Федеральное агенство связи

Федеральное Государственное Бюджетное образовательное учреждение высшего образования

«Сибирский Государственный Университет Телекомуникаций и Информатики»

КУРСОВАЯ РАБОТА

по дисциплине «Программирование»

Вариант – “Понг”

Выполнил:

Проверил: ассистент кафедры ПмиК

Дьячкова Ирина Сергеевна

Содержание:

1)Введение  
2)Описание используемых алгоритмов и функций   
3)Код программы на языке C++  
4)Результат работы программы

**1. Введение**

***Задача:*** сделать аркадную видеоигру на двоих игроков – “Понг”

***Pong*** — одна из ранних [аркадных](https://ru.wikipedia.org/wiki/%D0%90%D1%80%D0%BA%D0%B0%D0%B4%D0%B0_(%D0%B8%D0%B3%D1%80%D0%B0)) [видеоигр](https://ru.wikipedia.org/wiki/%D0%92%D0%B8%D0%B4%D0%B5%D0%BE%D0%B8%D0%B3%D1%80%D0%B0). Это теннисная спортивная игра с использованием [двумерной графики](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D0%BA%D0%B0), разработанная и выпущенная фирмой [Atari](https://ru.wikipedia.org/wiki/Atari) в [1972](https://ru.wikipedia.org/wiki/1972_%D0%B3%D0%BE%D0%B4_%D0%B2_%D0%BA%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D1%8B%D1%85_%D0%B8%D0%B3%D1%80%D0%B0%D1%85) годe. *Pong* называют первой в истории коммерчески успешной видеоигрой, а с её именем связывают появление [индустрии интерактивных развлечений](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D0%B4%D1%83%D1%81%D1%82%D1%80%D0%B8%D1%8F_%D0%BA%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D1%8B%D1%85_%D0%B8%D0%B3%D1%80).

**2. Описание используемых алгоритмов и функций**

Программа выполнена с использованием графической мультимедийной библиотеки SFML 2.5.1

* class Player – класс, где отображены координаты игроков(position.x и position.y), спрайт с “ракеткой”(sf::RectangleShape player\_bat), скорость движения ракеток(move\_speed) и количество очков игрока(int score). Методы void move\_up() и void move\_down() отвечают за движение вверх и вниз соответственно. Метод void update\_position() – отвечает за обновление координат “ракетки”. А метод restart\_position() – возвращает игроков на исходные позиции, когда-то кто-то из них выигрывает
* class Ball – класс, где отображен мяч, которым играют игроки: position.x и position.y отвечают за текущее расположение мяча, direction.x и direction.y отвечают за направление движения мяча, sf::CircleShape ball является спрайтом мяча, move\_speed – скорость движение мяча. Метод void move() осуществляет движение мяча по игровому полю, а метод  void update\_position() осуществляет обновление координат и направление движения(содержит в себе проверки на пересечение с “ракетками” игроков)
* Функция float rand\_float(float a, float b) – генерирует случайное вещественное число, нужна для создания направления движения мячика.

**3.  Код программы на языке C++**

#include <SFML/Audio.hpp>

#include <SFML/Graphics.hpp>

#include <cmath>

#define PI 3.14159265

float rand\_float(float a, float b) {

  return ((b - a) \* ((float)rand() / RAND\_MAX)) + a;

}

using namespace sf;

class Player {

public:

  Player(float x, float y) {

    position.x = x;

    position.y = y;

    player\_bat.setFillColor(sf::Color::White);

    player\_bat.setSize(sf::Vector2f(20, 100));

  }

  void update\_position() {

    if (position.y < 0)

      position.y = 0;

    if (position.y > 600)

      position.y = 600;

    player\_bat.setPosition(position);

  }

  void move\_up() { position.y -= move\_speed; }

  void move\_down() { position.y += move\_speed; }

  RectangleShape get\_shape() { return player\_bat; }

  void restart\_position() {

    position.y = 250;

    player\_bat.setPosition(position);

  }

  float move\_speed = 5;

  sf::RectangleShape player\_bat;

  sf::Vector2f position;

  int score = 0;

};

Player player1 = Player(50, 250);

Player player2 = Player(925, 250);

class Ball {

public:

  sf::CircleShape ball;

  sf::Vector2f position;

  sf::Vector2f direction;

  float move\_speed = 7;

  Ball(float x, float y) {

    ball.setFillColor(sf::Color::Red);

    ball.setRadius(15);

    position.x = x;

    position.y = y;

    float angle = rand\_float(0, 360 \* PI/180);

    direction.x = sin(angle);

    direction.y = cos(angle);

  }

  void move() {

    position.x += direction.x \* move\_speed;

    position.y += direction.y \* move\_speed;

  }

  void update\_position() {

    if (position.y < 0) {

      direction.y \*= -1;

    }

    if (position.y > 670) {

      direction.y \*= -1;

    }

    if (ball.getGlobalBounds().intersects(

            player1.get\_shape().getGlobalBounds())) {

      float new\_angle = rand\_float(-45 \* PI / 180, 45 \* PI / 180);

      direction.x = cos(new\_angle);

      direction.y = sin(new\_angle);

      position.x += 10.1;

    }

    if (ball.getGlobalBounds().intersects(

            player2.get\_shape().getGlobalBounds())) {

      float new\_angle = rand\_float(135 \* PI / 180, 225 \* PI / 180);

      direction.x = cos(new\_angle);

      direction.y = sin(new\_angle);

      position.x -= 10.1;

    }

    if (position.x <= 25) {

      player1.restart\_position();

      player2.restart\_position();

      float new\_angle = rand\_float(-45 \* PI / 180, 45 \* PI / 180);

      direction.x = cos(new\_angle);

      direction.y = sin(new\_angle);

      position.x = 500;

      position.y = 350;

      player2.score++;

    }

    if (position.x >= 950) {

      player1.restart\_position();

      player2.restart\_position();

      float new\_angle = rand\_float(135 \* PI / 180, 225 \* PI / 180);

      direction.x = cos(new\_angle);

      direction.y = sin(new\_angle);

      position.x = 500;

      position.y = 350;

      player1.score++;

    }

    ball.setPosition(position);

  }

  sf::CircleShape get\_shape() { return ball; }

};

int main() {

  sf::Music music;

  if (!music.openFromFile("music.ogg"))

    return -1; // error

  srand(time(NULL));

  music.play();

  music.setLoop(true);

  sf::Font font;

  font.loadFromFile("font.ttf");

  sf::Text score1(std::to\_string(player1.score), font);

  sf::Text score2(std::to\_string(player2.score), font);

  score1.setCharacterSize(30);

  score2.setCharacterSize(30);

  score1.setColor(sf::Color::White);

  score2.setColor(sf::Color::White);

  score1.setPosition(250, 20);

  score2.setPosition(750, 20);

  sf::RenderWindow window(sf::VideoMode(1000, 700), "Pong");

  window.setVerticalSyncEnabled(true);

  window.setFramerateLimit(60);

  RectangleShape line;

  line.setSize(sf::Vector2f(10, 700));

  line.setPosition(500, 0);

  sf::RectangleShape left\_border\_line;

  left\_border\_line.setSize(sf::Vector2f(10, 700));

  left\_border\_line.setPosition(0, 0);

  sf::RectangleShape right\_border\_line;

  right\_border\_line.setSize(sf::Vector2f(10, 700));

  right\_border\_line.setPosition(985, 0);

  player1.update\_position();

  player2.update\_position();

  Ball ball = Ball(500, 350);

  while (window.isOpen()) {

    score1.setString(std::to\_string(player1.score));

    score2.setString(std::to\_string(player2.score));

    sf::Event event;

    while (window.pollEvent(event)) {

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

        window.close();

    }

    if (Keyboard::isKeyPressed(Keyboard::Up)) {

      player2.move\_up();

      player2.update\_position();

    }

    if (Keyboard::isKeyPressed(Keyboard::Down)) {

      player2.move\_down();

      player2.update\_position();

    }

    if (Keyboard::isKeyPressed(sf::Keyboard::W)) {

      player1.move\_up();

      player1.update\_position();

    }

    if (Keyboard::isKeyPressed(sf::Keyboard::S)) {

      player1.move\_down();

      player1.update\_position();

    }

    if (Keyboard::isKeyPressed(sf::Keyboard::Escape)) {

      window.close();

    }

    ball.move();

    ball.update\_position();

    window.clear();

    window.draw(player1.get\_shape());

    window.draw(player2.get\_shape());

    window.draw(line);

    window.draw(left\_border\_line);

    window.draw(right\_border\_line);

    window.draw(ball.get\_shape());

    window.draw(score1);

    window.draw(score2);

    window.display();

  }

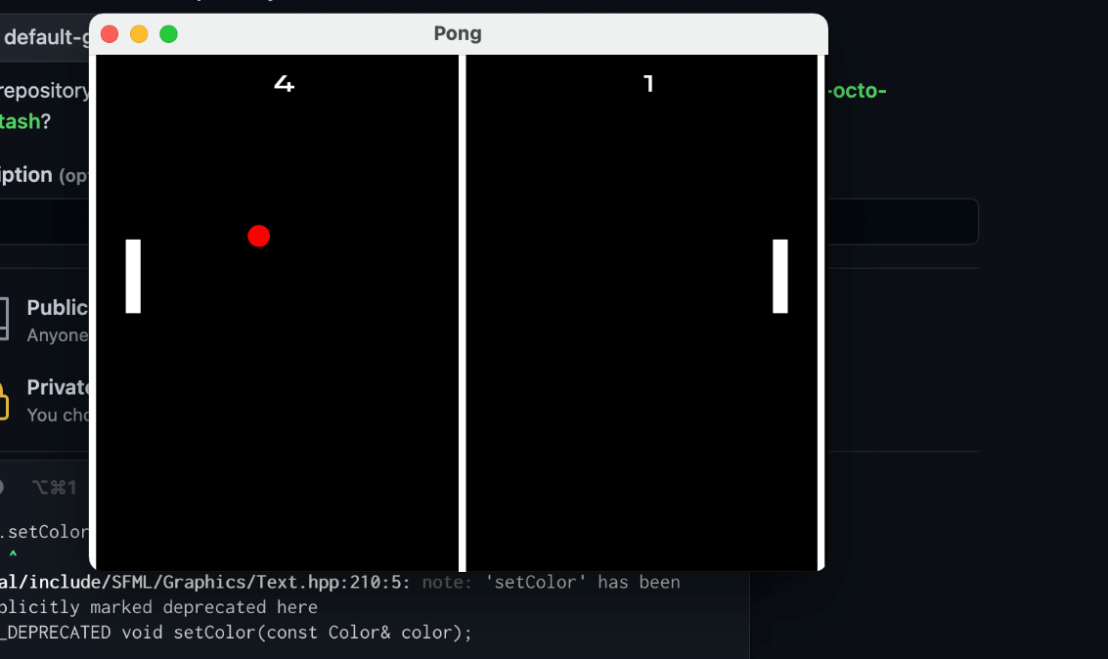
  return 0;

}

**4. Результат работы программы**

Управление для первого игрока осуществляется через клавиши W и S,

а для второго стрелочками “Вверх” и “Вниз”

****

