### 23.06.2017 Wrocław

### Wydział Elektronika

### Kierunek: Informatyka

### Nazwisko i Imię: Michał Tomaszewicz

### NR indeksu: 235568

### Grupa: Pt. 11:15 – 13:45, C-16, s. L2.4

### Semestr: 2016/17 LATO

### Prowadzący: mgr inż. Michał Koziarski

### PROGRAMOWANIE OBIEKTOWE

### Program semestralny C++

### wersja Visual Studio *2015*

TEMAT

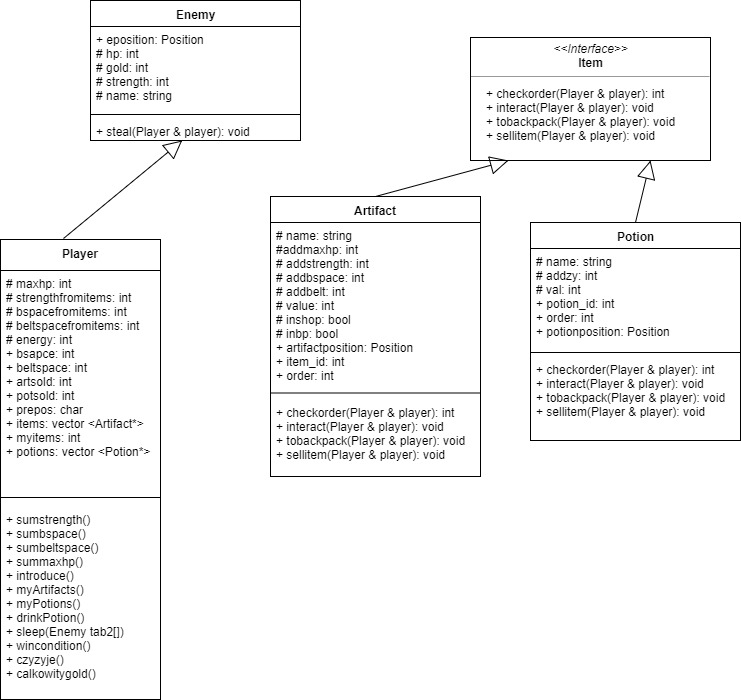
Gra: HELP YOUR BROTHER

OCENA:

## Opis programu

Mój program to niewielka gra konsolowa z dość prostym, ale działającym na bieżąco interfejsem graficznym. Wcielamy się w postać, której brat został porwany. Naszym celem jest wpłacić za niego okup w wielkości X złota. W trakcie rozgrywki poruszamy się po mapie, po której porozrzucani są przeciwnicy, z którymi możliwa jest walka, różne lokacje z odpowiednimi interakcjami oraz przedmioty. Te ostatnie dzielą się na artefakty, które wzmacniają poszczególne statystyki naszego bohatera oraz mikstury, które przywracają brakujące punkty życia. Jedną z lokacji jest sklep, który umożliwia kupno oraz sprzedaż przedmiotów. Pełni on kluczową rolę w rozgrywce, gdyż znacząco przybliża nas do uzbierania potrzebnej ilości złota. Walka z przeciwnikami jest konieczna do ukończenia gry. W walce możemy, w zależności od jej wyniku, wygrać lub stracić pewną ilość złota. Mechanika jest dość prosta. Wymaga od użytkownika znajomości mnożenia i polega na rozwiązywaniu pewnej ilości działań. Gra kończy się przegraną gdy gracz umiera, bądź nie ma wystarczającej ilości złota aby się pojedynkować. Wygrywamy gdy uda nam się wpłacić okup i uratować brata.

## Diagram klas



## Listing kodu

SAVE\_YOUR\_BROTHER.cpp

// SAVE\_YOUR\_BROTHER.cpp: Określa punkt wejścia dla aplikacji konsoli.

//

#include "stdafx.h"

#include <iostream>

#include <cstdlib>

#include <fstream>

#include <stdlib.h>

#include <windows.h>

#include <conio.h>

#include "Enemy.h"

#include "Player.h"

#include "Item.h"

#include <vector>

#include <ctime>

#include "Artifact.h"

#include "Potion.h"

using namespace std;

struct Position

{

int x;

int y;

};

Artifact tab[6];///////////////////////////////////////////////////tablica artefaktow

Potion tab1[5];/////////////////////////////////////////////tablica potionow

Enemy tab2[6];//////////////// tablica przeciwnikow

int rozmiar = 10;

char \*\* mapa = new char \*[rozmiar];

class Player;

void sklep(Player & player);// predefinicja sklepu

void home(Player & player); // predefinicja domu

void create\_artifact(ifstream & file, Artifact & ar)

{

file >> ar.name;

file >> ar.addmaxhp;

file >> ar.addstrength;

file >> ar.addbspace;

file >> ar.addbelt;

file >> ar.value;

file >> ar.artifactposition.x;

file >> ar.artifactposition.y;

file >> ar.item\_id;

file >> ar.inshop;

file >> ar.inbp;

}

void create\_potion(ifstream & file, Potion & pot)

{

file >> pot.name;

file >> pot.addzy;

file >> pot.val;

file >> pot.potionposition.x;

file >> pot.potionposition.y;

file >> pot.potion\_id;

}

void create\_enemy(ifstream & file, Enemy & enem)

{

file >> enem.name;

file >> enem.hp;

file >> enem.gold;

file >> enem.strength;

file >> enem.eposition.x;

file >> enem.eposition.y;

}

int main()

{

ifstream plik;

plik.open("mapa.txt");

for (int i = 0; i < rozmiar; i++) // dynamiczna tablica typu char jako mapa

{

mapa[i] = new char[rozmiar];

}

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

{

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

{

plik >> mapa[i][j];

}

}

plik.close(); //koniec pracy z plikiem

int ilosc;

plik.open("artifacts.txt");////////////////////////////////////////////wczytujemy artefakty z pliku

plik >> ilosc;

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

{

create\_artifact(plik, tab[i]);

}

plik.close();

int il;

plik.open("potions.txt");///////////////////////////////////////////////////wczytujemy potiony z pliku

plik >> il;

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

{

create\_potion(plik, tab1[i]);

}

plik.close();

int ilo;

plik.open("enemies.txt");

plik >> ilo;

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

{

create\_enemy(plik, tab2[i]);

}

cout << " psss. d s. d b Ssssss Ss sS sSSSs d b d ss. " << endl;

cout << " d S O S S S S S S S S S S b " << endl;

cout << " Y S b S S S S S S S S S P " << endl;

cout << " Sss. SssSSO S S Ssssss S S S S S S sS' " << endl;

cout << " b S O S S S S S S S S S S " << endl;

cout << " P S O S S S S S S S S S S " << endl;

cout << " `Sss' P P ""SS PssSSs P ""sssss"" ""sssss"" P P " << endl;

cout << " " << endl;

cout << " ssssp dsss. sSSSs sssssssss S S Ssssss sddss. " << endl;

cout << " S b S b S S S S S S S b " << endl;

cout << " S P S P S S S S S S S P " << endl;

cout << " SSSSS' SSsS' S S S SSSSSS Ssssss S sS' " << endl;

cout << " S b S S S S S S S S S S " << endl;

cout << " S P S S S S S S S S S S " << endl;

cout << " PSSSP P P ""ssss"" S P P PssSSs P P " << endl;

Sleep(5000);

cout << endl;

cout << " nacisnij <dowolny klawisz> aby kontynuowac" << endl;

char a11 = \_getch();

switch (a11)

{

default:

system("cls");

break;

}

cout << "Twojemu bratu grozi niebespieczenstwo!" << endl;

cout << "Porwali go zli ludzie i zadaja okupu." << endl;

cout << "Cena za brata wynosi 200 zlota" << endl;

cout << "Twoim zadaniem bedzie uzbierac zadana sume i wplacic okup za brata" << endl;

Sleep(3000);

cout << endl;

cout << "nacisnij <dowolny klawisz> aby kontynuowac" << endl;

char a10 = \_getch();

switch (a10)

{

default:

system("cls");

break;

}

cout << "-----INSTRUKCJA-----" << endl;

cout << endl;

cout << "Znajdujesz sie na pozycji X" << endl;

cout << "Litera W oznacza lokalizacje porywaczy, wlasnie tam musisz wplacic okup (200 zlota)" << endl;

cout << "Nie mozesz poruszac sie poza mape ORAZ po znakach '\_'" << endl;

cout << "Litera P oznaczona jest mikstura, ktora mozesz podniesc, a nastepnie wypic" << endl;

cout << "Po podniesieniu mikstury laduje ona na Twoim pasku" << endl;

cout << "Litera I oznaczony jest przedmiot, ktory moze zwiekszyc Twoje statystyki" << endl;

cout << "Po podniesieniu przedmioty laduja w Towim plecaku" << endl;

cout << "Litera H oznaczony jest Twoj dom. mozesz w nim odpoczac" << endl;

cout << "Litera E oznaczeni sa wrogowie. Posiadaja oni swoje zloto. Mozesz sie z nimi" << endl;

cout << "pojedynkowac o zloto, ktore przyda Ci sie do oplacenia okupu." << endl;

cout << "Aby walczyc musisz posiadac conajmniej 10 zlota. Jesli przegrasz to tracisz 10 zlota." << endl;

cout << "Kazdy pojedynek jest dosc wyczerpujacy. Tracisz w nich energie. Gdy Twoja energia osiagnie niskipoziom" << endl;

cout << "nie bedziesz juz mogl brac udzialu w pojedynkach. Nalezy wtedy udac sie do domu i zaczerpnac odpoczynku" << endl;

cout << "Litera S oznaczony jest sklep. Za zlotomozesz tam kupowac oraz sprzedawac posiadane przedmioty oraz mikstury" << endl;

cout << endl;

cout << "Gdy Twoje punkty zycia spadna do 0 to przegrywasz." << endl;

cout << "Gdy suma wartosci Twoich przedmiotw, wartosci potionow i ilosci posiadanego zlota" << endl;

cout << "bedzie mniejsza niz 10 to przegrywasz" << endl;

cout << "Poruszasz sie za pomoca strzalek" << endl;

cout << "Aby zakonczyc gre nacisnij Esc" << endl;

cout << endl;

cout << "Oto Twoja mapa!" << endl;// wypisywanie mapy

cout << endl;

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

{

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

{

cout << mapa[i][j] << " ";

}

cout << endl;

}

cout << endl;

cout << "nacisnij <dowolny klawisz> aby kontynuowac" << endl;

char a12 = \_getch();

switch (a12)

{

default:

system("cls");

break;

}

system("cls");

///////////////////////////////////////////////////////////////////

char a13;

Position me;

Position end;

Position eloss;

Position shop;

Position hom;

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

{

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

{

if (mapa[i][j] == 'X')// znajduje moja pozycje na mapie

{

me.x = j;

me.y = i;

}

if (mapa[i][j] == 'W')// znajduje pozycje wygranej

{

end.x = j;

end.y = i;

}

if (mapa[i][j] == 'L')// znajduje pozycje przegranej

{

eloss.x = j;

eloss.y = i;

}

if (mapa[i][j] == 'S')// znajduje pozycje sklepu

{

shop.x = j;

shop.y = i;

}

if (mapa[i][j] == 'H')// znajduje pozycje domu

{

hom.x = j;

hom.y = i;

}

}

}

///////////////////////////////////////////////////////////////////////// klasy

Player ja("Michal", 100, 80, 10, 1, me.x, me.y, 4, 2); // miejsce w plecaku, miejsce na pasku

/////////////////////////////////////////////////////////////////////

cout << "Moja pozycja to ";

cout << "[" << ja.eposition.x << ", " << ja.eposition.y << "]" << endl;

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

{

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

{

cout << mapa[i][j] << " ";

}

cout << endl;

}

cout << endl;

cout << "ENTER - aby otrzymac pomoc" << endl;

cout << "s - aby wyswietlic informacje o graczu" << endl;

cout << "d - aby wypic miksture." << endl;

bool winnn = 0;

/////////////////////////////////////////////////////////////////////////////////////////////////////////////

while (winnn == 0) //petla

{

unsigned char znak = \_getch();

system("cls");

switch (znak)

{

case 72: //strzałka w górę

if ((ja.eposition.y - 1 >= 0) && (mapa[ja.eposition.y - 1][ja.eposition.x] != '\_'))

{

//problemy ze wspolrzednymi: rozwiazany

if (mapa[ja.eposition.y - 1][ja.eposition.x] != 'I' && mapa[ja.eposition.y - 1][ja.eposition.x] != 'P')

{

mapa[ja.eposition.y][ja.eposition.x] = ja.prepos;

ja.prepos = mapa[ja.eposition.y - 1][ja.eposition.x];

mapa[ja.eposition.y - 1][ja.eposition.x] = 'X';

ja.eposition.y -= 1;

}

else

{

if (ja.sumbspace() > 0 && mapa[ja.eposition.y - 1][ja.eposition.x] == 'I')

{

mapa[ja.eposition.y][ja.eposition.x] = 'o';

mapa[ja.eposition.y - 1][ja.eposition.x] = 'X';

ja.eposition.y -= 1;

}

else

{

if (ja.sumbeltspace() >0 && mapa[ja.eposition.y - 1][ja.eposition.x] == 'P')

{

mapa[ja.eposition.y][ja.eposition.x] = 'o';

mapa[ja.eposition.y - 1][ja.eposition.x] = 'X';

ja.eposition.y -= 1;

}

else

{

mapa[ja.eposition.y][ja.eposition.x] = ja.prepos;

ja.prepos = mapa[ja.eposition.y - 1][ja.eposition.x];

mapa[ja.eposition.y - 1][ja.eposition.x] = 'X';

ja.eposition.y -= 1;

}

}

}

if (ja.eposition.x == shop.x && ja.eposition.y == shop.y)/////////////////////////////////////czy w sklepie

{

sklep(ja);

}

if (ja.eposition.x == hom.x && ja.eposition.y == hom.y)/////////////////////////////////////czy w domu

{

home(ja);

}

cout << "Moja pozycja to ";

cout << "[" << ja.eposition.x << ", " << ja.eposition.y << "]" << endl;

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

{

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

{

cout << mapa[i][j] << " ";

}

cout << endl;

}

cout << endl;

cout << "ENTER - aby otrzymac pomoc" << endl;

cout << "s - aby wyswietlic informacje o graczu" << endl;

cout << "d - aby wypic miksture." << endl;

}

else

{

cout << "Moja pozycja to ";

cout << "[" << ja.eposition.x << ", " << ja.eposition.y << "]" << endl;

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

{

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

{

cout << mapa[i][j] << " ";

}

cout << endl;

}

cout << endl;

cout << "ENTER - aby otrzymac pomoc" << endl;

cout << "s - aby wyswietlic informacje o graczu" << endl; cout << endl;

cout << "d - aby wypic miksture." << endl;

cout << "Nie mozesz tam pojsc!" << endl;

}

break;

case 80: //strzałka w dół

if (ja.eposition.y + 1 < rozmiar && (mapa[ja.eposition.y + 1][ja.eposition.x] != '\_'))

{

if (mapa[ja.eposition.y + 1][ja.eposition.x] != 'I' && mapa[ja.eposition.y + 1][ja.eposition.x] != 'P')

{

mapa[ja.eposition.y][ja.eposition.x] = ja.prepos;

ja.prepos = mapa[ja.eposition.y + 1][ja.eposition.x];

mapa[ja.eposition.y + 1][ja.eposition.x] = 'X';

ja.eposition.y += 1;

}

else

{

if (ja.sumbspace() > 0 && mapa[ja.eposition.y + 1][ja.eposition.x] == 'I')

{

mapa[ja.eposition.y][ja.eposition.x] = 'o';

mapa[ja.eposition.y + 1][ja.eposition.x] = 'X';

ja.eposition.y += 1;

}

else

{

if (ja.sumbeltspace() >0 && mapa[ja.eposition.y + 1][ja.eposition.x] == 'P')

{

mapa[ja.eposition.y][ja.eposition.x] = 'o';

mapa[ja.eposition.y + 1][ja.eposition.x] = 'X';

ja.eposition.y += 1;

}

else

{

mapa[ja.eposition.y][ja.eposition.x] = ja.prepos;

ja.prepos = mapa[ja.eposition.y + 1][ja.eposition.x];

mapa[ja.eposition.y + 1][ja.eposition.x] = 'X';

ja.eposition.y += 1;

}

}

}

if (ja.eposition.x == shop.x && ja.eposition.y == shop.y)/////////////////////////////////////czy w sklepie

{

sklep(ja);

}

if (ja.eposition.x == hom.x && ja.eposition.y == hom.y)/////////////////////////////////////czy w domu

{

home(ja);

}

cout << "Moja pozycja to ";

cout << "[" << ja.eposition.x << ", " << ja.eposition.y << "]" << endl;

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

{

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

{

cout << mapa[i][j] << " ";

}

cout << endl;

}

cout << endl;

cout << "ENTER - aby otrzymac pomoc" << endl;

cout << "s - aby wyswietlic informacje o graczu" << endl;

cout << "d - aby wypic miksture." << endl;

}

else

{

cout << "Moja pozycja to ";

cout << "[" << ja.eposition.x << ", " << ja.eposition.y << "]" << endl;

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

{

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

{

cout << mapa[i][j] << " ";

}

cout << endl;

}

cout << endl;

cout << "ENTER - aby otrzymac pomoc" << endl;

cout << "s - aby wyswietlic informacje o graczu" << endl;

cout << "d - aby wypic miksture." << endl;

cout << endl;

cout << "Nie mozesztam pojsc!" << endl;

}

break;

case 75: //strzałka w lewo

if ((ja.eposition.x - 1 >= 0) && (mapa[ja.eposition.y][ja.eposition.x - 1] != '\_'))

{

if (mapa[ja.eposition.y][ja.eposition.x - 1] != 'I' && mapa[ja.eposition.y][ja.eposition.x - 1] != 'P')

{

mapa[ja.eposition.y][ja.eposition.x] = ja.prepos;

ja.prepos = mapa[ja.eposition.y][ja.eposition.x - 1];

mapa[ja.eposition.y][ja.eposition.x - 1] = 'X';

ja.eposition.x -= 1;

}

else

{

if (ja.sumbspace() > 0 && mapa[ja.eposition.y][ja.eposition.x - 1] == 'I')

{

mapa[ja.eposition.y][ja.eposition.x] = 'o';

mapa[ja.eposition.y][ja.eposition.x - 1] = 'X';

ja.eposition.x -= 1;

}

else

{

if (ja.sumbeltspace() > 0 && mapa[ja.eposition.y][ja.eposition.x - 1] == 'P')

{

mapa[ja.eposition.y][ja.eposition.x] = 'o';

mapa[ja.eposition.y][ja.eposition.x - 1] = 'X';

ja.eposition.x -= 1;

}

else

{

mapa[ja.eposition.y][ja.eposition.x] = ja.prepos;

ja.prepos = mapa[ja.eposition.y][ja.eposition.x - 1];

mapa[ja.eposition.y][ja.eposition.x - 1] = 'X';

ja.eposition.x -= 1;

}

}

}

if (ja.eposition.x == shop.x && ja.eposition.y == shop.y)/////////////////////////////////////czy w sklepie

{

sklep(ja);

}

if (ja.eposition.x == hom.x && ja.eposition.y == hom.y)/////////////////////////////////////czy w domu

{

home(ja);

}

cout << "Moja pozycja to ";

cout << "[" << ja.eposition.x << ", " << ja.eposition.y << "]" << endl;

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

{

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

{

cout << mapa[i][j] << " ";

}

cout << endl;

}

cout << endl;

cout << "ENTER - aby otrzymac pomoc" << endl;

cout << "s - aby wyswietlic statystyki" << endl;

cout << "d - aby wypic miksture." << endl;

}

else

{

cout << "Moja pozycja to ";

cout << "[" << ja.eposition.x << ", " << ja.eposition.y << "]" << endl;

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

{

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

{

cout << mapa[i][j] << " ";

}

cout << endl;

}

cout << endl;

cout << "ENTER - aby otrzymac pomoc" << endl;

cout << "s - aby wyswietlic informacje o graczu" << endl;

cout << "d - aby wypic miksture." << endl;

cout << endl;

cout << "Nie mozesz tam pojsc!" << endl;

}

break;

case 77: //strzałka w prawo

if ((ja.eposition.x + 1 <rozmiar) && (mapa[ja.eposition.y][ja.eposition.x + 1] != '\_'))

{

if (mapa[ja.eposition.y][ja.eposition.x + 1] != 'I' && mapa[ja.eposition.y][ja.eposition.x + 1] != 'P')

{

mapa[ja.eposition.y][ja.eposition.x] = ja.prepos;

ja.prepos = mapa[ja.eposition.y][ja.eposition.x + 1];

mapa[ja.eposition.y][ja.eposition.x + 1] = 'X';

ja.eposition.x += 1;

}

else

{

if (ja.sumbspace() > 0 && mapa[ja.eposition.y][ja.eposition.x + 1] == 'I')

{

mapa[ja.eposition.y][ja.eposition.x] = 'o';

mapa[ja.eposition.y][ja.eposition.x + 1] = 'X';

ja.eposition.x += 1;

}

else

{

if (ja.sumbeltspace() > 0 && mapa[ja.eposition.y][ja.eposition.x + 1] == 'P')

{

mapa[ja.eposition.y][ja.eposition.x] = 'o';

mapa[ja.eposition.y][ja.eposition.x + 1] = 'X';

ja.eposition.x += 1;

}

else

{

mapa[ja.eposition.y][ja.eposition.x] = ja.prepos;

ja.prepos = mapa[ja.eposition.y][ja.eposition.x + 1];

mapa[ja.eposition.y][ja.eposition.x + 1] = 'X';

ja.eposition.x += 1;

}

}

}

if (ja.eposition.x == shop.x && ja.eposition.y == shop.y)/////////////////////////////////////czy w sklepie

{

sklep(ja);

}

if (ja.eposition.x == hom.x && ja.eposition.y == hom.y)/////////////////////////////////////czy w domu

{

home(ja);

}

cout << "Moja pozycja to ";

cout << "[" << ja.eposition.x << ", " << ja.eposition.y << "]" << endl;

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

{

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

{

cout << mapa[i][j] << " ";

}

cout << endl;

}

cout << endl;

cout << "ENTER - aby otrzymac pomoc" << endl;

cout << "s - aby wyswietlic statystyki" << endl;

cout << "d - aby wypic miksture." << endl;

}

else

{

cout << "Moja pozycja to ";

cout << "[" << ja.eposition.x << ", " << ja.eposition.y << "]" << endl;

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

{

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

{

cout << mapa[i][j] << " ";

}

cout << endl;

}

cout << endl;

cout << "ENTER - aby otrzymac pomoc" << endl;

cout << "s - aby wyswietlic statystyki" << endl;

cout << "d - aby wypic miksture." << endl;

cout << endl;

cout << "Nie mozesz tam pojsc!" << endl;

}

break;

case 13:

system("cls");

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

{

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

{

cout << mapa[i][j] << " ";

}

cout << endl;

}

cout << endl;

cout << "Znajdujesz sie na pozycji X" << endl;

cout << "Litera W oznacza lokalizacje porywaczy, wlasnie tam musisz wplacic okup (200 zlota)" << endl;

cout << "Nie mozesz poruszac sie poza mape ORAZ po znakach '\_'" << endl;

cout << "Litera P oznaczona jest mikstura, ktora mozesz podniesc, a nastepnie wypic" << endl;

cout << "Po podniesieniu mikstury laduje ona na Twoim pasku" << endl;

cout << "Litera I oznaczony jest przedmiot, ktory moze zwiekszyc Twoje statystyki" << endl;

cout << "Po podniesieniu przedmioty laduja w Towim plecaku" << endl;

cout << "Litera H oznaczony jest Twoj dom. mozesz w nim odpoczac" << endl;

cout << "Litera E oznaczeni sa wrogowie. Posiadaja oni swoje zloto. Mozesz sie z nimi" << endl;

cout << "pojedynkowac o zloto, ktore przyda Ci sie do oplacenia okupu." << endl;

cout << "Aby walczyc musisz posiadac conajmniej 10 zlota. Jesli przegrasz to tracisz 10 zlota." << endl;

cout << "Kazdy pojedynek jest dosc wyczerpujacy. Tracisz w nich energie. Gdy Twoja energia osiagnie niskipoziom" << endl;

cout << "nie bedziesz juz mogl brac udzialu w pojedynkach. Nalezy wtedy udac sie do domu i zaczerpnac odpoczynku" << endl;

cout << "Litera S oznaczony jest sklep. Za zlotomozesz tam kupowac oraz sprzedawac posiadane przedmioty oraz mikstury" << endl;

cout << endl;

cout << "Gdy Twoje punkty zycia spadna do 0 to przegrywasz." << endl;

cout << "Gdy suma wartosci Twoich przedmiotw, wartosci potionow i ilosci posiadanego zlota" << endl;

cout << "bedzie mniejsza niz 10 to przegrywasz" << endl;

cout << "Poruszasz sie za pomoca strzalek" << endl;

cout << "Aby zakonczyc gre nacisnij Esc" << endl;

Sleep(2000);

cout << endl;

cout << "nacisnij <dowolny klawisz> aby kontynuowac" << endl;

a13 = \_getch();

switch (a13)

{

default:

system("cls");

cout << "Moja pozycja to ";

cout << "[" << ja.eposition.x << ", " << ja.eposition.y << "]" << endl;

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

{

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

{

cout << mapa[i][j] << " ";

}

cout << endl;

}

cout << endl;

cout << "ENTER - aby otrzymac pomoc" << endl;

cout << "s - aby wyswietlic statystyki" << endl;

cout << "d - aby wypic miksture." << endl;

break;

}

break;

case 27:

char decision;

system("cls");

cout << "Czy na pewno chcesz zakonczyc gre?" << endl;

cout << "t - tak, chce wyjsc z programu" << endl;

cout << "<dowolny przycisk> -aby wrocic do gry" << endl;

decision = \_getch();

switch (decision)

{

case 't':

system("cls");

system("Pause");

return 0;

break;

default:

system("cls");

cout << "Moja pozycja to ";

cout << "[" << ja.eposition.x << ", " << ja.eposition.y << "]" << endl;

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

{

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

{

cout << mapa[i][j] << " ";

}

cout << endl;

}

cout << endl;

cout << "ENTER - aby otrzymac pomoc" << endl;

cout << "s - aby wyswietlic statystyki" << endl;

cout << "d - aby wypic miksture." << endl;

break;

}

break;

case 's':

system("cls");

cout << "Moja pozycja to ";

cout << "[" << ja.eposition.x << ", " << ja.eposition.y << "]" << endl;

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

{

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

{

cout << mapa[i][j] << " ";

}

cout << endl;

}

cout << endl;

cout << "ENTER - aby otrzymac pomoc" << endl;

cout << "s - aby wyswietlic statystyki" << endl;

cout << "d - aby wypic miksture." << endl;

ja.introduce();

ja.myArtifacts();

ja.myPotions();

break;

case 'd':

system("cls");

cout << "Moja pozycja to ";

cout << "[" << ja.eposition.x << ", " << ja.eposition.y << "]" << endl;

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

{

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

{

cout << mapa[i][j] << " ";

}

cout << endl;

}

cout << endl;

cout << "ENTER - aby otrzymac pomoc" << endl;

cout << "s - aby wyswietlic statystyki" << endl;

cout << "d - aby wypic miksture." << endl;

ja.drinkPotion();

break;

default:

cout << "Moja pozycja to ";

cout << "[" << ja.eposition.x << ", " << ja.eposition.y << "]" << endl;

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

{

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

{

cout << mapa[i][j] << " ";

}

cout << endl;

}

cout << endl;

cout << "ENTER - aby otrzymac pomoc" << endl;

cout << "s - aby wyswietlic statystyki" << endl;

cout << "d - aby wypic miksture." << endl;

break;

}

//////////////////////// interakcja z przeciwnikiem

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

{

if (ja.eposition.x == tab2[i].eposition.x && ja.eposition.y == tab2[i].eposition.y)

{

tab2[i].steal(ja);

}

}

////////////////////////interakcja z przedmiotem/////////////////////////////////////////////////////szukać w tablicy

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

{

if (ja.eposition.x == tab[i].artifactposition.x && ja.eposition.y == tab[i].artifactposition.y)

{

if (ja.sumbspace() > 0)

{

tab[i].tobackpack(ja);

}

else

{

cout << endl;

cout << "Nie mozesz podniesc tego przedmiotu. Brak miejsca w ekwipunku." << endl;

}

}

}

/////////////////////////interakcja z potionem

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

{

if (ja.eposition.x == tab1[i].potionposition.x && ja.eposition.y == tab1[i].potionposition.y)

{

if (ja.sumbeltspace() > 0)

{

tab1[i].tobackpack(ja);

}

else

{

cout << endl;

cout << "Nie mozesz podniesc tego eliiksiru. Brak miejsca na pasku" << endl;

}

}

}

if ((ja.eposition.x == end.x) && (ja.eposition.y == end.y))

{

if (ja.wincondition() == 1)

{

winnn = 1;

}

else

{

cout << endl;

cout << "-Nie masz zlota? A juz mi stad zmykaj!" << endl;

}

}

if (ja.czyzyje() == 0)

{

system("cls");

cout << "Przegrales" << endl;

system("pause");

return 0;

}

if (ja.calkowitygold() < 10)

{

system("cls");

cout << "Przegrales" << endl;

system("pause");

return 0;

}

}

cout << "Twoj brat jest wolny" << endl;

Sleep(3000);

system("cls");

cout << "Brawo udalo Ci sie ukonczyc gre!" << endl;

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

{

delete[] mapa[i];

}

delete[] mapa;

system("pause");

return 0;

}

//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

void sklep(Player & player)

{

bool exit = 0;

while (exit == 0) {

cout << endl;

cout << "Zloto: " << player.gold << endl;

cout << "-----SKLEP-----" << endl;

cout << " a - aby sprzedac przedmioty" << endl;

cout << " b - aby kupic przedmioty" << endl;

cout << "<dowlony klawisz> - aby wyjsc ze sklepu" << endl;

unsigned char znak1 = \_getch();

unsigned char znak2;

unsigned char znak3;

//int suma = (player.myitems.size() + player.mypotions.size());

switch (znak1)

{

case 'a':

system("cls");

cout << "Zloto: " << player.gold << endl;

cout << endl;

cout << "Przedmioty:" << endl;

if (player.items.size() == 0)

{

cout << "Brak przedmiotow" << endl;

}

else

{

for (int i = 0; i < player.myitems.size(); i++)

{

cout << i << " - sprzedaj " << (\*player.items[i]).name << endl;

cout << " + " << (\*player.items[i]).addmaxhp << " maksymalnego zycia" << endl;

cout << " + " << (\*player.items[i]).addstrength << " sily" << endl;

cout << " + " << (\*player.items[i]).addbspace << " miejsca w ekwipunku" << endl;

cout << " + " << (\*player.items[i]).addbelt << " miejsca na pasku z miksturami" << endl;

cout << " wartosc: " << (\*player.items[i]).value << " zlota" << endl;

}

}

cout << endl;

cout << "Mikstury:" << endl;

if (player.potions.size() == 0)

{

cout << "Brak misktur" << endl;

}

else

{

for (int j = 97; j < (97 + player.potions.size()); j++)

{

cout << (char)j << " - sprzedaj " << (\*player.potions[j - 97]).name << endl;

cout << " + " << (\*player.potions[j - 97]).addzy << "punktow zdrowia" << endl;

cout << " wartosc: " << (\*player.potions[j - 97]).val << " zlota" << endl;

}

}

cout << endl;

cout << "Swoj wybor zatwierdz ENTERem." << endl;

cout << "<dowolny znak> - aby nic nie sprzedac" << endl;

cin >> znak3;

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////// sprzedawanie

Item\* wsk;

for (int i = 0; i < player.myitems.size(); i++)

{

if ((int)(znak3) == 48 + i)

{

wsk = player.items[i];

wsk->sellitem(player);

}

}

for (int j = 0; j < player.mypotions.size(); j++)

{

if ((int)(znak3) == 97 + j)

{

wsk = player.potions[j];

wsk->sellitem(player);

}

}

Sleep(3000);

system("cls");

break;

case 'b':

system("cls");

cout << "Dostepne zloto: " << player.gold << endl;

cout << endl;

cout << "Przedmioty:" << endl;

for (int i = 0; i < 3; i++)

{

cout << " " << i << " - kup " << tab[i].name << endl;

cout << " + " << tab[i].addmaxhp << " maksymalnego zycia" << endl;

cout << " + " << tab[i].addstrength << " sily" << endl;

cout << " + " << tab[i].addbspace << " miejsca w ekwipunku" << endl;

cout << " + " << tab[i].addbelt << " miejsca na pasku z miksturami" << endl;

cout << " wartosc: " << tab[i].value << " zlota" << endl;

}

cout << endl;

cout << "Mikstury:" << endl;

for (int j = 3; j < 6; j++)

{

cout << " " << j << " - kup " << tab1[j - 3].name << endl;

cout << " + " << tab1[j - 3].addzy << "punktow zdrowia" << endl;

cout << " wartosc: " << tab1[j - 3].val << " zlota" << endl;

}

cout << endl;

cout << "Swoj wybor zatwierdz ENTERem." << endl;

cout << "<dowolny znak> - aby nic nie kupic" << endl;

cin >> znak2;

//znak2 = \_getch();

if (znak2 == '0')

{

if (player.sumbspace() < 1)

{

cout << "Brak miejsca w plecaku" << endl;

Sleep(3000);

system("cls");

}

else

{

if (tab[0].inbp == 1)

{

cout << "Mozesz posiadac tylko jeden taki przedmiot!" << endl;

Sleep(3000);

system("cls");

}

else

{

if (player.gold < tab[0].value)

{

cout << "Nie posiadasz wystarczajacej ilosci zlota" << endl;

Sleep(3000);

system("cls");

}

else

{

tab[0].tobackpack(player);

cout << "Kupiles " << tab[0].name << endl;

player.gold -= tab[0].value;

cout << "-" << tab[0].value << " zlota" << endl;

}

Sleep(3000);

system("cls");

}

}

}

else if (znak2 == '1')

{

if (player.sumbspace() < 1)

{

cout << "Brak miejsca w plecaku" << endl;

Sleep(3000);

system("cls");

}

else

{

if (tab[1].inbp == 1)

{

cout << "Mozesz posiadac tylko jeden taki przedmiot!" << endl;

Sleep(3000);

system("cls");

}

else

{

if (player.gold < tab[1].value)

{

cout << "Nie posiadasz wystarczajacej ilosci zlota" << endl;

Sleep(3000);

system("cls");

}

else

{

tab[1].tobackpack(player);

cout << "Kupiles " << tab[1].name << endl;

player.gold -= tab[1].value;

cout << "-" << tab[1].value << " zlota" << endl;

}

Sleep(3000);

system("cls");

}

}

}

else if (znak2 == '2')

{

if (player.sumbspace() < 1)

{

cout << "Brak miejsca w plecaku" << endl;

Sleep(3000);

system("cls");

}

else

{

if (tab[2].inbp == 1)

{

cout << "Mozesz posiadac tylko jeden taki przedmiot!" << endl;

Sleep(3000);

system("cls");

}

else

{

if (player.gold < tab[2].value)

{

cout << "Nie posiadasz wystarczajacej ilosci zlota" << endl;

Sleep(3000);

system("cls");

}

else

{

tab[2].tobackpack(player);

cout << "Kupiles " << tab[2].name << endl;

player.gold -= tab[2].value;

cout << "-" << tab[2].value << " zlota" << endl;

}

Sleep(3000);

system("cls");

}

}

}

else if (znak2 == '3') //////////////////////////////////////////////////////////////// potiony

{

if (player.sumbeltspace() < 1)

{

cout << "Brak miejsca na pasku" << endl;

Sleep(3000);

system("cls");

}

else

{

if (player.gold < tab1[0].val)

{

cout << "Nie posiadasz wystarczajacej ilosci zlota" << endl;

Sleep(3000);

system("cls");

}

else

{

tab1[0].tobackpack(player);

cout << "Kupiles " << tab1[0].name << endl;

player.gold -= tab1[0].val;

cout << "-" << tab1[0].val << " zlota" << endl;

}

Sleep(3000);

system("cls");

}

}

else if (znak2 == '4')

{

if (player.sumbeltspace() < 1)

{

cout << "Brak miejsca na pasku" << endl;

Sleep(3000);

system("cls");

}

else

{

if (player.gold < tab1[1].val)

{

cout << "Nie posiadasz wystarczajacej ilosci zlota" << endl;

Sleep(3000);

system("cls");

}

else

{

tab1[1].tobackpack(player);

cout << "Kupiles " << tab1[1].name << endl;

player.gold -= tab1[1].val;

cout << "-" << tab1[1].val << " zlota" << endl;

}

Sleep(3000);

system("cls");

}

}

else if (znak2 == '5')

{

if (player.sumbeltspace() < 1)

{

cout << "Brak miejsca na pasku" << endl;

Sleep(3000);

system("cls");

}

else

{

if (player.gold < tab1[2].val)

{

cout << "Nie posiadasz wystarczajacej ilosci zlota" << endl;

Sleep(3000);

system("cls");

}

else

{

tab1[2].tobackpack(player);

cout << "Kupiles " << tab1[2].name << endl;

player.gold -= tab1[2].val;

cout << "-" << tab1[2].val << " zlota" << endl;

}

Sleep(3000);

system("cls");

}

}

else

{

//Sleep(3000);

system("cls");

}

break;

default:

exit = 1;

system("cls");

break;

}

}

}

//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

void home(Player & player)

{

bool exit = 0;

while (exit == 0)

{

cout << endl;

cout << "-----DOM-----" << endl;

cout << "0 - aby odpoczac" << endl;

cout << "1 - aby wyswietlic informacje o graczu" << endl;

cout << "<dowolny klawisz> - aby wyjsc z domu" << endl;

char znk = \_getch();

char zn = 'a';

switch (znk)

{

case '0':

system("cls");

player.sleeep(tab2);

cout << "Jestes wypoczety! Twoja energia: 100/100" << endl;

Sleep(3000);

system("cls");

break;

case'1':

while (zn != 'e')

{

system("cls");

player.introduce();

player.myArtifacts();

player.myPotions();

cout << endl;

cout << "e - aby wrocic do menu domu" << endl;

zn = \_getch();

}

system("cls");

break;

default:

exit = 1;

system("cls");

break;

}

}

}

Enemy.cpp

#include "stdafx.h"

#include "Enemy.h"

#include <iostream>

#include <string>

#include <cstdlib>

#include "Player.h"

#include <fstream>

#include <time.h>

#include <conio.h>

#include <windows.h>

using namespace std;

//Enemy::Enemy() {};

Enemy tab2[];

Enemy::Enemy(string nazwa, int zycie, int zloto, int sila, int x, int y)

{

name = nazwa;

hp = zycie;

gold = zloto;

strength = sila;

eposition.x = x;

eposition.y = y;

}

Enemy::~Enemy()

{

}

void Enemy::steal(Player & player)

{

for (int i = 0; i < 6; i++)////////////warunkiem liczba wszystkich wrogow

{

if ((\*this).name == tab2[i].name)

{

if (player.gold < 10)

{

cout << "Nie masz wystarczajaaco duzo zlota aby sie pojedynkowac" << endl;

}

else

{

if (player.energy < 30)

{

cout << endl;

cout << "Nie mozesz pojedynkowac sie z tym przeciwnikiem, jestes zbyt zmeczony. Energia: " << player.energy << "/100" << endl;

cout << "Wroc do domu, aby odpoczac." << endl;

}

else

{

if (tab2[i].gold == 0)

{

cout << endl;

cout << "Przeciwnik jest splukany, nie chce moze z Toba walczyc." << endl;

}

else

{ //////////walka

int rounds;

int liczba1;

int liczba2;

int wynik;

int mojwynik;

int won = 0;

cout << endl;

cout << "-----WALKA-----" << endl;

cout << endl;

if (tab2[i].strength - player.strength <= 0)

{

rounds = 1;

}

else

{

rounds = tab2[i].strength - player.strength;

}

cout << "Ilosc rund: " << rounds << endl;

cout << "Podawaj wyniki!" << endl;

cout << "Zatwierdzaj ENTERem" << endl;

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

{

srand(time(NULL));

liczba1 = rand() % 10 + 1;

liczba2 = rand() % 5 + 1;

wynik = liczba1 \* liczba2;

cout << endl;

cout << liczba1 << " x " << liczba2 << " = ";

cin >> mojwynik;

if (wynik == mojwynik)

{

won++;

}

}

if (won == rounds)

{

cout << endl;

cout << "Pokonales " << tab2[i].name << ", otrzymujesz wygrana w wysokosci: " << tab2[i].gold << " zlota" << endl;

player.gold += tab2[i].gold;

tab2[i].gold = 0;

player.energy -= 30;

cout << "Twoja energia: " << player.energy << "/100" << endl;

}

else

{

cout << endl;

cout << "Zostales pokonany. Tracisz 10 zlota" << endl;

player.gold -= 10;

player.energy -= 30;

cout << "Twoja energia: " << player.energy << "/100" << endl;

Sleep(5000);

}

}

}

}

}

}

}

Enemy.h

#pragma once

#include <iostream>

#include <string>

#include <cstdlib>

//#include "Player.h //napotkany problem: w plikach nagłówkowych powstał "cykl"(a nie powinien) - rozwiązane

#include <fstream>

using namespace std;

class Player;

class Enemy

{

public:

struct Position

{

int x;

int y;

};

Position eposition;

protected:

int hp, gold, strength;

string name;

public:

Enemy() {};

Enemy(string, int, int, int, int, int);

~Enemy();

void steal(Player & player);

friend class Player;

friend void home(Player & player);

friend void create\_enemy(ifstream & file, Enemy & enem);

};

Player.cpp

#include "stdafx.h"

#include "Player.h"

#include <iostream>

#include <string>

#include <cstdlib>

#include "Enemy.h"

#include <fstream>

#include <vector>

#include <conio.h>

#include <ctime>

using namespace std;

Artifact tab[];

Potion tab1[];

Enemy tab2[];

Player::Player(string nazwa, int maxzycie, int zycie, int zloto, int sila, int x, int y, int pojemnosc, int pasek)

{

name = nazwa;

hp = zycie;

gold = zloto;

strength = sila;

eposition.x = x;

eposition.y = y;

maxhp = maxzycie;

bspace = pojemnosc;

beltspace = pasek;

}

Player::~Player()

{

}

int Player::sumstrenght()

{

return strength + strengthfromitems;

}

int Player::sumbspace()

{

return bspace += bspacefromitems;

}

int Player::sumbeltspace()

{

return beltspace += beltspacefromitems;

}

int Player::summaxhp()

{

return maxhp += maxhpfromitems;

}

void Player::introduce()

{

cout << endl;

cout << "-----STSTYSTYKI GRACZA-----" << endl;

cout << "Moje zloto: " << gold << endl;

cout << "Zycie: " << hp << "/" << summaxhp() << endl;

cout << "Energia: " << energy << "/100" << endl;

cout << "Moja sila: " << sumstrenght() << endl;

cout << "Wolne miejsce w plecaku: " << sumbspace() << endl;

cout << "Wolne miejsce na pasku: " << sumbeltspace() << endl;

}

void Player::myArtifacts()

{

cout << endl;

cout << "-----MOJE PRZEDMIOTY-----" << endl;

//cout << myitems.size();

int size = myitems.size();////

if (size == 0)

{

cout << endl;

cout << "Nie posiadasz zadnych przedmiotow." << endl;

}

else

{

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

{

for (int j = 0; j < 6; j++)// warunkiem będzie ilosc artekfaktow W OgOLE

{

if (myitems[i] == tab[j].item\_id)

{

cout << endl;

cout << "- " << tab[j].name << endl;

cout << " + " << tab[j].addmaxhp << " maksymalnego zycia" << endl;

cout << " + " << tab[j].addstrength << " sily" << endl;

cout << " + " << tab[j].addbspace << " miejsca w ekwipunku" << endl;

cout << " + " << tab[j].addbelt << " miejsca na pasku z miksturami" << endl;

cout << " wartosc: " << tab[j].value << " zlota" << endl;

}

}

}

}

}

void Player::myPotions()

{

cout << endl;

cout << "-----MOJE ELIKSIRY-----" << endl;

int size = potions.size();

if (size == 0)

{

cout << endl;

cout << "Nie posiadasz zadnych eliksirow." << endl;

}

else

{

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

{

for (int j = 0; j < 5; j++)// warunkiem będzie ilosc potionow W OgOLE

{

if (mypotions[i] == tab1[j].potion\_id)

{

cout << endl;

cout << "- " << tab1[j].name << endl;

cout << " +" << tab1[j].addzy << " punktow zycia" << endl;

cout << " wartosc: " << tab1[j].val << " zlota" << endl;

}

}

}

}

}

int Player::wincondition()

{

if (Player::gold >= 200)

{

return 1;

}

else

{

return 0;

}

}

void Player::drinkPotion()

{

if (potions.size() > 0)

{

//Potion a, b, c, d, e, f, g, h, i, j;

Potion p;

cout << endl;

cout << "-----MOJE POTIONY-----" << endl;

for (int i = 0; i < potions.size(); i++)

{

p = \*potions[i];

cout << i << " - wypij " << p.name << endl;

}

cout << "<dowlony klawisz> - aby nie pic" << endl;

unsigned char z = \_getch();

if (z == '0')

{

(\*potions[0]).interact(\*(this));

potions.erase(potions.begin() + (int)(z - 48));

mypotions.erase(mypotions.begin() + (int)(z - 48));

}

else if (z == '1')

{

if (potions.size() >1)

{

(\*potions[1]).interact(\*(this));

potions.erase(potions.begin() + (int)(z - 48));

mypotions.erase(mypotions.begin() + (int)(z - 48));

}

}

else if (z == '2')

{

if (potions.size() > 2)

{

(\*potions[2]).interact(\*(this));

potions.erase(potions.begin() + (int)(z - 48));

mypotions.erase(mypotions.begin() + (int)(z - 48));

}

}

else if (z == '3')

{

if (potions.size() > 3)

{

(\*potions[3]).interact(\*(this));

potions.erase(potions.begin() + (int)(z - 48));

mypotions.erase(mypotions.begin() + (int)(z - 48));

}

}

else if (z == '4')

{

if (potions.size() > 4)

{

(\*potions[4]).interact(\*(this));

potions.erase(potions.begin() + (int)(z - 48));

mypotions.erase(mypotions.begin() + (int)(z - 48));

}

}

else if (z == '5')

{

if (potions.size() > 5)

{

(\*potions[5]).interact(\*(this));

potions.erase(potions.begin() + (int)(z - 48));

mypotions.erase(mypotions.begin() + (int)(z - 48));

}

}

else if (z == '6')

{

if (potions.size() > 6)

{

(\*potions[6]).interact(\*(this));

potions.erase(potions.begin() + (int)(z - 48));

mypotions.erase(mypotions.begin() + (int)(z - 48));

}

}

else if (z == '7')

{

if (potions.size() > 7)

{

(\*potions[7]).interact(\*(this));

potions.erase(potions.begin() + (int)(z - 48));

mypotions.erase(mypotions.begin() + (int)(z - 48));

}

}

else if (z == '8')

{

if (potions.size() > 8)

{

(\*potions[8]).interact(\*(this));

potions.erase(potions.begin() + (int)(z - 48));

mypotions.erase(mypotions.begin() + (int)(z - 48));

}

}

else if (z == '9')

{

if (potions.size() > 9)

{

(\*potions[9]).interact(\*(this));

potions.erase(potions.begin() + (int)(z - 48));

mypotions.erase(mypotions.begin() + (int)(z - 48));

}

}

}

else

{

cout << endl;

cout << "Nie masz zadnych potionow do wypicia" << endl;

}

}

void Player::sleeep(Enemy tab2[])

{

energy = 100;

for (int i = 0; i < 6; i++)//warunkiem jest ilosc wrogow w OGOLE

{

tab2[i].gold += 10;

}

cout << tab2[0].gold << endl;

}

bool Player::czyzyje()

{

if (hp <= 0)

{

return 0;

}

else

{

return 1;

}

}

int Player::calkowitygold()

{

int goldfromitems = 0;

int goldfrompotions = 0;

for (int i = 0; i < items.size(); i++)

{

goldfromitems += (\*items[i]).value;

}

for (int j = 0; j < items.size(); j++)

{

goldfrompotions += (\*potions[j]).val;

}

return gold + goldfromitems + goldfrompotions;

}

Player.h

#pragma once

#include <iostream>

#include <string>

#include <cstdlib>

#include "Enemy.h"

#include <fstream>

#include "Artifact.h"

#include "Item.h"

#include <vector>

#include <ctime>

#include "Potion.h"

using namespace std;

//Artifact tab[1];/////////////////lista artefaktow W OGOLE

class Enemy;

class Player : public Enemy

{

protected:

int maxhp;

int strengthfromitems = 0;

int maxhpfromitems = 0;

int bspacefromitems = 0;

int beltspacefromitems = 0;

int energy = 100;

public:

int bspace;

int beltspace;

int artsold = 0;//ilosc sprzedane przedmioty

int potsold = 0;//potiony

char prepos = 'o';

vector <Artifact\*> items;// wektor adresów

vector <int> myitems;// wektor numerow id przedmiotów w moim ekwipunku

vector <Potion\*> potions;

vector <int> mypotions;

Player(string, int, int, int, int, int, int, int, int);//nazwa maxzycie zycie zloto sila x y bspace pasek

~Player();

int sumstrenght();

int sumbspace();

int sumbeltspace();

int summaxhp();

void introduce();

void myArtifacts();

void myPotions();

void drinkPotion();

void sleeep(Enemy tab2[]);

int wincondition();

bool czyzyje();

int calkowitygold();

friend class Artifact;

friend class Potion;

friend class Enemy;

friend void sklep(Player & player);

friend void home(Player & player);

};

Item.cpp

#include "stdafx.h"

#include "Item.h"

#include <iostream>

#include <string>

#include <cstdlib>

#include "Player.h"

#include <vector>

#include <ctime>

using namespace std;

Item.h

#pragma once

#include <iostream>

#include <string>

#include <cstdlib>

#include <vector>

#include <ctime>

using namespace std;

class Player;

class Item

{

public:

virtual void interact(Player & player) = 0;

virtual void tobackpack(Player & player) = 0;

virtual void sellitem(Player & player) = 0;

};

Potion.cpp

#include "stdafx.h"

#include "Potion.h"

#include <iostream>

#include <string>

#include <cstdlib>

#include "Item.h"

#include <vector>

#include <ctime>

#include "Player.h"

using namespace std;

Potion::Potion() {};

Potion tab1[];

Potion::Potion(string nazwa, int dodzdrowie, int wart, int wx, int wy, int id)

{

name = nazwa;

addzy = dodzdrowie;

val = wart;

potionposition.x = wx;

potionposition.y = wy;

potion\_id = id;

}

Potion::~Potion()

{

}

int Potion::checkorder(Player &player)

{

for (int i = 0; i < player.potions.size(); i++)

{

for (int j = 0; j <4; j++)///////////////////warunkiem jest ilosc wszystkich przedmiotow W OGOLE

{

if ((\*player.potions[i]).potion\_id == tab1[j].potion\_id)

{

return i;

}

}

}

}

void Potion::interact(Player & player)

{

if (player.hp + addzy > player.maxhp)

{

player.hp = player.maxhp;

player.beltspace += 1;

cout << endl;

cout << "Wypiles miksture: " << name << endl;

}

else

{

player.hp += addzy;

player.beltspace += 1;

cout << endl;

cout << "Wypiles miksture: " << name << endl;

}

order = -1;

for (int i = 0; i < player.potions.size(); i++)

{

(\*player.potions[i]).order = checkorder(player);

}

}

void Potion::tobackpack(Player & player)

{

cout << endl;

player.potions.push\_back(this);

player.mypotions.push\_back(potion\_id);

cout << "Na pasek dodano przedmiot: " << name << endl;

potionposition.x = -8; // na nasz pasek

potionposition.y = -8; // na nasz pasek

player.beltspace -= 1;

order = player.potions.size() - 1;

}

void Potion::sellitem(Player & player)

{

checkorder(player);

cout << endl;

player.potions.erase(player.potions.begin() + order);

player.mypotions.erase(player.mypotions.begin() + order);

cout << "Sprzedano za " << val - 5 << " zlota" << endl;

potionposition.x = -1; // do naszego ekwipunku

potionposition.y = -1; // do naszego ekwipunku

//inbp = 0;

//inshop = 1;

player.beltspace += 1;

player.potsold++;

player.gold += (val - 5);

order = -1;

for (int i = 0; i < player.potions.size(); i++)

{

(\*player.potions[i]).order = checkorder(player);

}

}

Potion.h

#pragma once

#include <iostream>

#include <string>

#include <cstdlib>

#include <vector>

#include <ctime>

#include "Item.h"

using namespace std;

class Item;

class Potion : public Item

{

protected:

string name;

int addzy;

int val;

public:

int potion\_id;

int order;

struct Position

{

int x;

int y;

};

Position potionposition;

Potion();

Potion(string, int, int, int, int, int);

~Potion();

int checkorder(Player &player);

virtual void interact(Player & player);

virtual void tobackpack(Player & player);

virtual void sellitem(Player & player);

friend class Player;

friend void sklep(Player & player);

friend void create\_potion(ifstream &file, Potion & pot);

};

Artifact.cpp

#include "stdafx.h"

#include "Artifact.h"

#include <iostream>

#include <string>

#include <cstdlib>

#include "Item.h"

#include <vector>

#include <ctime>

#include "Player.h"

using namespace std;

Artifact::Artifact() {};

Artifact tab[];

Artifact::Artifact(string nazwa, int dodhp, int dodsile, int dodspace, int dodpasek, int wartosc, int posx, int posy, int id, bool czywsklepie, bool czywbp)

{

name = nazwa;

addmaxhp = dodhp;

addstrength = dodsile;

addbspace = dodspace;

addbelt = dodpasek;

value = wartosc;

artifactposition.x = posx;

artifactposition.y = posy;

item\_id = id;

inshop = czywsklepie;

inbp = czywbp;

}

Artifact::~Artifact()

{

}

int Artifact::checkorder(Player &player)

{

for (int i = 0; i < player.items.size(); i++)

{

for (int j = 0; j <4; j++)///////////////////warunkiem jest ilosc wszystkich przedmiotow W OGOLE

{

if ((\*player.items[i]).item\_id == tab[j].item\_id)

{

return i;

}

}

}

}

void Artifact::interact(Player & player)

{

}

void Artifact::tobackpack(Player & player)

{

cout << endl;

player.items.push\_back(this);

player.myitems.push\_back(item\_id);

cout << "Do ekwipunku dodano przedmiot: " << name << endl;

artifactposition.x = -9; // do naszego ekwipunku

artifactposition.y = -9; // do naszego ekwipunku

inbp = 1;

inshop = 0;

player.strengthfromitems += addstrength;

player.maxhpfromitems += addmaxhp;

player.bspacefromitems += addbspace;

player.beltspacefromitems += addbelt;

player.bspace -= 1;

order = player.items.size() - 1;

}

void Artifact::sellitem(Player & player)

{

checkorder(player);

cout << endl;

player.items.erase(player.items.begin() + order);//

player.myitems.erase(player.myitems.begin() + order);//

cout << "Sprzedano za " << value - 5 << " zlota" << endl;

artifactposition.x = -1; // do naszego ekwipunku

artifactposition.y = -1; // do naszego ekwipunku

inbp = 0;

inshop = 1;

player.strengthfromitems -= addstrength;

player.maxhpfromitems -= addmaxhp;

player.bspacefromitems -= addbspace;

player.beltspacefromitems -= addbelt;

player.bspace += 1;

player.artsold++;

player.gold += (value - 5);

order = -1;

for (int i = 0; i < player.items.size(); i++)

{

(\*player.items[i]).order = checkorder(player);

}

}

Artifact.h

#pragma once

#include <iostream>

#include <string>

#include <cstdlib>

#include <vector>

#include <ctime>

#include "Item.h"

#include <fstream>

using namespace std;

class Item;

class Artifact : public Item

{

protected:

string name;

int addmaxhp;

int addstrength;

int addbspace;

int addbelt;

int value;

bool inshop;

bool inbp;

public:

int item\_id;

int order;

struct Position

{

int x;

int y;

};

Position artifactposition;

public:

Artifact();

Artifact(string, int, int, int, int, int, int, int, int, bool, bool); // nazwa +hp +strenght +bspace +pasekspace wartosc x y item\_id inshop inbp

~Artifact();

int checkorder(Player &player);

virtual void interact(Player & player);

virtual void tobackpack(Player & player);

virtual void sellitem(Player & player);

friend class Player;

friend void sklep(Player & player);

friend void create\_artifact(ifstream & file, Artifact & ar);

};

Pliki .txt potrzebne do prawidłowego działania programu:

* mapa.txt
* enemies.txt
* potions.txt
* artifacts.txt

mapa.txt

\_\_\_\_\_\_W\_oE

\_\_P\_\_ooIo\_

E\_oI\_ooooE

oooo\_o\_\_o\_

\_\_oo\_oPo\_\_

H\_oooooooS

o\_ooEooo\_\_

o\_oo\_ooooo

oooI\_ooooo

oE\_\_\_E\_ooX

enemies.txt

6

Sesja

200

1

6

0

2

Kuba

100

20

1

9

0

Pawel

120

25

2

9

2

Janek

135

18

3

4

6

Szymon

150

21

4

1

9

Alojzy

177

13

5

5

9

potions.txt

5

Health\_Potion\_+\_80

80

40

-2

-2

0

Health\_Potion\_+\_20

20

8

-2

-2

1

Health\_Potion\_+\_60

60

25

-2

-2

2

Health\_Potion\_+\_40

40

15

2

1

3

Healt\_Potion\_+\_100

100

50

6

4

4

artifacts.txt

6

Fidget\_Spinner

0

1

0

0

100

-2

-2

0

1

0

Pencil

20

0

0

0

20

-2

-2

1

1

0

Candle

0

0

1

0

70

-2

-2

2

1

0

Cap\_of\_Strength

0

2

0

0

49

3

2

3

0

0

Old\_Shoe

0

0

1

0

19

7

1

4

0

0

Weird\_Book

0

0

0

1

29

3

8

5

0

0

## Wnioski

Cały program, moim zdaniem działa nie najgorzej. Jako, że jest to gra, w różnych fazach powstawania wymagała testów pod kątem prawidłowego działania zastosowanych mechanizmów. Użytkownik nie ma tu ściśle sprecyzowanego schematu działania. Ta dowolność tworzy przestrzeń do powstawania wielu nieprzewidzianych wcześniej błędów, które wymagają poprawy. Wiele z nich udało mi się skorygować, ale nie jestem pewny, czy to wszystkie.

Powstawanie programu było rozłożone w czasie, więc struktura kodu może wydawać się miejscami mało uporządkowana. W trakcie powstawały fragmenty kodu - elementy, które można porównać do klocków LEGO. Na samym końcu musiałem je wszystkie połączyć. Niestety nie odbyło się bez komplikacji. Niektóre fragmenty kodu wymagały całkowitej zmiany, aby dopasować się do całości. Dzięki temu nauczyłem się, że programy tego typu wymagają bardzo precyzyjnego planowania.

Jest to mój pierwszy projekt pisany obiektowo w c++ o takiej objętości. Starałem się w nim zawrzeć wiedzę wyciągniętą z laboratoriów, oraz wykładów.

Bibliografia:

Strona, dzięki której powstał ekran startowy:

http://patorjk.com/software/taag/#p=display&f=AMC%20Tubes&t=Save%20%20%20your%0A%20%20Brother