

Studenckie RPG - dokumentacja końcowa projektu

Antoni Przybylik

PROI 23L

GRUPA 103

Zoja Hordyńska

PROI 23L

GRUPA 103

27 maja 2023

Opis gry

Studencka gra RPG to wyzwanie dla miłośników przygód, którzy chcą zmierzyć się z największą zmurą inżynierów - Politechniką. Czy uda Ci się przeżyć te studia? Przejdź się korytarzami uczelni wśród gąszczy książek, pustych biurek i płaczących nad kodami studentów, na ścianach widząc istny koszmar programisty – wypisany czerwonymi literami ERROR. Niech nie zgasi Twego zapachu oschłość Pani z dziekanatu, dbałość o szczegóły prowadzącego czy bezwzględność, wobec Twego zdrowia psychicznego, sesji. Legenda miejsca głosi, iż nikt jeszcze nie przeżył Politechniki bez uszczerbku na zdrowiu. Czy będziesz pierwszym, który dokona niemożliwego?

Instrukcja obsługi

Po uruchomieniu programu, na ekranie ukaze się menu startowe. Aby wybrać jedną z dostępnych opcji należy użyć strzałek (górnej lub dolnej), a następnie wcisnąć klawisz ENTER. "Play" rozpocznie grę, "Options" da możliwość zapisania stanu gry, a także wznowienie wcześniej zapisanej gry, "Exit" za to spowoduje zamknięcie okna menu.

Po wciśnięciu "Play" i pojawieniu się postaci na mapie, możesz swobodnie się po niej poruszać, używając, intuicyjnie działających, strzałek. Po spotkaniu na swej drodze jednego z trzech bossów, kliknij na niego, aby rozpocząć quiz. Wybierz odpowiedź za pomocą strzałek, a następnie zatwierdź swój wybór klawiszem ENTER. Wciśnij spację, aby przejść do kolejnego pytania. Po odpowiedzeniu na wszystkie pytania (bądź utracie wszystkich punktów zdrowia) zamknij okno za pomocą klawisza ESC. Nie tracąc wszystkich punktów zdrowia, odpowiedz na wszystkie pytania i przeżyj studia!

Zmiany względem wcześniejszych założeń

<TODO>

Architektura

Projekt jest podzielony na cztery moduły:

- Silnik gry
- Ciało gry
- Launcher
- Launcher pytań
- Aplikacja gry

Silnik

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. Polimorfizm w języku C++ pozwala na nadpisywanie w klasach dziedziczących po klasie Sprite (duszek) funkcji wirtualnych i dzięki temu można w duszkach umieszczać kod który zostanie wykonany na odpowiednich zdarzeniach takich jak zderzenie dwóch duszków lub kodu, który jest wykonywany w każdym tyknięciu zegara.

Ciało Gry

Ciało gry to kod implementujący właściwą rozgrywkę.

Klasa Player przechowuje liczbę punktów zdrowia gracza.

Klasa Enemy jest klasą, opisującą wrogów. Każdy wróg posiada wektor indywidualnych pytań oraz punktów zdrowia, które zabierają graczowi za niepoprawną odpowiedź. Metody klasy pozwalają na zarządzanie zestawem pytań, wyświetlaniem ich treści oraz odpowiedzi, a także modyfikację parametrów gracza.

Klasa Question przechowują informacje dotyczące pytań - ich treść oraz dostępne odpowiedzi.

Launcher

Launcher jest programem służącym do uruchomienia gry. Z jego pomocą rozpoczyna się rozgrywkę, zapisuje stan gry do pliku, a także wczytuje stan wcześniej zapisanej gry.

Launcher pytań

Launcher jest programem odpowiednio zarządzającym, dostępnym w grze, quizem. Wyświetla się tam treść pytań wraz z dostępnymi odpowiedziami. Zła odpowiedź odbiera życie graczowi, a odpowiedzenie na wszystkie pytania przeciwnika sprawia, że znika z mapy.

Aplikacja Gry

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.

Wymagania

Gra wymaga zainstalowanej biblioteki SFML. Założone jest działanie gry na systemach operacyjnych Windows i Linux. Opcjonalnie gra mogłaby wspierać system MacOS.

Biblioteki i Narzędzia

Projekt w całości ma zostać zrealizowany w języku C++, w standardzie C++20. Dopuszczalne jest używanie rozszerzeń GCC.

Skrypty pozwalające na zbudowanie projektu mają być kompatybilne z programem make i do zbudowania projektu ma zostać użyty toolchain GCC. Do kompilacji gry na Windowsa zostanie użyty MinGW.

Projekt ma zostać zrealizowany przy użyciu biblioteki SFML, biblioteki standardowej języka C++ (libstdc++), biblioteki standardowej języka C (libc) i biblioteki matematycznej (libm).

Dodatkowe aplikacje takie jak launcher mogą zostać zrealizowane przy użyciu innych bibliotek np. QT.

W projekcie nie jest wymagane ścisłe trzymanie się kanonicznej wersji tych bibliotek. Dopuszczalne jest użycie rozszerzeń GLIBC.

Podział Pracy

Antoni

- Silnik
- Wybrane części gry

Zoja

- Menu
- Interfejs pytań
- Tekstury i obrazki
- Ciało gry; Klasy: Player, Enemy, Question