

Robert Jaskólski

Projekt zaliczeniowy z programowania I

Tytuł: Monopoly

1. Opis projektu

Gra monopolu została wykonana jako projekt zaliczeniowy, przedmiotu - Programowanie I. W grę może grać jednocześnie trzech graczy. Gra polega która polega na pomnażaniu kapitału poprzez kupowanie nieruchomości i pobieranie czynszu od pozostałych graczy. Gracze poprzez swoje działania prowadzą do bankructwa innego gracza wówczas gra się kończy. Każdy gracz na początku gry otrzymuje 2500\$. Na planszy do gry znajdują się także pola oznaczone jako **szansa** wówczas graczowi, w zależności od szansy mogą zostać przyznane mu pieniądze, odebrane lub może zostać przesunięty. Na planszy znajduje się pole **podatek** które zobowiązuje do zapłaty należności w wysokości 200\$. Mapa posiada trzy pola neutralne czyli, **start**, **parking** i **impreza**. Gracz przekraczając **start** dostaje 400\$. Pole **wyspa** zatrzymuje gracza na dwie kolejki.

2. Funkcjonalności

Funkcjonalnościami głównymi gry monopoly są:

- generowanie planszy do gry,
- poruszanie się graczy po mapie,
- możliwość kupna miast,
- pieniądze które gracze mogą wydawać.

Funkcjonalnościami dodatkowymi gry monopoly są:

- pole szansa która losuje szansę dla gracza (szansa może dodać lub odjąć pieniądze gracza, lub przemieścić gracza na mapie),
- pole wyspa które wstrzymuje gracza na dwie kolejki,
- pole podatek które odejmuje graczowi jego pieniądze.

3. Spis funkcji

Plik source.cpp zawiera:

```
int main() . //Funkcja główna.
```

Plik gra_struktura.cpp zawiera:

```
void gra_struktura() //Funkcja odpowiedzialna za strukturę gry mianowicie  
za ruchy, rysowanie pionków, zarządzanie pieniędzmi.
```

```
int losowy_ruch() //Funkcja wylosowania liczby
```

```
void wyswietlanie_gotowki_i_miejsca() //Funkcja wyrysowania  
gotówki posiadanej przez graczy i ich miejsca na planszy.
```

```
void menu(int) //Funkcja wyświetla menu z możliwością rzutu kostki.
```

```
void menu_kupowania(int, int) //Funkcja wyświetla menu kupowania  
miasta dla pola które można kupić.
```

```
void szansa(int) //Funkcja odpowiedzialna za pole szansa.
```

```
bool sprawdz_czy_koniec(int,int,int) //Funkcja sprawdza czy któryś z  
graczy nie zbankrutował co jest równe z końcem gry.
```

Plik kupowanie.cpp zawiera:

```
void kupowanie(int int) //Funkcja odpowiedzialna za zmianę graficzną ceny  
danego pola na cenę czynszu oraz zaznacza który gracz kupił pole.
```

Plik regulamin.cpp zawiera:

```
void regulamin() //Funkcja odpowiedzialna za generację ekranu startowego  
wraz z regulaminem.
```

Plik rysowanie_pionkow.cpp zawiera:

```
void rysowanie_pionkow(int,int) //Funkcję odpowiedzialną za  
wyrysowanie pionków na planszy do gry.
```

Plik usuwanie_pionkow.cpp zawiera:

```
void usuwanie_pionkow(int) //Funkcja odpowiedzialna jest za usuwanie  
starego pionka czyli naszego poprzedniego miejsca.
```

Plik rysowanie_mapy.cp zawiera:

```
void rysowanie_mapy() //Łączy on niżej wymienione funkcję w jedność, co  
pozwala na wyrysowanie całej planszy.
```

```
void rysowanie_linipodNazwaZCena(int,int,int) //Funkcja  
odpowiedzialna za rysowanie z ceną linii pod nazwą | cena |.
```

```
void rysowanie_linipozioamej() //Generowanie wszystkich pól  
znajdujących się w pierwszym wierszu od startu do wyspy.
```

```
void rysowanie_pierwszejLini() //Linia z poziomą kreską górną.
```

```
void rysowanie_liniznazwaMiasta(int,int,int); //Generowanie linii z  
nazwą miasta | nazwa |.
```

```
void rysowanie_linipodNazwa() //Generowanie linii pustej | |.
```

```
void rysowanie_ostatniejLini() //Generowanie linii poziomej na końcu  
kwadratu.
```

```
void rysowanie_linipozioamejDolnej(); //Generuje całą linię poziomą od  
imprezy do parkingu.
```

```
void  
rysowanie_linipozioomychZdwomaMiastami(int,int,int,int,int,int,  
int) //Funkcja odpowiedzialna za generowanie wiersza gdzie występują tylko dwa  
miasta.
```

Plik setCursor.cpp zawiera:

```
void setCursor(int,int) //Funkcja odpowiedzialna za ustawienie kursora w  
konsoli na daną pozycję.
```

4. Ważne zmienne

Przykłady będą dla gracza A

```
int moneyA = 2500; //Zmienna przechowuje ile gotówki ma gracz A
```

```
int pozycja_gracza_A = 1; //Zmienna przechowuje pozycje gracza A na mapie.
```

```
int Los_gracza_A = 0; //Zmienna ta jest nadpisywana i jest do niej wpisana randomowa liczba z funkcji losowy_ruch().
```

```
int czy_moze_A = 1; //Zmienna ta jest ważna dla pola wyspa ponieważ wyspa ustawia go na -1, co stopuje gracza na dwie kolejki.
```

```
int ceny_miast1[32] //Zmienna przechowuje cenny miast.
```

```
int kto_kupil[32] //Zmienna przechowuje kto jest właścicielem danego miasta.
```

```
int wartosc_zaplaty1[32] //Zmienna przechowuje wartości czynszu, który gracze muszą płacić.
```

```
string miasta[32] //Tablica z zapisanymi nazwami miast.
```

5. Przykładowy kod

```
int losowy_ruch()
{
    srand(time(NULL));
    int a;
    a = rand() % 6;
    a = a + 1;
    return a;
}
```

```
void setCursor(int x, int y)
{
    COORD c;
    c.X = x - 1;
    c.Y = y - 1;
    SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE),
c);
}
```

//Ruch gracza A w gra_struktura.cpp

```
if (czy_moze_A == 1)
{
    Los_gracza_A = losowy_ruch();
    rysowanie_pionkow(pozycja_gracza_A, 1);
    setCursor(2, 46);
    menu(1);
    Sleep(500);
    for (int i = 0; i < 6; i++)
    {
        setCursor(1, 46 + i);
        cout << "
";
    }
    usuwanie_pionkow(pozycja_gracza_A);
    rysowanie_pionkow(pozycja_gracza_B, 2);
    rysowanie_pionkow(pozycja_gracza_C, 3);
    pozycja_gracza_A += Los_gracza_A;
    if (pozycja_gracza_A > 32)
    {
        pozycja_gracza_A = pozycja_gracza_A % 32;
        moneyA += 400;
    }
    wyswietlanie_gotowki_i_miejsca();
    rysowanie_pionkow(pozycja_gracza_A, 1);
    setCursor(2, 46);
    menu_kupowania(1, pozycja_gracza_A);
    for (int i = 0; i < 6; i++)
    {
        setCursor(2, 46 + i);
        cout << "
";
    }
}
else
    czy_moze_A += 1;
wyswietlanie_gotowki_i_miejsca();
if (sprawdz_czy_koniec(moneyA, moneyB, moneyC) == true)
{
    system("cls");
    setCursor(120, 10);
    cout << "KONIEC GRY";
    break;
}
```