ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ И

ИНФОРМАТИКИ»

Кафедра вычислительных систем

КУРСОВАЯ РАБОТА

по дисциплине «Технологии разработки программного обеспечения»

на тему "Игра «Виселица»"

Выполнил:  
 Зимин Денис

Проверил:  
доц., к.ф.-м.н. Пудов С. Г.

Новосибирск, 2017

Содержание

[ВВЕДЕНИЕ И ПОСТАНОВКА ЗАДАЧИ 3](#_Toc482440476)

[ТЕХНИЧЕСКОЕ ЗАДАНИЕ 4](#_Toc482440477)

[ОПИСАНИЕ ВЫПОЛНЕННОГО ПРОЕКТА 5](#_Toc482440478)

[ЛИЧНЫЙ ВКЛАД В ПРОЕКТ](#_Toc482440479) 8

[ПРИЛОЖЕНИЕ. ТЕКСТ ПРОГРАММЫ](#_Toc482440480) 10

**Введение и постановка задачи**

Задача курсовой работы заключалась в командной поэтапной реализации игры “Виселица” с использованием систем контроля версий.

Правила игры:

Компьютером загадывается слово, которое игрок должен угадать, используя буквы алфавита и возможность совершить ограниченное количество ошибок. Если буква правильная , то показывается позиция ее в слове, а если не правильная, то рисуется сама виселица, а после человек в ней начиная с головы. Всего 9 попыток и 3 подсказки . При проигрыше вам предлагается сыграть еще раз , либо закончить игру и тем самым посмотреть свой лучший результат.

Для правильной работы программы требовалось реализовать:

* + - Проверку корректности входных данных
    - Функцию генерации случайного числа компьютером
    - Функцию выведения состояния в виде картинки , на данный момент
    - Функция подсчета шага на котором мы находимся.
    - Функция , которая проверяет остались ли подсказки.

3

# ТЕХНИЧЕСКОЕ ЗАДАНИЕ

# Техническое задание

# "Виселица"

Приложение «Hangman» - однопользовательское приложение, созданное на основе известной игры на отгадывание «Виселица». Язык программирования, используемый для написания приложения: С (Си++).

1. Функционал проекта: В игру можно играть только в оффлайн режиме ( без интернет соединения ). Игра будет предоставлять ваши лучшие результаты, которые можно улучшить. Если игрок не в состоянии угадать слово, ему предлагают воспользоваться тремя подсказками . Слово обязательно должно быть именем существительным, единственного числа в именительном падеже. Игрок должен угадать слово , за каждую неверно угаданную букву рисуется сама виселица , а после части тела человека к виселице, начиная с головы. Игра продолжается до тех пор , пока не нарисуются ноги человека или игрок не угадает все буквы загаданного слова.
2. Формат входных данных : символ (переменная типа char ), который соответствует одному из кода в таблицу ACII от С0 до FF ( 32 прописных и строчных русских букв ). Осуществляется сравнение введенного символа с каждым символом начального массива.
3. Интерфейс приложения: Приложения интерактивное.
4. Работа с извне файлами : Будет создан words.txt, в котором будут храниться слова, tips.txt будут хранится подсказки и record.txt, где буду хранится рекорды .
5. Приложение не будет принимать аргументы командной строки.

4

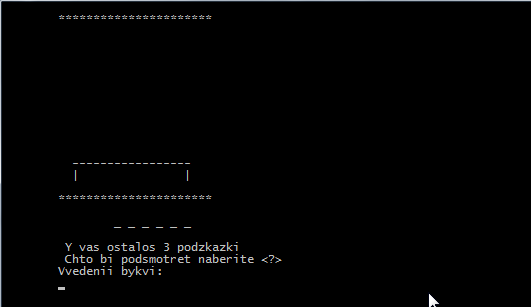
# ОПИСАНИЕ ВЫПОЛНЕННОГО ПРОЕКТА

Общекомандная часть, с примерами работы ПО

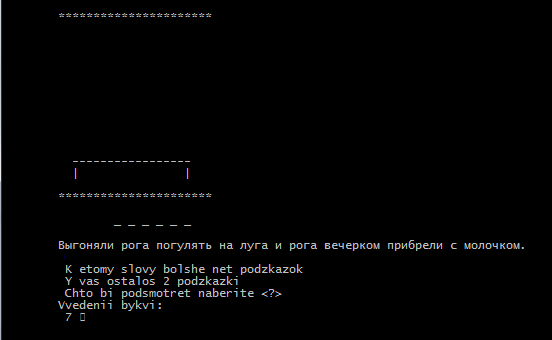
Выполненный проект соответствует ТЗ. В общекомандную часть проекта вошли совместные усилия по устранению возникших во время разработки проблем (ошибок, багов и т.д.),

**Примеры работы ПО:**

При запуске приложения нас встречает сама картинка виселицы, а ниже ее нижние подчеркивания, сколько подчеркиваний , столько и букв. Показывает сколько осталось у вас подсказок и , что бы воспользоваться подсказкой нам приложение подсказывает, что нужно набрать “7”.

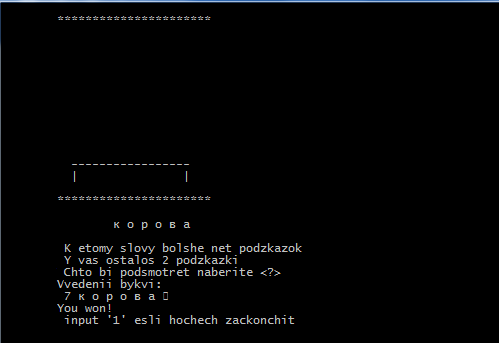


Воспользуемся подсказкой….



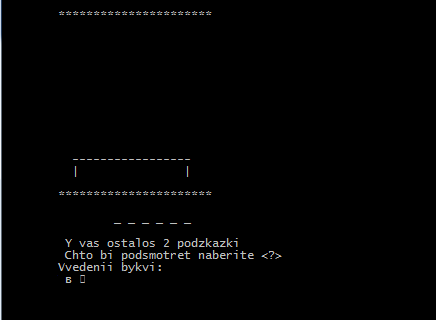
После использования, мы видим результат работы , количество подсказок уменьшилось, а так же показались введенные буквы( не повторяет предыдущие символы ).

5



Введенные символы совпали со словом и нам приложение написало “You won”, а так же предлагает закончить игру ( приложение закроется ).

Далее вводим любой символ и игра продолжается, теперь попробуем проиграть.



Постепенно рисуется виселица, а далее и тело человека начиная с головы.

При проигрыше нам напишет наш рекорд за эту игру ( мы отгадали одно слово, поэтому 1 ) и наш лучший рекорд за все время.



6

При нажатии “1” игра завершается и показывается сколько рекордов мы побили , и наш лучший рекорд. После нажатия любой клавиши приложение закроется.



7

# ЛИЧНЫЙ ВКЛАД В ПРОЕКТ

Работа над постановкой технического задания

Изучение и постановка правил игры

Составление загадок к заданным словам в текстовом документе

Разработка функции Tips

Разработка точки входа программы в документ с загадками

ifstream myfile("tips.txt");

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

getline(myfile, letters[i].podz);

}

Покрытие тестами функций :

- int Tips(char inputChar, int quantityPodzOnLetters, int podzkazki, int flag)

- int checkErros(unsigned int errors, char \*slovo, int step)

- int correctLetters(char \*slovo, char inputChar, char outchar[], unsigned int \*pravilno)

- int errorsInLetters(char \*slovo, char inputChar)

# ПРИЛОЖЕНИЕ. ТЕКСТ ПРОГРАММЫ

|  |  |
| --- | --- |
| #include <stdio.h> | |
|  | #include <stdio.h> |
|  | #include <stdlib.h> |
|  | #include <time.h> |
|  | #include <Windows.h> |
|  | #include <conio.h> |
|  | #include <ctype.h> |
|  | #include <iostream> |
|  | #include <cstdio> |
|  | #include <fstream> |
|  | #include <string> |
|  | #include "vis.h" |
|  | #define SIZE 10 |
|  |  |
|  | using namespace std; |
|  |  |
|  |  |
|  | struct slova{ |
|  | char key[9][20]; |
|  | string podz; |
|  | }; |
|  |  |
|  | int main() |
|  | { |
|  | setlocale(LC\_CTYPE, "Russian"); |
|  | SetConsoleCP(1251); |
|  | SetConsoleOutputCP(1251); |
|  |  |
|  | time\_t currentTime; |
|  | time(&currentTime); |
|  |  |
|  | srand(currentTime); |
|  |  |
|  | const int maxLetters = 20; |
|  | const int numberStep = 10; |
|  | int record = 0; |
|  | int lastrecord = 0;  9 |
|  |  |
|  |  |
|  | //char letters[] = "???"; |
|  | slova letters[SIZE]; |
|  |  |
|  | FILE \*WORDS = fopen("words.txt", "r");//opening file with the words |
|  |  |
|  | if (WORDS) |
|  | for (int i = 0; i < SIZE; i++) { |
|  | fscanf(WORDS, "%s", letters[i].key); |
|  | } |
|  | else { |
|  | printf("Error: file doesn't exists!\n"); |
|  | return -1; |
|  | } |
|  |  |
|  | ifstream myfile("tips.txt"); |
|  | for (int i = 0; i < SIZE; i++){ |
|  | getline(myfile, letters[i].podz); |
|  | } |
|  |  |
|  |  |
|  | char inputChar = 0; |
|  |  |
|  |  |
|  |  |
|  | char \*slovo; |
|  | string podz; |
|  |  |
|  | int random = 0; // Random number from 0 to size |
|  |  |
|  | int podzkazki = 3; |
|  | do |
|  | { |
|  | random = randomGenerator(random); // Random number for structure |
|  | slovo = letters[random].key[0]; |
|  | podz = letters[random].podz;  10 |
|  |  |
|  | int sizeLetters = strlen(slovo); // How many letters in a word |
|  | char outchar[maxLetters]; // The data to be displayed on the screen |
|  | char inputletters[maxLetters]; // For input characters |
|  | memset(inputletters, 0, maxLetters); // Clean the array |
|  |  |
|  | for (int i = 0; i < sizeLetters; i++) // We put all the letters not guessing |
|  | { |
|  | outchar[i] = '\_'; |
|  | } |
|  |  |
|  |  |
|  | //Game settings |
|  | unsigned int pravilno = 0; |
|  | unsigned int step = 0; |
|  | unsigned int errors = 1; |
|  | //int number = 0; |
|  | int counter = 0; |
|  | int quantityPodzOnLetters = 1; |
|  |  |
|  |  |
|  | do |
|  | { |
|  | int flag = 0; |
|  |  |
|  | errors = errorsInLetters(slovo, inputChar); |
|  | pravilno = 0; |
|  | step = checkErros(errors, slovo, step); |
|  |  |
|  | system("cls"); |
|  | ris(step); |
|  | printf("\n\n\t\t"); |
|  |  |
|  | correctLetters(slovo, inputChar, outchar, &pravilno); |
|  |  |
|  | printLetters(slovo, outchar);  11 |
|  |  |
|  |  |
|  | flag = Tips(inputChar, quantityPodzOnLetters, podzkazki, flag); |
|  | if (flag == 1) |
|  | { |
|  | printf("\n\t"); |
|  | cout << podz << endl; |
|  | quantityPodzOnLetters--; |
|  | podzkazki--; |
|  | } |
|  | if (quantityPodzOnLetters == 0 && podzkazki > 0) |
|  | { |
|  | printf("\n\t K etomy slovy bolshe net podzkazok"); |
|  | } |
|  |  |
|  |  |
|  | printf("\n\t Y vas ostalos %d podzkazki", podzkazki); |
|  | printf("\n\t Chto bi podsmotret naberite <?>"); |
|  | printf("\n\t"); |
|  | printf("Vvedenii bykvi:"); |
|  |  |
|  |  |
|  | inputletters[counter] = inputChar; |
|  | if (inputletters[counter] != inputletters[counter - 1]) |
|  | { |
|  | counter++; // Check for similar symbols |
|  | } |
|  | printf("\n\t"); |
|  | printInputLetters(inputletters);// Display the entered letters |
|  |  |
|  |  |
|  | inputChar = \_getch(); |
|  |  |
|  |  |
|  | } while ((step < numberStep - 1) && (pravilno < strlen(slovo))); |
|  | FILE \*RECORDS = fopen("record.txt", "r+");//opening file with the words |
|  | 12 |
|  | if (!RECORDS){ |
|  | printf("Error: file doesn't exists!\n"); |
|  | return -1; |
|  | } |
|  | else{ |
|  | fscanf(RECORDS, "%d", &lastrecord); |
|  |  |
|  | if (record > lastrecord){ |
|  | freopen("record.txt", "w", RECORDS); |
|  | fprintf(RECORDS, "%d letters is Yours best record ", record); |
|  | fclose(RECORDS); |
|  |  |
|  | } |
|  | } |
|  |  |
|  |  |
|  | if (pravilno == strlen(slovo)) |
|  | { |
|  | printf("\n\tYou won!\n"); |
|  | record++; |
|  | } |
|  | else |
|  | { |
|  | printf("\nYou lose new record = %d last best record = %d\n", record, lastrecord); |
|  | record = 0; |
|  | } |
|  |  |
|  |  |
|  | printf("input 1"); |
|  | inputChar = \_getch(); |
|  | } while (inputChar != '1'); |
|  | printf("\nYour new record: %d \nYour last best record: %d\n", record, lastrecord); |
|  | return 0; |
|  | }  13 |