

Projekt z przedmiotu

## **Programowanie I**

Gra w c++

### **Kości**

Opracował:

Robert Kwoll

Grupa 1A

## Spis treści:

1. Charakterystyka gry
2. Instrukcja obsługi
3. Funkcjonalność
4. Specyfikacja techniczna i szczegóły

## 1.Charakterystyka gry

Kości to gra, która polega na rzucaniu 5 kośćmi (bądź mniej) i wyrzuceniu odpowiedniej ilości oczek. Następnie należy wybrać kategorię, do której chcemy wpisać dany wynik. Wygrywa ta osoba, która uzyska najwięcej punktów.

Bardziej szczegółowa instrukcja znajduje, się w grze po wpisaniu odpowiedniej literki w konsoli.

```
| Gra w kosci |  
|-----|  
| A.Gra      |  
| B.Instrukcja |  
| C.Wyniki   |
```

```
| A.Zatrzymanie wybranej kosci |  
| B.Zapis wyniku do tabeli     |  
| C.Wyswietlanie tabeli       |  
| D.Wroc do rzucania koscmi   |  
| E.Instrukcja                 |
```

## 2.Instrukcja obsługi

Program został stworzony w środowisku C++. Należy uruchomić grę poprzez Visual Studio 2015 lub nowszy, albo inny program obsługujący język c++.

## 3.Funkcjonalność

- A) wykorzystanie podstawowych zasad języka c++
- B) zdefiniowanie prostej struktury i operacje na niej
- C) podział programu na funkcje
- D)zapis i odczyt na plikach
- E) wykorzystanie wskaźników
- F) Losowanie liczb
- G) Połączenie pożytecznego z przyjemnością (gry i programowanie)

## 4.Specyfikacja techniczna i szczegóły

Projekt składa się z pliku głównego *main.cpp* i dokumentu *instrukcja.txt*, natomiast gdy skończymy grę chociaż raz, zostanie nasz wynik zapisany i utworzy się nowy plik o nazwie *wynik.txt*.

Plik główny składa się z następujących elementów.

Struktury o nazwie *tabela* oraz dodanych do tej struktury pól z nazwami poszczególnych kategorii.

```
struct tabela
{
    string nazwa;
    int punkty;
};

//Dodawanie elementow do struktury
tabela jeden{"   jedyнки   ",0 };
tabela dwa{ "   dwójki   ",0 };
tabela trzy{ "   trojki   ",0 };
tabela cztery{ "   czwórki   ", 0 };
tabela piec{ "   piatki   ",0 };
tabela szesc{ "   szóstki   ",0 };

tabela jed3{ "   3 jednakowe",0 };
tabela jed4{ "   4 jednakowe",0 };
tabela full{ "   full       ",0 };
tabela mstrit{ "   maly strit ",0 };
tabela dstrit{ "   duzy strit ",0 };
tabela general{ "   general   ",0 };
tabela szansa{ "   szansa     ",0 };
```

Zmiennych globalnych

```
int rzut = 1, rz[5]{}, n = 5; //
int k = 0, tablica[5]{}, a = 0;
int wynik=0, tab[13]{};
```

## Funkcji

```
void menu();
void rzuty();
void instrukcja();
void wyniki();
void sprawdz(char);
void sprawdz2(char);
void zapytanie();
void zatrzym(int* kosc);
void zapis(int* reszta);
void wyswietl();
```

Funkcja *menu()* rysuje nam początkowe menu z nazwą gry oraz 3 opcjami do wyboru

```
-----
|  Gra w kosci  |
|-----|
|  A.Gra        |
|  B.Instrukcja |
|  C.Wyniki     |
|-----|
```

Funkcja *rzuty()* najpierw wyświetla komunikat, który obecnie mamy rzut a następnie losuje nam z pośród 5 kości(bądź mniej), liczbę oczek od 1 do 6 na poszczególnych kościach.

```
int los = 0;
cout << rzut << ". Rzut " << endl;
cout << "Wyrzucone kosci to: ";
for (int i = 1; i <= n; i++)
{
    do
    {
        rz[los] = rand() % 7; //
    } while (rz[los] == 0);

    cout << rz[los] << " ";
    los++;
}
```

Funkcja *instrukcja()* wyświetla instrukcję z pliku *instrukcja.txt*.

```
string tekst{};

ifstream plik("instrukcja.txt");
if (plik.is_open())
{
    while (!plik.eof())
    {
        getline(plik, tekst);
        cout << tekst << endl;
    }
}
```

Funkcja *wyniki()* zapisuje wynik z całej gry do pliku *wynik.txt*

```
string nick;
cout << "Podaj swój nick" << endl;
cin >> nick;

ofstream plik;
plik.open("wynik.txt");
plik << nick << " " << wynik;

plik.close();

cout << "Twój wynik został zapisany :)" << endl;
```

Funkcja *zapytanie()* działa dokładnie tak samo jak funkcja *menu()*

```
| A.Zatrzymanie wybranej kosci |
| B.Zapis wyniku do tabeli     |
| C.Wyswietlanie tabeli       |
| D.Wroc do rzucania kosmi    |
| E.Instrukcja                 |
```

Funkcja *zatrzym()* za pomocą wskaźnika pobiera kości z liczbą oczek i na podstawie liczby, którą wprowadzi użytkownik sprawdza, czy rzeczywiście została taka kość wylosowana i zapisuje jej liczbę oraz zmniejsza ilość kości w następnym rzucie.

```

for (int i = 0; i < 5; i++)
{
    if (i > 0 && *(kosc + i) == oczko)
    {
        *(kosc + i) = 0;    //usuwa zapisana kosc z wskaznika
        tablica[a] = oczko;
        n--; //zmniejsza ilosc kosci do rzutu
        a++; //zapisuje kosc
        break;
    }

    if (i == 0 && *kosc == oczko)
    {
        *kosc = 0;
        tablica[a] = oczko;
        n--;
        a++;
        break;
    }
}

```

Funkcja *wyswietl()* wyświetla tabele z kategoriami oraz liczbą punktów(jeśli wyświetla przy danej kategorii 0 to albo wynik został już tam wpisany, bądź też nie).

Funkcja *zapis()* odpowiada za następujące czynności.

Wyświetla kości, które są do wykorzystania przy zapisie.

```

cout << "Twoje kosc to: ";
for (int i = 0; i < 5; i++)
{
    if (tablica[i] == 0)
    {
        while (*reszta == 0)
        {
            reszta++;
        }
        tablica[i] = *reszta;
        reszta++;
    }
    cout << tablica[i] << " ";
    suma += tablica[i];
}

```

Pyta użytkownika, do której kategorii chce wpisać dany wynik.  
Następnie sprawdza, która kategoria została wybrana i czy nie została wybrana po raz kolejny.

```
if (liczba == 1 && tab[0] == 0) { ... }  
if (liczba == 2 && tab[1] == 0) { ... }  
if (liczba == 3 && tab[2] == 0) { ... }  
if (liczba == 4 && tab[3] == 0) { ... }  
if (liczba == 5 && tab[4] == 0) { ... }  
if (liczba == 6 && tab[5] == 0) { ... }  
if (liczba == 7 && tab[6] == 0) { ... }  
if (liczba == 8 && tab[7] == 0) { ... }  
if (liczba == 9 && tab[8] == 0) { ... }  
if (liczba == 10 && tab[9] == 0) { ... }  
if (liczba == 11 && tab[10] == 0) { ... }  
if (liczba == 12 && tab[11] == 0) { ... }  
if (liczba == 13 && tab[12] == 0) { ... }
```

W każdym ifie następuje sprawdzenie czy spełnione zostały zasady by w danej kategorii mogła zostać przyznana pewna ilość punktów.

Na samym końcu następuje ustawienie zmiennych globalnych na pozycjach początkowych oraz sprawdzenie czy nie zostały wykorzystane już wszystkie kategorie.