Министерство науки и высшего образования РФ

ФГБОУ ВПО «Пермский государственный национальный исследовательский университет»

Кафедра прикладной математики и информатики

**Индивидуальное задание №2**

***по дисциплине «Алгоритмизация и программирование»***

|  |  |
| --- | --- |
|  | **Выполнил:** |
|  | Студент 1-го курса  Института компьютерных наук и технологий  \_\_\_\_\_\_\_\_\_\_\_ Арланова А.А.,  группа ПМИ-1,2-2022 |
|  | **Проверил:** |
|  | Старший преподаватель кафедры прикладной математики и информатики ПГНИУ,  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Шеина Т. Ю. |
|  | «\_\_\_»\_\_\_\_\_\_\_\_\_ 2023 г. |

Пермь 2023

## **Оглавление**

[1 Постановка задачи 3](#_Toc131437059)

[2 Определение идеи алгоритма, выбор методов решения 4](#_Toc131437060)

[3 Тестирование программы 5](#_Toc131437061)

[4 Текст программы 6](#_Toc131437062)

# Постановка задачи

По заданию необходимо написать программу-игру «Война с инопланетянами», в которой с верхней части экрана будут падать мобы (инопланетяне), а внизу экрана будет находиться игрок, который может сбивать врагов выстрелами (с помощью мыши). Вести подсчет количества сбитых врагов. Игрок должен иметь 3 жизни, каждую из которых он теряет при столкновении с врагом. Добавить в игру звуковое сопровождение (фоновую музыку, звуки выстрелов и звуки взрывов при попадании). Общее количество мобов - не более 30. Падать они должны неравномерно и с разной скоростью.

Так же я хочу реализовать возможность игры вдвоем на одном устройстве, следовательно, и выбор управления, так как два игрока не должны иметь один и тот же тип управления. Добавить, чтобы существовало два типа врагов, за которых добавляло разное количество очков. А чтобы игроки бесконечно не стреляли во врагов, решила, что необходимо ограничить количество пуль, которые они могут использовать. Но чтобы не было так сложно, необходимо добавить различные бонусы, такие как, увеличение количество жизней, увеличение запаса пуль и на время бесконечные пули. Реализовать два игровых режима. Первый – сюжетный, где вначале проигрывается катсцена (текст с картинками), после запускается сама игра, и при наборе определенного очков, чтобы выходил босс инопланетян (враг с очень большим запасом здоровья и разными атаками). После победы над ним проигрывается завершающая катсцена. Второй режим – бесконечный, в нем будут отсутствовать какие-либо катсцены и босс, но зато есть возможность ставить рекорды, которые сохраняются в файле. Так же есть желание создать главное меню с кнопками, которые позволяют не только выбрать какой-либо режим игры, но и выключить/включить музыку, сменить полноэкранный режим на оконный. И последняя моя задача - ставить игру на паузу при нажатии клавиши.

# Определение идеи алгоритма, выбор методов решения

Для создания этой игры я использовала IDE PyCharm Community Edition и модуль Pygame.

В ходе написания данной программы, я пыталась придерживаться методологии ООП, где основные элементы программы должны представлять собой объекты, у которых есть свойства и методы. Рассмотрим классы, которые я сформировала:

1. **Player**

Описание: Player является объектом, с которым себя идентифицирует игрок. Пользователь способен при помощи клавиш или мышки его перемещать и создавать объекты – пули.

Свойства: изображение, размер, скорость, координаты, тип управления (мышка, стрелки ←→ + ↑ или клавиши ad + w), массив с объектами Bullet, бесконечные ли пули (логическая переменная), время до окончания бесконечных пуль, прямоугольная область (объект класса Rect, который в дальнейшем позволит отслеживать столкновение других объектов), количество жизней, пуль и очков

Методы:

1. Движение по оси x

Если управление через клавиши, то проверяем, какие клавиши нажаты. Если совпадает тип управления и необходимые клавиши для него, то прибавляем (или отнимаем) к координатам по x скорость игрока.

Если же управление через мышку, то просто приравниваем координаты x значению координат мышки по x

1. Стрельба

Если количество пуль в игре не превосходит максимальное количество пуль и при этом игрок жив (то есть у него жизней больше чем 0), тогда мы проверяем, какая клавиша была отжата, и если она совпала с типом управления, то добавляем в массив пуль новый объект Bullet, проигрывая звук выстрела.

1. Отрисовка

Метод, который отображает игрока на заданной поверхности.

1. **Bullet**

Описание: объект, цель которого перемещаться по оси y, и если встречает какой-то объект, то самоуничтожаться при этом давая очки игроку и уничатожая врага (или отнимая жизнь у игрока, если пуля, была выпущена боссом)

Свойства:

Методы:

1. Enemy

У объектов типа враг (Enemy) есть такие свойства: тип врага, размер, скорость по y, скорость по x, изображение, координаты и прямоугольная область. Методами же являются движение врага по осям x, y и отрисовка врага на экране.

1. FallingBuff

Данный класс резализует объект падающий бонус для игрока. У него есть тип бонуса, изображение, размеры, координаты, скорость и прямоугольная область. Падающий бонус может двигаться по оси y и так же отрисовываться на экране.

1. BossEvent

Благодаря этому классу мы создаем объект Босс. У которого есть количество жизней, равное ширине полоски здоровья.

1. Laser
2. Button
3. Star

# Тестирование программы

# ЛИСТИНг программы

Исходный файл программы *hangman.cpp*  прилагается к работе в электронном виде, ниже привожу его полностью. Также прилагается exe-файл с txt-файлом, хранящий слова для игры.

#include <iostream>

#include <fstream>

#include <windows.h>

#include <ctime>

#include <string>

using namespace std;

char getUserChar() //получение русской буквы

{

cout << "Введите букву: ";

while (true)

{

string UserString;

string ABC = "йцукенгшщзхъфывапролджэячсмитьбюёЙЦУКЕНГШЩЗХЪФЫВАПРОЛДЖЭЯЧСМИТЬБЮЁ";

cin >> UserString;

if (UserString.length() == 1 && ABC.find(UserString, 0) < 33 && ABC.find(UserString, 0) != -1)

return ABC[ABC.find(UserString, 0)+33]; //преобразование строчных букв в заглавные

else if (UserString.length() == 1 && ABC.find(UserString, 0) >= 33 && ABC.find(UserString, 0) != -1)

return UserString[0]; //возврат заглавных букв

else cout << "\nВведите пожалуйста русскую букву: ";

}

}

bool getGame() //получение только 1 или 0

{

while (true)

{

string Check;

cout << "1 - Да\n";

cout << "0 - Нет\n";

cin >> Check;

if (Check == "1")

return true;

else if (Check == "0")

return false;

else cout << "Некорректное значение\n";

}

}

//--------Отрисовка виселицы---------------

void DrawZero()

{

cout << " \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";

cout << " | |\n";

cout << " | |\n";

cout << " |\n";

cout << " |\n";

cout << " |\n";

cout << " |\n";

cout << " |\n";

cout << " |\n";

cout << " |\n";

cout << " |\n";

cout << " |\n";

cout << " |\n";

cout << " |\n";

cout << " |\n";

cout << " |\n";

cout << " |\n";

cout << " |\n";

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\n\n";

}

void DrawOne()

{

cout << " \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";

cout << " | |\n";

cout << " | |\n";

cout << " --- |\n";

cout << " / \\ |\n";

cout << " \\ / |\n";

cout << " --- |\n";

cout << " |\n";

cout << " |\n";

cout << " |\n";

cout << " |\n";

cout << " |\n";

cout << " |\n";

cout << " |\n";

cout << " |\n";

cout << " |\n";

cout << " |\n";

cout << " |\n";

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\n\n";

}

void DrawTwo()

{

cout << " \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";

cout << " | |\n";

cout << " | |\n";

cout << " --- |\n";

cout << " / \\ |\n";

cout << " \\ / |\n";

cout << " --- |\n";

cout << " | |\n";

cout << " | |\n";

cout << " | |\n";

cout << " | |\n";

cout << " | |\n";

cout << " |\n";

cout << " |\n";

cout << " |\n";

cout << " |\n";

cout << " |\n";

cout << " |\n";

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\n\n";

}

void DrawThree()

{

cout << " \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";

cout << " | |\n";

cout << " | |\n";

cout << " --- |\n";

cout << " / \\ |\n";

cout << " \\ / |\n";

cout << " --- |\n";

cout << " | |\n";

cout << " /| |\n";

cout << " / | |\n";

cout << " / | |\n";

cout << " / | |\n";

cout << " |\n";

cout << " |\n";

cout << " |\n";

cout << " |\n";

cout << " |\n";

cout << " |\n";

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\n\n";

}

void DrawFour()

{

cout << " \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";

cout << " | |\n";

cout << " | |\n";

cout << " --- |\n";

cout << " / \\ |\n";

cout << " \\ / |\n";

cout << " --- |\n";

cout << " | |\n";

cout << " /|\\ |\n";

cout << " / | \\ |\n";

cout << " / | \\ |\n";

cout << " / | \\ |\n";

cout << " |\n";

cout << " |\n";

cout << " |\n";

cout << " |\n";

cout << " |\n";

cout << " |\n";

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\n\n";

}

void DrawFive()

{

cout << " \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";

cout << " | |\n";

cout << " | |\n";

cout << " --- |\n";

cout << " / \\ |\n";

cout << " \\ / |\n";

cout << " --- |\n";

cout << " | |\n";

cout << " /|\\ |\n";

cout << " / | \\ |\n";

cout << " / | \\ |\n";

cout << " / | \\ |\n";

cout << " / |\n";

cout << " / |\n";

cout << " / |\n";

cout << " / |\n";

cout << "/ |\n";

cout << " |\n";

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\n\n";

}

void DrawSix()

{

cout << " \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";

cout << " | |\n";

cout << " | |\n";

cout << " --- |\n";

cout << " / \\ |\n";

cout << " \\ / |\n";

cout << " --- |\n";

cout << " | |\n";

cout << " /|\\ |\n";

cout << " / | \\ |\n";

cout << " / | \\ |\n";

cout << " / | \\ |\n";

cout << " / \\ |\n";

cout << " / \\ |\n";

cout << " / \\ |\n";

cout << " / \\ |\n";

cout << "/ \\ |\n";

cout << " |\n";

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\n\n";

}

void DrawGallows(string wrongChars)

{

switch (wrongChars.length())

{

case 0:

DrawZero();

break;

case 1:

DrawOne();

break;

case 2:

DrawTwo();

break;

case 3:

DrawThree();

break;

case 4:

DrawFour();

break;

case 5:

DrawFive();

break;

case 6:

DrawSix();

break;

default:

cout << "ERROR";

break;

}

}

//-----------------------------------------

bool GameResult(string keyWord,string UserWord, string &wrongChars)

{

char UserChar;

while (true)

{

//вывод интерфейса

cout << "Слово: ";

for (int i = 0; i < UserWord.length(); i++) cout << UserWord[i] << ' ';

cout << "\n\n";

cout << "Ошибки(" << wrongChars.length() << ") : " << wrongChars << "\n\n";

DrawGallows(wrongChars);

//------------------

UserChar = getUserChar();

bool Mistake = true;

for (int i = 0; i < keyWord.length(); i++)

{

if (keyWord[i] == UserChar)

{

UserWord[i] = UserChar; //заполнение строки пользоателя правильно отгаданными символами

Mistake = false;

}

}

if (Mistake && wrongChars.find(UserChar, 0) == -1) wrongChars = wrongChars + UserChar; //дополнение ошибочно отгаданного символа к строке

system("cls");

if (keyWord == UserWord) return true; //выигрыш

if (wrongChars.length() == 6) return false; //проигрыш

}

}

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

srand(time(0));

string keyWord = "", UserWord = "", wrongChars = "";

char UserChar;

bool Game = false, victory = false;

ifstream f;

int keyNumber;

string\* words = new string[256];

f.open("words.txt");

int j = 0;

while (getline(f, words[j])) ++j; //заполнение массива словами из файла

f.close();

cout << "-----------------ВИСЕЛИЦА-----------------\n";

cout << "Правила игры:\n";

cout << "1. Компьютер «загадывает» слово.\n";

cout << "2. На экран выводится столько символов «подчеркивания», сколько необходимо для написания слова.\n";

cout << "3. Затем на экран выводится изображение виселицы.\n";

cout << "4. Вы предлагаете букву, которая может входить в это слово.\n";

cout << "5. Если вы угадываете -> компьютер пишет ее в нужных незаполненных местах.\n";

cout << "Иначе компьютер пишет эту букву в стороне и начинает дорисовывать виселицу, добавляя к петле голову\n";

cout << "6. Вы продолжаете отгадывать буквы до тех пор, пока не отгадает все слово.\n";

cout << "За каждый неправильный ответ компьютер добавляет одну часть повешенного к виселице(голова, тело, две руки, две ноги).\n";

cout << "7. Если повешенный нарисован раньше, чем вы смогли угадать слово -> компьютер побеждает.\n";

cout << "Иначе побеждаете вы :)\n";

cout << "Хорошей игры!\n";

cout << "------------------------------------------\n\n";

cout << "Хотите сыграть?\n";

Game = getGame();

system("cls");

while (Game)

{

keyNumber = rand() % j; //генерация номера рандомного слова

keyWord = words[keyNumber]; //загаданное слово

UserWord = "";

for (int i = 0; i < keyWord.length(); i++) UserWord = UserWord + "\_"; //заполнение пользовательского слова символами нижнего подчеркивания

wrongChars = "";

victory = GameResult(keyWord, UserWord, wrongChars); //победа или поражение?

if (victory) cout << "Поздравляем! Вы выиграли! :)\n";

else cout << "Сожалеем, но вы проиграли :(\n";

//вывод интерфейса

cout << "Слово: " << keyWord << "\n";

cout << "Ошибки(" << wrongChars.length() << ") : " << wrongChars << "\n\n";

DrawGallows(wrongChars);

//-----------------

cout << "Хотите еще раз сыграть?\n";

Game = getGame();

system("cls");

}

return 0;

}