

IORacer

Dokumentacja

22.04.2017

Autorzy:

Jacek Łysiak [345639]

Bartosz Burny [370739]

Paweł Zięcik [371915]

Maciej Biernaczyk [370721]

1. Opis problemu

Postawiliśmy sobie zadanie stworzenia prostej i przyjemnej gry wyścigowej. Przyjęliśmy koncepcję dwuwymiarowego świata gry z widokiem z góry. Staramy się możliwie realistycznie oddać fizykę jazdy, odwzorowując zachowanie samochodu wyścigowego na różnych podłożach i w kontakcie z różnymi przedmiotami. Ponadto postanowiliśmy zapewnić modyfikowalność parametrów pojazdu, docelowo poprzez dobór jego części.

2. Rozwiązanie

2.1. Strona funkcjonalna

Gra oferuje możliwość rozegrania wyścigu z widokiem 2D przy pomocy samochodu sterowanego strzałkami, a także możliwość modyfikowania jego parametrów za pomocą warsztatu.

Fizyka ruchu pojazdu uwzględnia wiele parametrów, takich jak: bezwładność, przyczepność kół czy też odmienne zachowanie na różnych nawierzchniach. Dzięki modelowaniu zachowania kół rozgrywka jest dużo bardziej realistyczna niż przy modelach samochodu będących pojedynczym, jednolitym prostokątem.

Warsztat umożliwia zmianę parametrów takich jak maksymalna prędkość, przyspieszenie oraz skrętność kół, co uatrakcyjnia rozgrywkę.

Sam wyścig posiada system punktów kontrolnych, które należy przejeżdżać w kolejności, aby zaliczyć okrążenie. Gra wyświetla aktualny czas przejazdu danego sektora oraz okrążenia jak również najlepszy i średni czas. Ponadto minimapa pokazuje położenie gracza i kolejnego punktu kontrolnego naniesione na kształt trasy. Interfejs użytkownika wyświetla także numer aktualnego sektora oraz okrążenia. Po przejechaniu kolejnego fragmentu trasy możemy zobaczyć specjalny czytelny komunikat mówiący nam o czasie przejazdu.

Mapa, interfejs użytkownika i obiekty na mapie posiadają specjalnie przygotowane tekstury. W czasie przejazdu możemy natrafić na porożstawiane po drodze skrzynki i opony, z którymi nasz pojazd może kolidować.

2.2. Strona techniczna

2.2.1. Uwagi ogólne

Gra jest napisana w języku C++. Korzysta z bibliotek Box2D oraz SFML. Box2D jest wykorzystywany do wsparcia przy modelowaniu fizyki. SFML jest używany do renderowania grafiki, odbierania sterowania od użytkownika oraz do mierzenia czasu rozgrywki.

Gra jest podzielona na klasy zwane modułami: menu, race oraz workshop. Klasa wyścigu odbiera informacje o wciskanych przyciskach od użytkownika, przesyłając je do

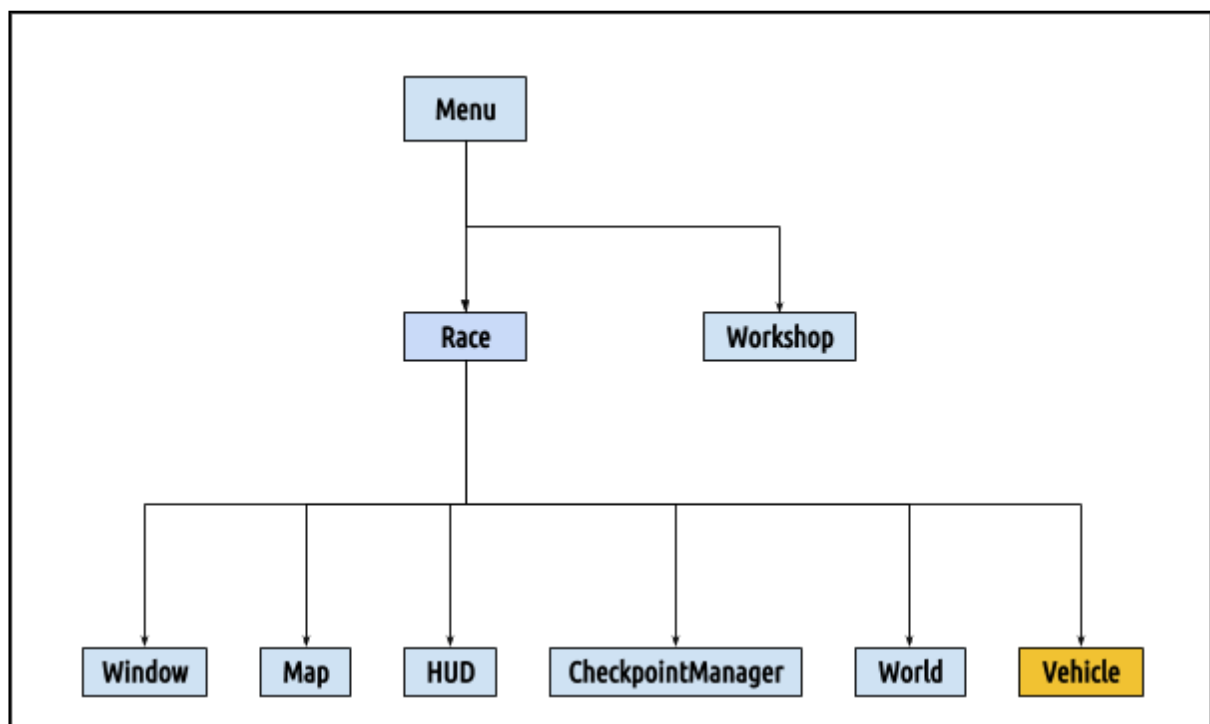
obiektu pojazdu. Wywołuje ona również metody *Update* na składowych gry; mapie, HUD (Head-Up Display), obiekcie świata biblioteki Box2D oraz oknie gry. CheckpointManager jak i HUD składają się z kilku mniejszych klas zarządzających funkcjonalnościami takimi jak zliczanie czasu przejazdu czy odbieranie informacji o osiągnięciu punktu kontrolnego.

2.2.2. Mechanika pojazdu

Samochód składa się z klasy pojazdu oraz klasy kół. Zarówno pojazd jak i koła posiadają wskaźniki do obiektów będącymi ich odpowiednikami z biblioteki box2d. Metoda update pojazdu wywołuje metody aktualizujące tarcie kół, skręt kół oraz aktualną siłę działającą na same koła. Koła na podstawie zaimplementowanej logiki wysyłają informację o zmianie przyspieszenia do obiektów swoich reprezentacji w Box2D.

2.2.3. Punkty kontrolne

Położenia punktów kontrolnych są wczytywane z pliku konfiguracyjnego. Posiadają one swoje odpowiedniki w box2d. Biblioteka przy wykryciu kolizji informuje obiekt punktu kontrolnego, który, jeśli jest oznaczony jako aktywny, wysyła informacje do managera. Manager śledzi, który punkt kontrolny jest aktywny, oraz korzystając z pomocniczej klasy timeManagera udostępnia dane przejazdu.



3. Scenariusz użycia

