

Poszerzony opis projektu

1. Opis gry

Po rozpoczęciu gry generowany jest automatycznie przeciwnik. Obok niego zostają wyświetlane informacje o nim (punkty zdrowia, siła itp.). Gracz może wybrać czy chce z nim walczyć czy wybrać innego przeciwnika. Po wybraniu opcji walki z potworem, wybiera się postać, która ma pokonać wroga (np. czarodziej, wojownik itp.). Obok postaci wyświetlany jest jej opis. Wybór postaci to ważny element od którego może zależeć wynik rozgrywki, jako że pierwszy ruch wykonuje przeciwnik.

2. Funkcje i interfejs

'Start Game' – opcja rozpoczynająca grę.

'Options' – opcja w której można zmienić dźwięk i zobaczyć wyniki poprzednich rozgrywek.

'Generate another enemy' – wybranie innego wroga.

'Fight' – podjęcie walki z przeciwnikiem.

'Choose character' – wybór postaci.

'Exit' – zamyka grę.



3. Klasy:

W projekcie znajdują się trzy klasy główne:

- klasa GameView
- klasa GameProgress
- klasa Character

Klasa GameProgress – klasa w której znajdują się funkcje pobierające dane z plików wejściowych. Przechowuje informacje o rozgrywkach i zapisuje do pliku końcowego informacje o tym czy dana rozgrywka została zakończona zwycięstwem lub porażką.

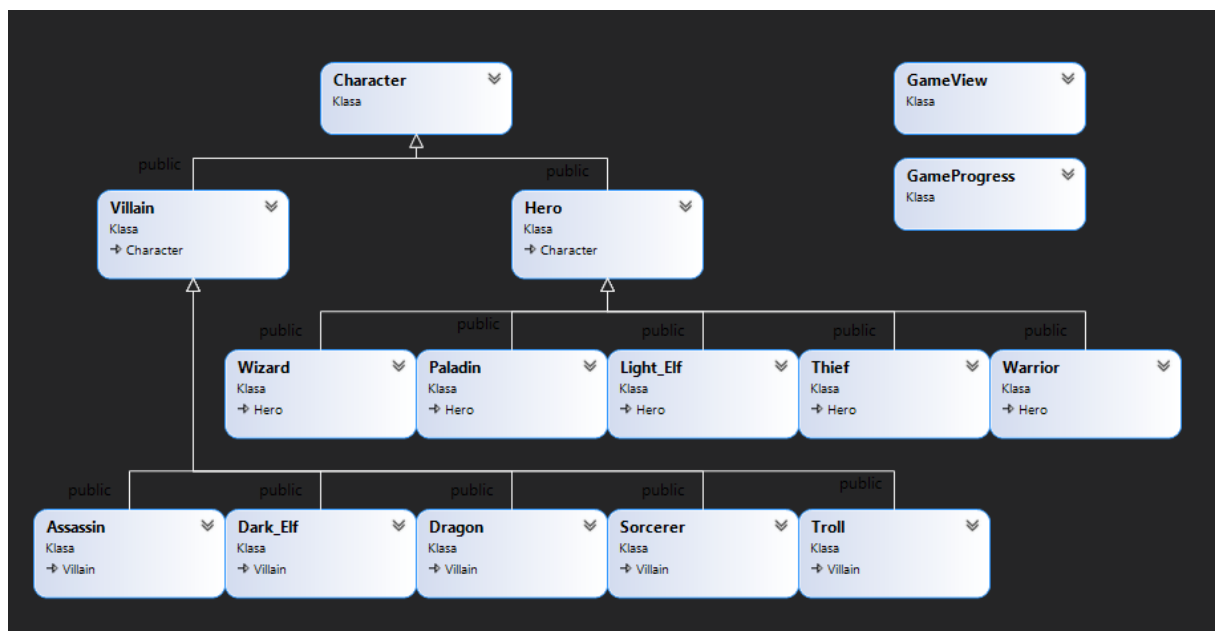
Klasa GameView – klasa odpowiedzialna za interfejs, wykorzystująca bibliotekę SFML.

Klasa Character – klasa bazowa dla klasy Hero i Villain.

Klasa Hero – klasa na podstawie której powstają klasy poszczególnych bohaterów (Warrior, Paladin, Light_Elf itp.)

Klasa Villain – klasa główna dla klas przeciwników (Dragon, Socerer, Dark_Elf).

Diagram klas:



4. Struktury danych

W programie jako główna struktura danych zostanie wykorzystany vector. Aby zapisać dane do pliku o wyniku starcia zostanie wykorzystana lista.

5. Techniki obiektowe

W programie zostaną wykorzystane :

- Mechanizm wyjątków
- Kontenery STL
- Algorytmy i iteratory STL
- Wzorce

6. Biblioteki zewnętrzne

Projekt zostanie wykonany w oparciu o bibliotekę graficzną SFML.