

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 «100 спичек»»

Выполнил:

ст. гр. ЗП-202

Черникова Е.А.

Проверил:

Содержание

Введение и постановка задачи:	3
Техническое задание:	3
Функциональность проекта:	3
Формат входных данных:	3
Интерфейс:	3
Описание выполненного проекта:	4
Unit-тесты:	5
Приложение. Текст программы:	6

Введение и постановка задачи:

Краткое изложение цели работы и задание на курсовую работу.

Цель работы: создать законченный и полностью рабочий программный продукт игру «100 спичек». Приложение должно иметь интуитивно понятный графический интерфейс, должно быть покрыто тестами, а также настроить непрерывную интеграцию (CI) для отслеживания состояния текущей версии программы.

Из кучки, первоначально содержащей 100 спичек игрок должен выбрать от 1 до 10 спичек.

Следующий ход делает бот, выбирая рандомно число от 1 до 10.

Целью игры является забрать со стола последнюю спичку.

Для написания приложения мы использовали язык программирования C++. Для тестирования функциональной части Unit-тесты были выполнены в коде программы.

Техническое задание:

Из кучки, первоначально содержащей 100 спичек, двое играющих поочередно берут по несколько спичек: не менее одной и не более десяти. Выигрывает взявший последнюю спичку. Нельзя выбрать меньше 1 и больше 10.

Функциональность проекта:

- 2.1. Стартовое меню, содержащие пункты: "Правила", "Играть", "Выход".
- 2.2. При выборе пункта "Правила" на экране выводятся правила игры.
- 2.3. При выборе пункта "Играть" начинается игра.
- 2.4. При выборе пункта "Выход" программа закрывается.

Формат входных данных:

- 3.1. Выборы пунктов в меню осуществляются цифрами "1", "2" и "3" соответственно.
- 3.2. При взятии спичек вводится целое число с клавиатуры от "1" до "10".

Интерфейс:

- 4.1. Работа пользователя с программой происходит в консоли.
- 4.2. На экран выводится: оставшееся количество спичек, количество взятых спичек игроком и количество спичек, которое игрок взял в последний раз.

Описание выполненного проекта:

При запуске программы открывается окно с интуитивно понятным интерфейсом (Рисунок 1.).

```
Выберете пункт меню:  
1. Правила  
2. Играть  
3. Выход  
>
```

Рисунок 1. «Меню»

При выборе пункта 1: «Правила» на консоль выводятся правила игры (Рисунок 2.).

```
>1  
Из кучки, первоначально содержащей 100 спичек, двое играющих поочередно  
берут по несколько спичек: не менее одной и не более десяти.  
Выигрывает взявший последнюю спичку.  
Выберете пункт меню:  
1. Правила  
2. Играть  
3. Выход  
>
```

Рисунок 2. «Правила игры»

При выборе пункта 2: «Играть» запускается игра (Рисунок 3.).

```
>2  
Ваш ход. На столе 100 спичек.  
Сколько спичек Вы берете?
```

Рисунок 3. «Начало игры»

При выборе числа от 1 до 10 игра продолжается, бот делает свой ход и игроку снова нужно выбрать количество спичек (Рисунок 4.).

```
Мой ход. Я взял 1 спичек  
Ваш ход. На столе 94 спичек.  
Сколько спичек Вы берете?
```

Рисунок 4. «Ход игры»

При выборе числа 0 или >10 появляется сообщение о неверном вводе. Игроку нужно повторить ввод корректно (Рисунок 5.).

```
15  
Неверно! Повторите ввод!  
Ваш ход. На столе 94 спичек.  
Сколько спичек Вы берете?
```

Рисунок 5. «Сообщение о неверном вводе данных»

Если последние спички забирает игрок, на экран выводится информация о победе и меню для выбора дальнейших действий (Рисунок 6.).

```

Мой ход. Я взял 9 спичек
Ваш ход. На столе 3 спичек.
Сколько спичек Вы берете?
3
Вы победили!
Выберете пункт меню:
1. Правила
2. Играть
3. Выход
>

```

Рисунок 6. «Информация о победе»

Если последние спички забирает бот, на экран выводится информация о поражении и меню для выбора дальнейших действий (Рисунок 7.).

```

Мой ход. Я взял 9 спичек
Ваш ход. На столе 16 спичек.
Сколько спичек Вы берете?
10
Мой ход. Я взял 6 спичек
Вы проиграли!
Выберете пункт меню:
1. Правила
2. Играть
3. Выход
>

```

Рисунок 7. «Информация о поражении»

По завершению игры можно сыграть повторно. При выборе пункта 3: «Выход» программа завершает работу.

Unit-тесты:

Unit-тесты были выполнены через тестовый фреймворк Catch.

```

#include "../function.h"
#define CATCH_CONFIG_MAIN
#include "../catch/catch.hpp"

TEST_CASE("Test Allowable Amount")
{
    bool expected = true;
    REQUIRE(check_number_match(2, 100) == expected);
}

TEST_CASE("Test Max Allowable Amount")
{
    bool expected = true;
    REQUIRE(check_number_match(10, 100) == expected);
}

TEST_CASE("Test Invalid Amount")
{
    bool expected = false;

```

```

    REQUIRE(check_number_match(15, 100) == expected);
}

```

Первые два теста проверяют доступное количество спичек, которые можно взять за 1 ход.

В ходе третьего теста входными параметрами были заведомо ложные данные.

Приложение. Текст программы:

```

#include "function.h"

using namespace std; //функция main прописана отдельным .cpp файлом

const int InitialCount = 100;

int main() {
    setlocale(LC_ALL, "Russian");
    srand(time(0));
    system("cls");

    int variant;

    do {
        print_menu();
        cin >> variant;

        switch (variant) {
            case 1:
                prav();
                break;

            case 2:
                game();
                break;

        }

    } while (variant != 3);

    return 0;
}

#include "function.h"
#include <sstream>
#include <stdio.h>
#include <stdlib.h>

void game()
{
    int Count, Num, Player;
    bool Correct;
    const int InitialCount = 100;
    Player = 2;
    Count = InitialCount;

    do {
        if (Player == 1)
            Player = 2;

```

```

else
    Player = 1;
if (Player == 1) {
    do {
        std::cout << "Ваш ход. На столе " << Count << " спичек.\n";
        std::cout << "Сколько спичек Вы берете?\n";
        std::cin >> Num;
        Correct = check_number_match(Num, Count);
    } while (!Correct);
}
else
{
    do {
        Num = rand() % 10 + 1;
        if (Num > Count)
            Num = Count;
        std::cout << "Мой ход. Я взял " << Num << " спичек\n";
    } while (!Correct);
}
Count -= Num;

} while (Count > 0);

if (Player == 1)
    std::cout << "Вы победили!\n";
else
    std::cout << "Вы проиграли!\n";
}

void print_menu() {
    cout << "Выберете пункт меню: " << endl;
    cout << "1. Правила" << endl;
    cout << "2. Играть" << endl;
    cout << "3. Выход" << endl;
    cout << ">";
}

void prav() {
    cout << " Из кучки, первоначально содержащей 100 спичек, двое играющих поочередно\n берут по несколько спичек: не менее одной и не более десяти.\n Выигрывает взявший последнюю спичку." << endl;
}

bool check_number_match(int n_pl_match, int quant)
{
    bool check = false;
    if (n_pl_match >= 1 && n_pl_match <= 10 && n_pl_match <= quant)
        check = true;
    else
    {
        std::cout << "Неверно! Повторите ввод!\n";
        check = false;
    }
    return check;
}

```