector2f aimDirNorm;

while (window.isOpen())

{

Event event;

while (window.pollEvent(event))

{

if (event.type == Event::Closed)

window.close();

}

//Update

//Vectors

playerCenter = Vector2f(player.getPosition());

mousePosWindow = Vector2f(Mouse::getPosition(window));

aimDir = mousePosWindow - playerCenter;

aimDirNorm = aimDir / (float)sqrt(pow(aimDir.x, 2) + pow(aimDir.y, 2));

//Player

if (Keyboard::isKeyPressed(Keyboard::A))

player.move(-10.f, 0.f);

if (Keyboard::isKeyPressed(Keyboard::W))

player.move(0.f, -10.f);

if (Keyboard::isKeyPressed(Keyboard::D))

player.move(10.f, 0.f);

if (Keyboard::isKeyPressed(Keyboard::S))

player.move(0.f, 10.f);

//Enemies

if (spawnCounter < 20)

spawnCounter++;

if(spawnCounter >= 20 && enemies.size() < 20)

{

enemy.setPosition(Vector2f(rand() % window.getSize().x, 0.f));

enemies.push\_back(CircleShape(enemy));

spawnCounter = 0;

}

//Shooting

if (Mouse::isButtonPressed(Mouse::Left))

{

b1.shape.setPosition(playerCenter);

b1.currVelocity = aimDirNorm \* b1.maxSpeed;

bullets.push\_back(Bullet(b1));

}

for (int i = 0; i < bullets.size(); i++)

{

bullets[i].shape.move(bullets[i].currVelocity);

//Menghapus peluru yg keluar dr window

if (bullets[i].shape.getPosition().x < 0 || bullets[i].shape.getPosition().x > window.getSize().x

|| bullets[i].shape.getPosition().y < 0 || bullets[i].shape.getPosition().y > window.getSize().y)

{

bullets.erase(bullets.begin());

}

else

{

//Mendeteksi tabrakan Enemy dengan Bullet

for (int k = 0; k < enemies.size(); k++)

{

if (bullets[i].shape.getGlobalBounds().intersects(enemies[k].getGlobalBounds()))

{

bullets.erase(bullets.begin() + i);

enemies.erase(enemies.begin() + k);

break;

}

}

}

}

//Draw

window.clear();

for (int i = 0; i < enemies.size(); i++)

{

window.draw(enemies[i]);

enemies[i].move(0.f, rand()%5 + 1);

}

window.draw(player);

for (int i = 0; i < bullets.size(); i++)

{

window.draw(bullets[i].shape);

}

window.display();

}

return 0;

}