

REV	DATA	Autor
0.1	20.12.2021	<i>Wojciech Nodzyński (nodzynsk@student.agh.edu.pl)</i>
0.2	14.01.2022	<i>Wojciech Nodzyński (nodzynsk@student.agh.edu.pl)</i>

# DUNGEON GAME – GRA TYPU DUNGEON

Autor: Wojciech Nodzyński  
Akademia Górniczo-Hutnicza

## Spis treści

1.	WSTĘP.....	3
2.	WYMAGANIA SYSTEMOWE ( <i>REQUIREMENTS</i> ).....	4
3.	FUNKCJONALNOŚĆ ( <i>FUNCTIONALITY</i> ).....	5
4.	PROJEKT TECHNICZNY ( <i>TECHNICAL DESIGN</i> ) .....	6
5.	OPIS REALIZACJI ( <i>IMPLEMENTATION REPORT</i> ).....	8
6.	OPIS WYKONANYCH TESTÓW ( <i>TESTING REPORT</i> ) - LISTA BUGGÓW, UZUPEŁNIEN, ITD.	9
7.	PODRĘCZNIK UŻYTKOWNIKA ( <i>USER'S MANUAL</i> ).....	10

# 1. Wstęp

Dokument przedstawia wykonanie i działanie prostej gry komputerowej typu dungeon. Celem projektu było utworzenie programu w języku obiektowym. W projekcie został wykorzystany język C++.

## **2. Wymagania systemowe (*requirements*)**

Podstawowe założenia projektu:

1. Stworzenie prostej wersji modułów
2. Scalenie modułów i logiki
3. Wprowadzenie elementów poruszania się

### **3. Funkcjonalność (functionality)**

Główną funkcją programu jest symulacja gry dungeon. Użytkownik wybiera bohatera, którym będzie grał i używając klawiatury przemieszcza się po mapie. Podczas każdego takiego ruchu, zostaje wylosowane zdarzenie

## 4. Projekt techniczny (*technical design*)

```
class interact
{
public:
    //constructor doesn't do anything
    interact();

    void static Draw();

    void static Move();
};
```

```
class Engine{
public:
    //Constructor doesn't do anything
    Engine();

    //Set Hero and Monster
    void static Choose_Hero();
    void static Choose_Monster();

    //Increases hero's stats
    void static Shop();

    //Uses Choose_Monster and then 2 objects fight until
    //program detects negative value of 'hp' in any object
    void static Fight();

    //Draw lots events. Uses previous methods aswell
    void static Random_Event();

    //Return false if 'Hero hp' is lower or equal 0
    bool static GameOver();

    //Add const value to object's 'hp' every time room is passed by
    void static Recover();
};
```



```
class Mob{
private:
    int hp;
    int dmg;
    int hit_rat;
    int def;
public:
    //Constructor
    Mob(int hp_, int dmg_, int hit_rat_, int def_);

    //Setters
    void Set_hp(int hp);
    void Set_dmg(int dmg);
    void Set_hit_rat(int hit_rat);
    void Set_def(int def);

    //Getters
    int Get_hp();
    int Get_dmg();
    int Get_hit_rat();
    int Get_def();
};
```

## 4.1 Klasy

Struktura programu oddaje idee programowania obiektowego, gdyż zawiera zaprojektowane i zaimplementowane klasy.

**Klasa Mob** zawiera w sobie zmienne i metody, które odpowiadają za tworzenie obiektów. Publiczne funkcje to Settery i Gettery zmiennych klasy, oraz konstruktor, do tworzenia obiektu. Zmienne są typu prywatnego.

**Klasa interact** zawiera funkcje, które odpowiadają za rysowanie prymitywnej mapy i poruszanie się wewnątrz niej. Funkcje są statyczne, gdyż klasa nie posiada zmiennych.

**Klasa Engine** zawiera funkcje, które wykonują główną funkcjonalność programu. Funkcje są statyczne, ponieważ nie ma zmiennych.

## **5. Opis realizacji (implementation report)**

Program został wykonany w Microsoft Visual Studio 2019 (wersja 16.11.5).

Działanie zostało sprawdzone na urządzeniach:

- Intel(R) Core(TM) i5-4460, Windows 10 Education wersja 20H2
- Intel(R) Core(TM) i7-8850U, Windows 10 Home wersja 20H2



## 6. Opis wykonanych testów (*testing report*) - lista buggów, uzupełnień, itd.

Data	Autor	Opis	Stan
20.12.2021	Wojciech Nodzyński	Podczas wyboru bohatera, wpisując inny typ danych niż „int”, program wybiera pierwszego bohatera.	W momencie wpisania z klawiatury, typ zmiennej zostaje sprawdzony
20.12.2021	Wojciech Nodzyński	Podczas walki, w przypadku gdy zmienna licząca wartość obrażeń jest ujemna, zmienna „hp” jest zwiększana	Sprawdzanie wartości zmiennej, w przypadku wartości ujemnej, nic się nie dzieje.
20.12.2021	Wojciech Nodzyński	Podczas walki, gdy jeden z obiektów, powinien „zginąć”, wykonywał on jeszcze jedną akcję, co czasem prowadziło do końca gry.	Sprawdzanie stanu zmiennej „hp” dla obu obiektów, przed ich „ruchem”
14.01.2022	Wojciech Nodzyński	Funkcja Recover() jest w stanie zwiększyć zmienną „hp” ponad stan początkowy.	Nierozwiązane, magia lochów.

## 7. Podręcznik użytkownika (*user's manual*)

Przemieszczanie się w programie i samo granie polega głównie na podejmowaniu decyzji, co wpisać do konsoli. Przy każdym takim wyborze, użytkownik ma podane możliwości oraz legendy. Niżej kilka przykładów

Ekran wyboru postaci, pierwszy wybór

```
Choose your hero
0 - Artur, famous knight of Silvar
1 - Rossalia, mysterious bounty hunter
2 - Dante, outsider
3 - Mirael, priestess of blue flame order
4 - Create your own character
```

Ekran wyboru kierunku, gdzie się udamy

```
Which way will you choose?
u - up, d - down, r - right, l - left
```

Ekran podjęcia decyzji w losowym wydarzeniu.

```
You feel thirsty...
Drink water?
y - yes, n - no
```