

# Studenckie RPG - Dokumentacja Projektu

Antoni Przybylik

PROI 23L

GRUPA 103

Zoja Hordyńska

PROI 23L

GRUPA 103

9 kwietnia 2023

## Cel i opis projektu

Przedmiotem projektu jest „Studenckie RPG” - gra RPG inspirowana grą Diablo. Gracz (student) przechodzi przez mapę Politechniki, zbierając przedmioty i walcząc z potworami godnymi studiów. Każdy gracz posiada pewne umiejętności, zdrowie i manę. Podczas walki, gracz rzuca umiejętności, które zadają losowe obrażenia z danego przedziału (obrażenia są zwiększane przez przedmioty oraz inne umiejętności). Celem gry jest zabicie wszystkich bossów i tym samym przeżycie studiów.

Projekt ma zostać zrealizowany w języku C++ z użyciem jego zaawansowanych mechanizmów i ze szczególnym uwzględnieniem paradygmatu obiektowego. Planowane jest stworzenie dedykowanego silnika gry zaimplementowanego przy użyciu biblioteki SFML, który będzie warstwą abstrakcji między wysokopoziomowym kodem właściwej gry operującym na „duszkach”, a interfejsem biblioteki SFML pozwalającym na proste operacje takie jak wyświetlanie obrazów.

## Architektura

Projekt jest podzielony na cztery moduły:

- Silnik gry
- Ciało gry
- Launcher
- Aplikacja gry

Silnik gry dostarcza interfejs zaprojektowany w paradygmacie obiektowym. Jest on oparty o model „Sprite’owy”. Użytkownik tworzy „duszki” (ang. sprite) i określa ich interakcje z otoczeniem. Następnie dodaje duszki do silnika i uruchamia silnik. Silnik symuluje zachowanie duszków podane przez użytkownika. Programowanie obiektowe pozwala na przetwarzanie klasy Sprite (duszek) i umieszczenie w nim kodu który zostanie wykonany na odpowiednich zdarzeniach takich jak zderzenie dwóch duszków lub kodu który jest wykonywany w każdym tyknięciu zegara.

```
1      /* Utworzenie duszka. */
2      skin = new SpriteSkin(*tiles, 3, 500, 1);
3      sprite = new Sprite(skin,
4                          Rect(0, 0, win_x, win_y),
5                          BA_BOUNCE);
6      sprite->set_position(Rect(100, 100, 75, 75));
7      sprite->set_velocity(Vector(120, 250));
8
9      /* Silnik. */
10     engine = new Engine();
11     engine->bind_window(window);
12     engine->add_sprite(sprite);
```

13  
14

```
engine->exec();
```

### Listing 1: Interfejs silnika - kod poglądowy

Ciało gry to kod implementujący właściwą rozgrywkę. To w nim znajdą się takie klasy jak: Player, Enemy, Spell, czy Item.

- Klasa Player przechowywać będzie wszystkie, potrzebne do gry, informacje odnośnie gracza: ilość punktów zdrowia, manę, moc, położenie i ekwipunek. Będzie umożliwiała działania na ekwipunku oraz walkę z przeciwnikami.
- Klasa Enemy będzie określać różnego rodzaju potwory oraz umożliwiać walkę z przeciwnikiem.
- Klasa Spell charakteryzować będzie zaklęcia - ich moc ataku oraz efekty specjalne. Zaklęcia są inne dla gracza, a inne dla każdego rodzaju potwora.
- Item to klasa dotycząca przedmiotów zbieranych w grze, każdy przedmiot dodawać będzie graczowi określoną ilość zdrowia i mocy, która przekładać się będzie na zwiększoną moc ataku.

Launcher jest programem służącym do uruchomienia gry. Można w nim modyfikować ustawienia rozgrywki - zmienić teren, wybrać rozgrywkę bez mniej znaczących przeciwników, a także wczytać stan gry z pliku. <TODO>

Aplikacja gry jest programem który służy do grania w grę. W oknie aplikacji gry jest renderowana symulacja gry. Aplikacja gry powinna dodatkowo pozwalać na zatrzymanie gry, zmianę ustawień i wyjście z gry. <TODO: Napisz coś o aplikacji gry.>

## Wymagania (ja bym póki co tego nie dawala)

Gra wymaga zainstalowanej biblioteki SFML. Założone jest działanie gry na systemach operacyjnych Windows i Linux. Opcjonalnie gra mogłaby wspierać system MacOS. Pewnie Makefile dla Linuksa będzie działał na MacOS, ale nie mam tego systemu i nie zobowiązuje się że gra będzie na nim prawidłowo działać i kompilować bez ostrzeżeń.

<TODO: Napisać więcej.>

## Instrukcja użycia

<TODO>

## Planowane biblioteki i narzędzia

<TODO>

## Podział pracy

- Antoni - <TODO>
- Zoja – menu i interfejs gry, tekstury i obrazki, klasy związane z ciałem gry (Player, Enemy, Spell, Item) <TODO>