**Министерство науки и высшего образования Российской Федерации**

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**ПЕТРОЗАВОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**Физико-технический институт**

**Командное задание в рамках курсового проекта «Разработка игры «Pacman++»»**

**Индивидуальный отчет**

**Выполнил:**

Студент

Физико-технического Института

группы 21418

Ботько Н.Д.

**Проверил:**

Преподаватель:

Бульба Артем Владимирович

**Петрозаводск 2023**

# Цель работы

Разработать аналог игры «Pacman», добавив в не функционал создания призраками, бомбы. Вести разработку с помощью системы контроля версий Git через веб-сервис GitHub. Закрепить полученные знания использования языков С++ с использованием библиотеки SFML и UML.

# Задачи

1. Познакомиться с принципами работы в команде.
2. Научиться использовать при работе систему контроля версии Git.
3. Научиться строить UML-диаграммы.
4. Научиться использовать SFML библиотеку в программной среде С++.
5. Написание отчёта о проделанной работе.

# Кратко о программной реализации

Программная реализация выполнена на языке c++ с использованием библиотеки SFML, среда разработки - IDE Qt Creator (Qt 5.4.2)

# Единицы компиляции:

pacman.h

pacman.cpp

Класс описывает сущность пакмана и его поведение.

bomb.h

bomb.cpp

Класс описывает сущность бомбы и ее поведение.

# Описание сюжета игры

Игрок управляет пакманом, передвигаясь по карте и избегая призраков. Целью игры является избегание контакта с призраком и сбор всей еды(точек). При съедении большой точки, у призраков включается состояние “испуг”, в котором его можно скушать, за что даются очки, а сам призрак начнет свое движение со своего домика. При сборе всех точек, начинается новый уровень.

# Заголовочные файлы

## bomb.h

#ifndef BOMB\_H

#define BOMB\_H

#include "game\_types.h"

class bomb

{

public:

   position cur\_position;

   unsigned short animation\_timer;

   unsigned short lifetime;

   bool exploded;

   bomb();

   void draw(sf::RenderWindow &window);

   bool pacman\_collision(const position &pacman\_position);

};

#endif // BOMB\_H

## pacman.h

#ifndef PACMAN\_H

#define PACMAN\_H

#include "game\_types.h"

#include "score.h"

class pacman

{

public:

   bool is\_animation\_over;

   bool is\_dead;

   byte direction;

   unsigned short animation\_timer;

   unsigned short energizer\_timer;

   position cur\_position;

   score score;

   pacman();

   void draw(bool victory, sf::RenderWindow &window);

   void reset();

   void update(byte level, game\_map &map);

};

#endif // PACMAN\_H

# Исполнительные файлы

## bomb.cpp

#include "bomb.h"

#include <math.h>

bomb::bomb() : cur\_position({0, 0}),

              animation\_timer(0),

              lifetime(0),

              exploded(0)

{

}

void bomb::draw(sf::RenderWindow &window)

{

   byte frame = (byte)(floor(animation\_timer / (float)(BOMB\_ANIMATION\_SPEED)));

   sf::Sprite sprite;

   sf::Texture texture;

   texture.loadFromFile("img/bomb" + std::to\_string(CELL\_SIZE) + ".png");

   sprite.setPosition(cur\_position.x, cur\_position.y);

   sprite.setTexture(texture);

   lifetime++;

   if (lifetime > BOMB\_MAX\_LIFETIME)

   {

       sprite.setTextureRect(sf::IntRect(CELL\_SIZE \* (BOMB\_BOOM\_FRAME - 1), 0, CELL\_SIZE, CELL\_SIZE));

       exploded = true;

   }

   else

   {

       sprite.setTextureRect(sf::IntRect(CELL\_SIZE \* frame, 0, CELL\_SIZE, CELL\_SIZE));

   }

   window.draw(sprite);

   animation\_timer = (1 + animation\_timer) % (BOMB\_ANIMATION\_FRAMES \* BOMB\_ANIMATION\_SPEED);

}

bool bomb::pacman\_collision(const position &pacman\_position)

{

   if (cur\_position.x > pacman\_position.x - CELL\_SIZE && cur\_position.x < CELL\_SIZE + pacman\_position.x)

       if (cur\_position.y > pacman\_position.y - CELL\_SIZE && cur\_position.y < CELL\_SIZE + pacman\_position.y)

           return 1;

   return 0;

}

## pacman.cpp

#include "pacman.h"

#include <array>

#include <cmath>

#include "SFML/Graphics.hpp"

#include "game\_types.h"

#include "tools.h"

pacman::pacman() : is\_animation\_over(0),

                  is\_dead(0),

                  direction(0),

                  energizer\_timer(0),

                  cur\_position({0, 0}),

                  score({0,0})

{

}

void pacman::draw(bool is\_victory, sf::RenderWindow &window)

{

   byte frame = (byte)(floor(animation\_timer / (float)(PACMAN\_ANIMATION\_SPEED)));

   sf::Sprite sprite;

   sf::Texture texture;

   sprite.setPosition(cur\_position.x, cur\_position.y);

   if (is\_dead || is\_victory)

   {

       if (animation\_timer < PACMAN\_DEATH\_FRAMES \* PACMAN\_ANIMATION\_SPEED)

       {

           animation\_timer++;

           texture.loadFromFile("img/pacman\_death" + std::to\_string(CELL\_SIZE) + ".png");

           sprite.setTexture(texture);

           sprite.setTextureRect(sf::IntRect(CELL\_SIZE \* frame, 0, CELL\_SIZE, CELL\_SIZE));

           window.draw(sprite);

       }

       else

       {

           is\_animation\_over = true;

       }

   }

   else

   {

       texture.loadFromFile("img/pacman" + std::to\_string(CELL\_SIZE) + ".png");

       sprite.setTexture(texture);

       sprite.setTextureRect(sf::IntRect(CELL\_SIZE \* frame, CELL\_SIZE \* direction, CELL\_SIZE, CELL\_SIZE));

       window.draw(sprite);

       animation\_timer = (1 + animation\_timer) % (PACMAN\_ANIMATION\_FRAMES \* PACMAN\_ANIMATION\_SPEED);

   }

}

void pacman::reset()

{

   is\_animation\_over = 0;

   is\_dead = 0;

   direction = 0;

   animation\_timer = 0;

   energizer\_timer = 0;

}

void pacman::update(byte level, game\_map &map)

{

   std::array<bool, 4> walls{};

   walls[0] = map\_collision(0, {PACMAN\_SPEED + cur\_position.x, cur\_position.y}, map);

   walls[1] = map\_collision(0, {cur\_position.x, cur\_position.y - PACMAN\_SPEED}, map);

   walls[2] = map\_collision(0, {cur\_position.x - PACMAN\_SPEED, cur\_position.y}, map);

   walls[3] = map\_collision(0, {cur\_position.x, PACMAN\_SPEED + cur\_position.y}, map);

   if (sf::Keyboard::isKeyPressed(sf::Keyboard::Right))

       if (!walls[directions::right])

           direction = directions::right;

   if (sf::Keyboard::isKeyPressed(sf::Keyboard::Up))

       if (!walls[directions::up])

           direction = directions::up;

   if (sf::Keyboard::isKeyPressed(sf::Keyboard::Left))

       if (!walls[directions::left])

           direction = directions::left;

   if (sf::Keyboard::isKeyPressed(sf::Keyboard::Down))

       if (!walls[directions::down])

           direction = directions::down;

   if (!walls[direction])

       switch (direction)

       {

       case directions::right:

           cur\_position.x += PACMAN\_SPEED;

           break;

       case directions::up:

           cur\_position.y -= PACMAN\_SPEED;

           break;

       case directions::left:

           cur\_position.x -= PACMAN\_SPEED;

           break;

       case directions::down:

           cur\_position.y += PACMAN\_SPEED;

       }

   if (-CELL\_SIZE >= cur\_position.x)

       cur\_position.x = CELL\_SIZE \* MAP\_WIDTH - PACMAN\_SPEED;

   else if (CELL\_SIZE \* MAP\_WIDTH <= cur\_position.x)

       cur\_position.x = PACMAN\_SPEED - CELL\_SIZE;

   if (collect\_pellets(cur\_position, map, score, level))

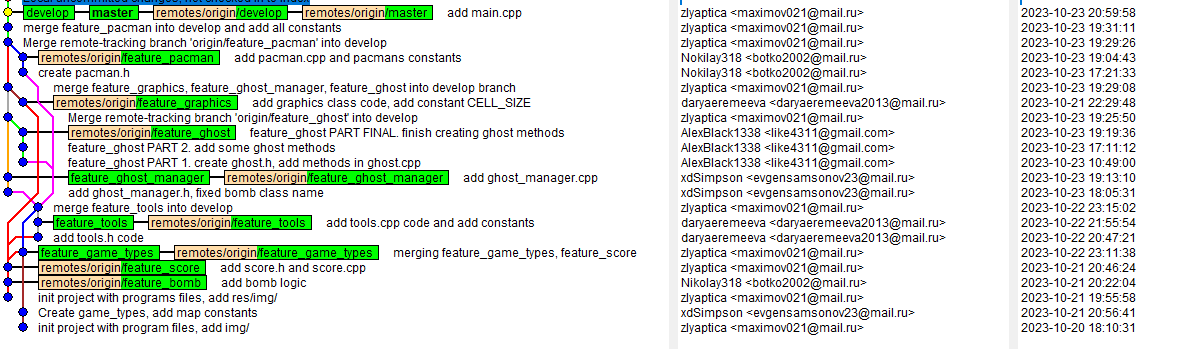
       energizer\_timer = (unsigned short)(ENERGIZER\_DURATION / pow(2, level));

   else

       energizer\_timer = std::max(0, energizer\_timer - 1);

}

# История проекта Github



**Мои коммиты:**

1. “add bomb logic”.

Добавлена логика класса бомба.

1. “create pacman.h”.

Создан код для файла pacman.h.

1. “add pacman.cpp and pacmans constants”.

Добавлен код для файла pacman.cpp и добавлены константы для пакмана.

# Вывод

В ходе выполнения командного задания в рамках курсового проекта использовалась среда разработки Qt язык C++ с использованием библиотеки SFML. Применялась система контроля версий git. Реализованы все прецеденты. В ходе тестирования игры зависания или сбои не были обнаружены. В алгоритмы методов старались не добавлять не нужных циклов, массивов и др. Были разработаны классы bomb и pacman, которые содержат в себе описание бомбы, пакмана и их поведение.