Trouble in Lecture Center

GRA DUNGEON EXPLORER

Laboratorium Programowania Obiektowego

Dominik Galewski
106575
dominik.galewski@student.put.edu.pl

Jakub Woźniak
109686
jakub.l.wozniak@student.put.edu.pl

Prowadzący: mgr inż. Mateusz Cicheński
Wydział Informatyki Politechniki Poznańskiej

16 grudnia 2013

1 Wprowadzenie

Trouble in Lecture Center to gra typu *dungeon explorer*, w której wcielamy się w postać studenta informatyki. Odwiedzając kolejne lokacje znajdujące się w mrocznych podziemiach jego uczelni, próbujemy zmierzyć się ze wszystkimi przeciwnościami: analizą matematyczną, programowaniem deklaratywnym, systemami operacyjnymi czy metodami probabilistycznymi. Nasze decyzje wpływają na los studenta i wynik starć z przeciwnikami.

2 Struktura projektu

2.1 Instrukcja kompilacji

Projekt został napisany w języku C++ (standard C++11) przy użyciu biblioteki *SFML* (http://sfml-dev.org) w wersji 2.0. Do kompilacji wymagany jest kompilator g++ w wersji 4.8. Dostarczony plik *Makefile* pozwala na skompilowanie projektu przy pomocy polecenia *make*. Plik wynikowy znajduje się w katalogu *bin*. Program należy uruchomić bedąc w katalogu *bin/*, w przeciwnym razie nie zostaną załadowane pliki z grafiką, itp.

2.2 Podział klas

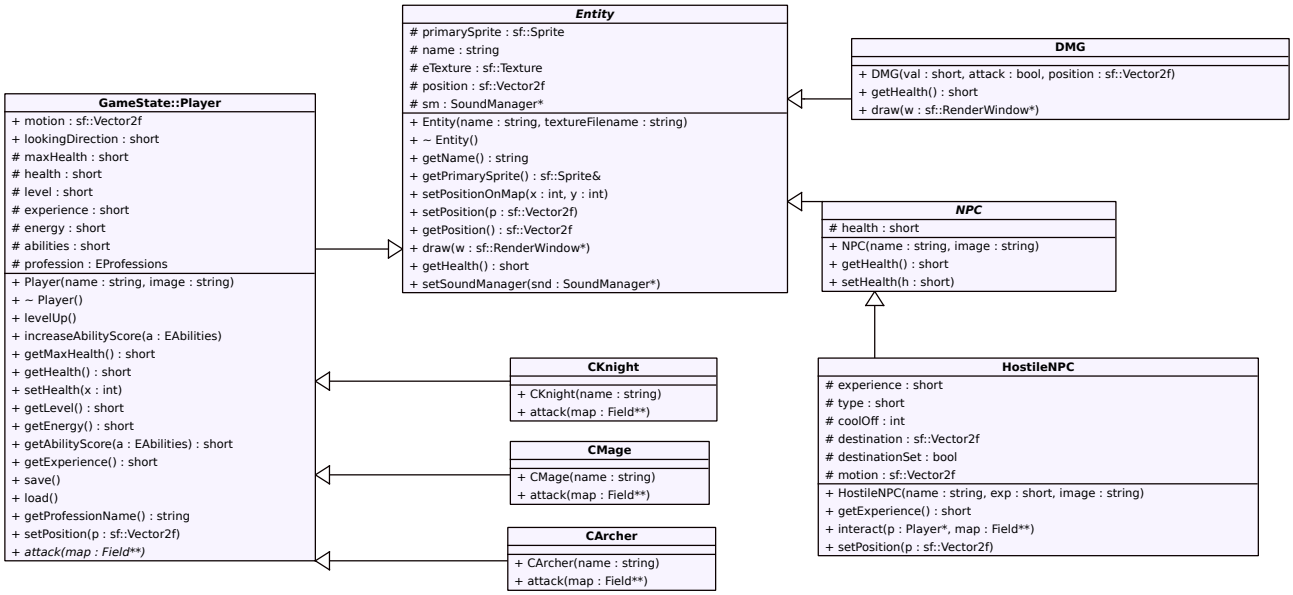
Rysunek 1 przedstawia podział klas związanych z rozgrywką. Główną klasę stanowi *Entity*, która reprezentuje wszystkie elementy w grze, które mogą być narysowane na mapie. Klasa NPC przedstawia postacie sterowane przez komputer, obecnie istnieje tylko jedna klasa pochodna - *HostileNPC*, które reprezentuje postacie wrogie użytkownikowi. Klasa *Player* odpowiada za reprezentowanie postaci gracza w grze, dzieli się na 3 klasy pochodne, które są profesjami postaci. *DMG* — wyświetla zadane obrażenia na mapie.

Rysynek 2 przedstawia zależność klas związanych z logiką gry, głównie reprezentuje klasę *IState* i jej klasy pochodne — odpowiednie stany w których gra może się znajdować (splash, tworzenie postaci, intro, menu, gra właściwa, zakończenie). Dodatkowo, klasa *GameState* zawiera pole typu *Level* będące obiektem reprezentującym aktualnie rozgrywany poziom.

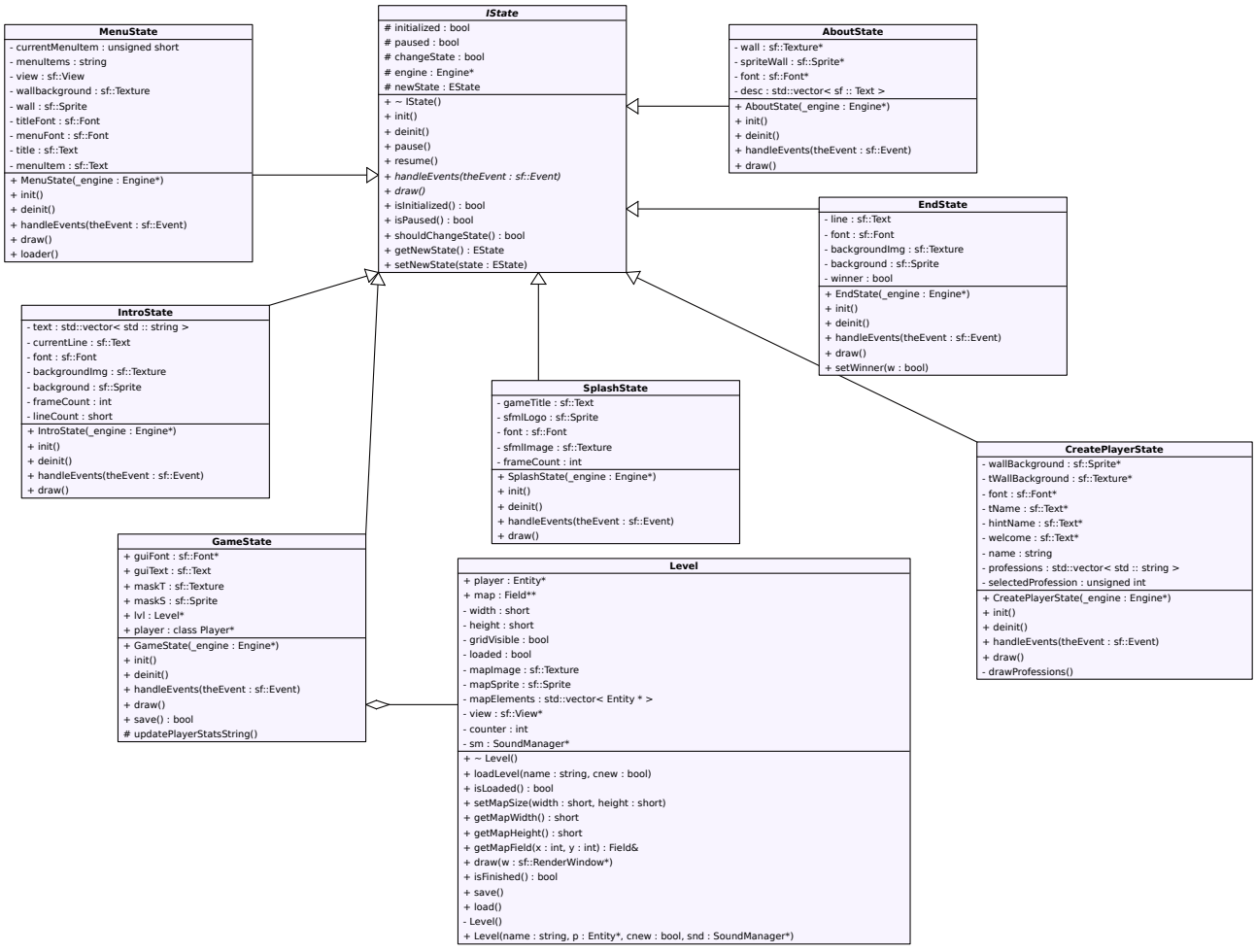
Rysunek 3 przedstawia pozostałe klasy: *Engine* — zawierająca logikę uruchomienia i utrzymania pętli zdarzeń wraz z przekazywaniem kontroli do stanów, *Log* — prosty interfejs do zapisywania logu, *StateManager* — maszyna stanów, odpowiadająca za przechowywanie i zarządzanie obiektami stanów gry, *SoundManager* — klasa zarządzająca dźwiękiem w grze.

2.3 Klawiszologia

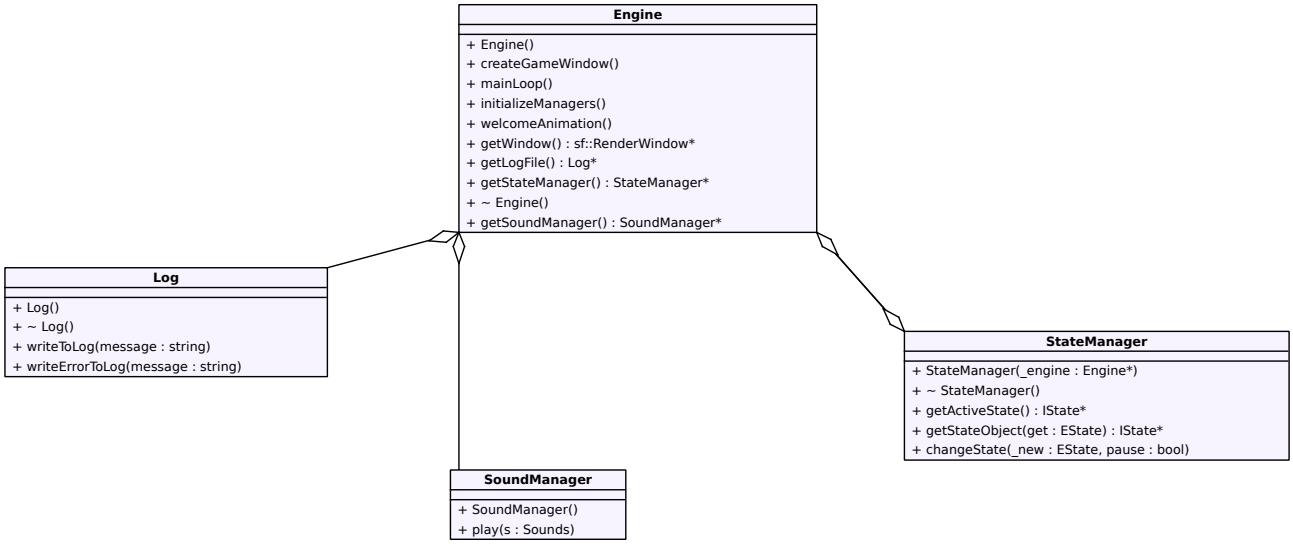
Postać porusza się przy pomocy klawiszy *WSAD*, atakuje przy pomocy *Spacji*. Natychmiastowy powrót do menu gry następuje poprzez wciśnięcie klawisza *Esc*, a zapis gry przy pomocy *F5*.



Rysunek 1: diagram podziału klas związanych z rozgrywką



Rysunek 2: diagram podziału klas związanych z logiką



Rysunek 3: diagram podziału pozostałych klas

3 Spis wymagań

3.1 Graficzny interfejs użytkownika

Interfejs użytkownika został zrealizowany przy pomocy biblioteki *SFML*. Wykorzystane elementy graficzne i dźwiękowe zostały wykonane przez nas lub są na wolnej licencji. Mapa zrealizowana jest w widoku „z lotu ptaka".

3.2 Wpływanie na rozgrywkę

Użytkownik dokonuje wyboru profesji, taktyki i sposobu walki z przeciwnikiem tak, aby pokonać wszystkich wrogów. Musi unikać ciosów, atakować w odpowiednich momentach i uważać na możliwość zablokowania przez wrogie jednostki.

3.3 Zapis stanu gry

Użytkownik w każdym momencie gry ma możliwość zapisu aktualnego stanu celem późniejszego odtworzenia.