#include <iostream>

#include <string>

#include <SFML/Graphics.hpp>

#include <Windows.h>

void HideConsole() {

ShowWindow(GetConsoleWindow(), SW\_HIDE);

}

class TicTacToe {

private:

sf::Image background\_i;

sf::Texture background\_t;

sf::Image board\_i;

sf::Texture board\_t;

sf::Image reset\_i;

sf::Texture reset\_t;

sf::Image cross\_i;

sf::Texture cross;

sf::Image circle\_i;

sf::Texture circle;

sf::Image blank\_i;

sf::Texture blank;

sf::Font font;

std::string msg;

int cur;

bool waitForReset;

bool checkWin(int placement);

bool checkDraw();

public:

sf::Sprite background;

sf::Sprite board;

sf::Sprite reset;

sf::Sprite pieces[9];

sf::Text text;

char set[9];

bool loadAssets();

bool loadBoard(int startPlayer);

void keyPress(sf::Vector2f pos);

};

bool TicTacToe::loadAssets() {

if (!this->background\_i.loadFromFile("Resources\\background.png"))

return false;

if (!this->background\_t.loadFromImage(this->background\_i))

return false;

this->background.setTexture(this->background\_t);

if (!this->board\_i.loadFromFile("Resources\\board.png"))

return false;

if (!this->board\_t.loadFromImage(this->board\_i))

return false;

this->board.setTexture(this->board\_t);

this->board.setPosition(sf::Vector2f(0.0f, 100.0f));

if (!this->reset\_i.loadFromFile("Resources\\reset.png"))

return false;

if (!this->reset\_t.loadFromImage(this->reset\_i))

return false;

this->reset.setTexture(this->reset\_t);

this->reset.setPosition(sf::Vector2f(25.0f, 525.0f));

if (!this->circle\_i.loadFromFile("Resources\\circle.png"))

return false;

if (!this->circle.loadFromImage(this->circle\_i))

return false;

if (!this->cross\_i.loadFromFile("Resources\\cross.png"))

return false;

if (!this->cross.loadFromImage(this->cross\_i))

return false;

if (!this->blank\_i.loadFromFile("Resources\\blank.png"))

return false;

if (!this->blank.loadFromImage(this->blank\_i))

return false;

if (!this->font.loadFromFile("Resources\\font.ttf"))

return false;

this->text.setFont(this->font);

this->text.setCharacterSize(40);

text.setFillColor(sf::Color::Black);

this->text.setPosition(sf::Vector2f(25.0f, 25.0f));

return true;

}

bool TicTacToe::loadBoard(int startPlayer) {

memset(this->set, 0, 9);

this->waitForReset = false;

this->pieces[0].setPosition(sf::Vector2f(10.0f, 110.0f));

this->pieces[1].setPosition(sf::Vector2f(150.0f, 110.0f));

this->pieces[2].setPosition(sf::Vector2f(290.0f, 110.0f));

this->pieces[3].setPosition(sf::Vector2f(10.0f, 250.0f));

this->pieces[4].setPosition(sf::Vector2f(150.0f, 250.0f));

this->pieces[5].setPosition(sf::Vector2f(290.0f, 250.0f));

this->pieces[6].setPosition(sf::Vector2f(10.0f, 390.0f));

this->pieces[7].setPosition(sf::Vector2f(150.0f, 390.0f));

this->pieces[8].setPosition(sf::Vector2f(290.0f, 390.0f));

for (int i = 0; i < 9; i++)

this->pieces[i].setTexture(this->blank);

this->cur = startPlayer;

this->msg = std::string(startPlayer == 1 ? "It is player 1's turn" : "It is player 2's turn");

this->text.setString(this->msg);

return true;

}

bool TicTacToe::checkWin(int placement) {

int row = (placement / 3) \* 3;

int columm = placement % 3;

if (this->set[row] == this->cur && this->set[row + 1] == this->cur && this->set[row + 2] == this->cur)

return true;

if (this->set[columm] == this->cur && this->set[columm + 3] == this->cur && this->set[columm + 6] == this->cur)

return true;

if (this->set[0] == this->cur && this->set[4] == this->cur && this->set[8] == this->cur)

return true;

if (this->set[2] == this->cur && this->set[4] == this->cur && this->set[6] == this->cur)

return true;

return false;

}

bool TicTacToe::checkDraw() {

for (int i = 0; i < 9; i++)

if (this->set[i] == 0)

return false;

return true;

}

void TicTacToe::keyPress(sf::Vector2f pos) {

if (!this->waitForReset) {

for (int i = 0; i < 9; i++) {

if (this->pieces[i].getGlobalBounds().contains(pos)) {

if (this->set[i] == 0) {

this->pieces[i].setTexture(this->cur == 1 ? this->cross : this->circle);

this->set[i] = this->cur;

if (this->checkWin(i)) {

this->waitForReset = true;

this->msg = this->cur == 1 ? "Player 1 wins!" : "Player 2 wins!";

this->text.setString(this->msg);

}

else {

if (this->checkDraw()) {

this->waitForReset = true;

this->msg = "Draw. Nobody wins.";

this->text.setString(this->msg);

}

else {

this->cur = this->cur == 1 ? 2 : 1;

this->msg = this->cur == 1 ? "It is player 1's turn" : "It is player 2's turn";

this->text.setString(this->msg);

}

}

}

break;

}

}

}

if (this->reset.getGlobalBounds().contains(pos))

this->loadBoard(this->cur == 1 ? 2 : 1);

}

int main() {

TicTacToe Game;

if (!Game.loadAssets()) {

printf("Unable to load game assets\n");

system("pause");

return 0;

}

if (!Game.loadBoard(1)) {

printf("Unable to load game\n");

system("pause");

return 0;

}

HideConsole();

sf::RenderWindow window(sf::VideoMode(400, 600), "Tic Tac Toe");

while (window.isOpen()) {

window.clear();

window.draw(Game.background);

window.draw(Game.board);

window.draw(Game.text);

window.draw(Game.reset);

for (int i = 0; i < 9; i++)

window.draw(Game.pieces[i]);

window.display();

sf::Event event;

while (window.pollEvent(event)) {

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

window.close();

if (event.type == sf::Event::MouseButtonPressed)

if(event.mouseButton.button == sf::Mouse::Button::Left)

Game.keyPress(window.mapPixelToCoords(sf::Mouse::getPosition(window)));

}

}

return 0;

}