Министерство образования и науки РФ

Федеральное государственное автономное образовательное учреждение высшего образования

«Омский государственный технический университет»

|  |  |
| --- | --- |
| Факультет (институт) | *Информационных технологий и компьютерных систем* |
|  |  |
| Кафедра | *Прикладная математика и фундаментальная информатика* |
|  |  |

**Расчетно-графическая работа**

|  |  |
| --- | --- |
| по дисциплине | ***Алгоритмизация и программирование*** |
|  |  |
| на тему | Разработка программы |

Пояснительная записка

|  |  |
| --- | --- |
| **Шифр проекта** | 020-РГР-02.03.02-№ 9-ПЗ |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | **Студента** | | Казаков Илья Михайлович | | | | | |
|  |  |  |  | | фамилия, имя, отчество полностью | | | | | |
|  |  |  | Курс | *1* |  | Группа | | ФИТ-**221** | | |
|  |  |  |  |  |  | |  |  | |  |
|  | | | **Направление (специальность)** | | | | | ***02.03.02*** | | |
|  | | | *Фундаментальная информатика и информационные технологии* | | | | | | | |
|  |  |  | код, наименование | | | | | | | |
|  |  |  | Руководитель | | ***ст. преподаватель*** | | | | | |
|  |  |  | ученая степень, звание | | | | | |
|  |  |  | ***Федотова И.В.*** | | | | | | | |
|  |  |  | фамилия, инициалы | | | | | | | |
|  |  |  | Выполнил | |  | | | | | |
|  |  |  | дата, подпись студента | | | | | |
|  |  |  | **Работа защищена с количеством баллов** | | | | | | | |
|  |  |  |  | | | | | |  | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | дата, подпись руководителя |  |  |  |

Омск 2023

**СОДЕРЖАНИЕ**

Введение 3

Теория 4

Разработка программы 5

Результаты 7

Заключение 8

Список литературы 9

**Введение**

Сегодня в мире тысячи программ и приложений, имеющих разное назначение. Главная задача любого ПО – решение потребности конкретной целевой группы. Однако разработка софта – сложная работа. Для ее выполнения требуется квалификация, опыт, знания со стороны разработчиков. При этом речь не только в сложности кода конкретного языка программирования. Любое веб-приложение или прикладная программа должна быть удобной, практичной, функциональной и интуитивно понятной.

Любая программа должна иметь преимущества, быть реализованной в короткие, оговоренные сроки. Это важно, чтобы конкуренты не смогли перехватить инициативу. Ведь если процесс разработки и тестирования затягивается, момент благоприятного выхода на рынок может быть упущен. Особый момент – функционал. Ведь ПО должно решать проблему целевой аудитории. В таком случае софт будет востребован, станет популярным среди активных пользователей.

**Теория**

Быки и коровы — логическая игра, в ходе которой за несколько попыток один из игроков должен определить, что задумал другой игрок. Варианты игры могут зависеть от типа отгадываемой последовательности — это могут быть числа, цвета, пиктограммы или слова. После каждой попытки задумавший игрок выставляет «оценку», указывая количество угаданного без совпадения с их позициями (количество «коров») и полных совпадений (количество «быков»). Роли участников игры не равнозначны — угадывающий должен анализировать сделанные попытки и полученные оценки, то есть его роль активна. Его партнёр лишь сравнивает очередной вариант с задуманным и выставляет оценку по формальным правилам, то есть его роль пассивна. Для уравновешивания ролей одновременно играют две встречные партии.

Первоначально игра была задумана для двух игроков, но с появлением компьютерных версий стал популярен вариант, когда игрок отгадывает число, задуманное программой, то есть играет в одиночку.

**Разработка программы**

#include <iostream>

#include <string>

#include <ctime>

int NUMBER\_OF\_LETTERS = 4;

bool AskToPlayAgain();

void Intro();

void game(*std*::*string* word);

*std*::*string* Random();

int NumberOfAttempts = NUMBER\_OF\_LETTERS\*2 ;

int main()

{

*srand*(*time*(*NULL*));

Intro();

do

{

game(Random());

} while (AskToPlayAgain());

return 0;

}

void game(*std*::*string* word)

{

for (char i = 0; i < NumberOfAttempts; i++)

{

int Bull = 0;

int Cow = 0;

*std*::*cout* << "Enter your guess:";

*std*::*string* words = "";

*std*::*getline*(*std*::*cin*, words);

for (int a = 0; a < NUMBER\_OF\_LETTERS; a++)

{

if (word[a] == words[a])

{

Bull++; Cow--;

}

for (int b = 0; b < NUMBER\_OF\_LETTERS; b++)

{

if (word[a] == words[b])

{

Cow++;

}

}

}

*std*::*cout* << "Bull = " << Bull << *std*::*endl*;

*std*::*cout* << "Cow = " << Cow << *std*::*endl*;

if (Bull == NUMBER\_OF\_LETTERS)

{

*std*::*cout* << "YOU WIN\n";

return;

}

}

*std*::*cout* << "You failed to guess my word\n";

}

void Intro()

{

*std*::*cout* << "Hi, go to game \"Bull & Cow\"" << *std*::*endl*;

*std*::*cout* << "How can i contact you?\t";

*std*::*string* yourname = "";

*std*::*getline*(*std*::*cin*, yourname);

*std*::*cout* << yourname << " go to play!\n";

*std*::*cout* << "I made a five letter word\n";

}

bool AskToPlayAgain()

{

*std*::*cout* << "Wanna try again?(Yes or No) ";

*std*::*string* response = "";

*std*::*getline*(*std*::*cin*, response);

*std*::*cout* << *std*::*endl*;

return (response[0] == 'y') || (response[0] == 'Y');

};

*std*::*string* Random()

{

int a = *rand*() % 10000;

*std*::*string* word = *std*::*to\_string*(a);

for (int i = 0; i < 4; i++)

{

for (int j = 0; j < 4; j++)

{

if (i == j) { continue; }

if (word[i] == word[j]) { return Random(); }

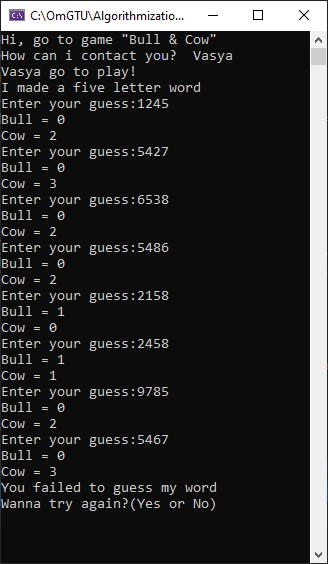
}

}

return word;

}

**Результаты**



**Рис. 1**

**Заключение**

В данной РГР была разработана игра “Быки и Коровы”. Целью данной работы было создание игры с которая будет хотя бы работать. Были изучены основные принципы работы с алгоритмами, реализованными для работы игры, а также основы работы с текстом.

Было проведено тестирование игры, в ходе которого были выявлены некоторые проблемы, такие как отсутствие проверки вводимых данных пользователем на корректность.

**Список литературы**

<https://learn.microsoft.com/en-us/cpp/>

<https://www.youtube.com/@SimpleCodeIT>