

Programowanie obiektowe

Implementacja

Implementacja

8.01.16, 8:30

Michał Leśniak

2015/2016

Dominik Paluch

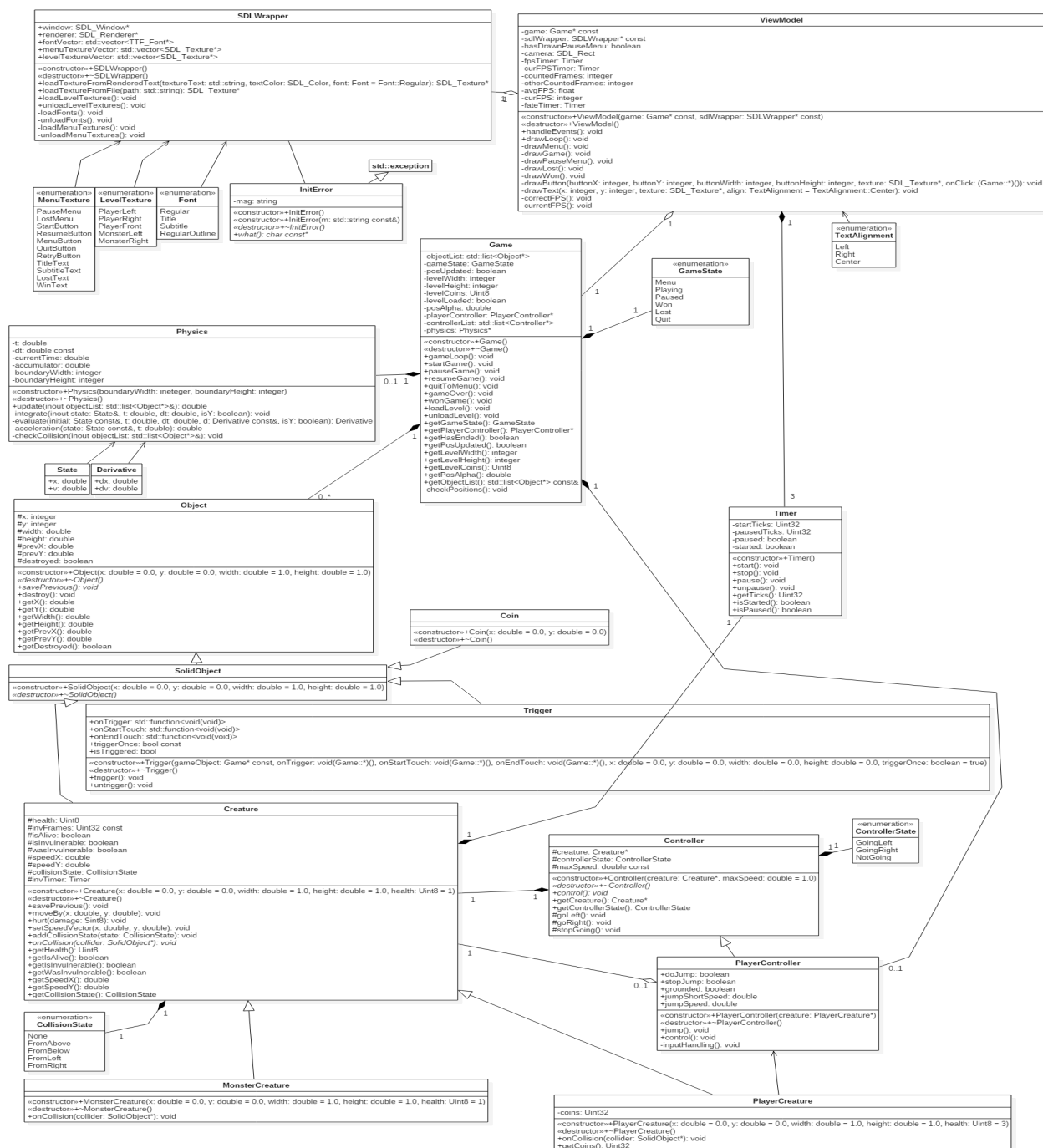
Opis funkcjonalności

Gra platformowa polega na ukończeniu poziomu poruszając się bohaterem pamiętając o omijaniu przeszkód w postaci potworów, przepaści czy ścian oraz zebraniu jak największej ilości monet. Gracz wygrywa jeśli dotrze do końca poziomu. Istnieje możliwość przegrania jeśli gracz straci wszystkie życia, na przykład poprzez obrażenia zadane przez potwora czy spadając w przepaść. Gra kończy się poprzez wybranie odpowiednich instrukcji w menu przez użytkownika.

Zmiany

Zasadniczą zmianą jest zmiana bibliotek z Qt na SDL, gdyż uznaliśmy, że SDL jest optymalniejszy w stosunku do naszych wymagań oraz oczekiwań. Dość dużo zmian zaszło również w implementacji, dodanie wielu innych klas, zmiana już istniejących, oddzielenie grafiki od logiki gry. Zmieniają się również właściwości naszego bohatera, który nie może zniszczyć potwora, a tylko unikać kontaktu z nim, który powoduje obrażenia bohatera. Odseparowaliśmy odpowiedzialność klas na mniejsze fragmenty.

Dokumentacja techniczna - diagram klas



Programowanie obiektowe	Implementacja
Dokumentacja techniczna - wybrane scenariusze działania <ul style="list-style-type: none">Gra jest uruchamiana i zamykana przez użytkownika. Tworzony jest obiekt klasy Game. Jego atrybut hasEnded jest ustawiany przez jego konstruktor na false. Następuje tworzenie obiektów. Wciśnięcie przycisku Esc jest wykrywane przez funkcję odpowiadającą za obsługę wejść. Atrybut hasEnded jest ustawiany na true. To powoduje wywołanie destruktora obiektu klasy Game, który z kolei usuwa wszystkie swoje atrybuty. Aplikacja się wyłącza.Gra jest uruchamiana i użytkownik steruje postacią do końca poziomu. Tworzony jest obiekt klasy Game. Jego atrybut hasEnded jest ustawiany przez jego konstruktor na false. Następuje tworzenie obiektów. Funkcja odpowiadająca za obsługę wejść odpowiednio zmienia atrybuty postaci sterowanej przez gracza, aby ta się poruszała. Gdy postać dochodzi od końca poziomu, atrybut hasEnded jest ustawiany na true. To powoduje wywołanie destruktora obiektu klasy Game, który z kolei usuwa wszystkie swoje atrybuty. Aplikacja się wyłącza.	
Dokumentacja techniczna - inne elementy nie opisane powyżej <p>Wykorzystaliśmy biblioteki SDL2, SDL2_ttf oraz SDL2_image. Do wczytania reprezentacji obiektów graficznych wykorzystaliśmy pliki graficzne z rozszerzeniem png. Do reprezentacji obiektów tekstowych wykorzystaliśmy pliki tekstowe z rozszerzeniem ttf.</p> <p>Wykorzystujemy wskaźniki na metody obiektów w celu komunikacji pomiędzy klasami użyte m. in. w klasach Trigger i ViewModel.</p> <p>W klasie Physics korzystamy z algorytmu Rungego-Kutty'ego rzędu 4.</p> <p>Kolizja wykrywana za pomocą algorytmu SAT (Separating Axis Theorem), która jest uproszczona dzięki zastosowaniu brył brzegowych w postaci prostokątów ułożonych równo z osiami układów współrzędnych (axis aligned bounding box).</p>	
Dokumentacja użytkownika <p>Gracz może odpalić grę, zakończyć grę, zapauzować a także wznowić działanie gry. Może również wygrać jak i przegrać, zdobywać monety oraz tracić życia.</p> <p>Osoba grająca może miło spędzić czas grając w naszą grę. Polecane szczególnie w celach relaksacyjnych po ciężkim dniu pracy.</p>	
Miejsce na uwagi prowadzącego	