

REV	DATA	ZMIANY
0.1	23.01.2018	<i>Michał Berdzik (michalberdzik97@gmail.com)</i>
0.1.1	01.02.2018	<i>Michał Berdzik (michalberdzik97@gmail.com)</i>

GRA “RIVER RAID”

Autor: Michał Berdzik
Akademia Górniczo-Hutnicza

Spis Treści

Spis Treści	2
Lista oznaczeń	3
Wstęp	4
Wymagania systemowe (requirements)	4
Funkcjonalność (functionality)	4
Analiza problemu (problem analysis)	5
Projekt techniczny (technical design)	6
Opisy klas	6
Diagram klas	8
Opis realizacji (implementation report)	9
Opis wykonanych testów (testing report) - lista buggów, uzupełnień, itd.	10
Podręcznik użytkownika (user's manual)	11
Opis gry	11
Sterowanie	11
Metodologia rozwoju i utrzymania systemu (system maintenance and deployment)	12

Lista oznaczeń

SDL	Simple DirectMedia Layer
SFML	Simple and Fast Multimedia Library
LMP	Lewy Przycisk Myszy
VBO	Vertex Buffer Array
NPC	Non Player Character
OpenGL	Open Graphics Library

Wstęp

Dokument dotyczy opracowania gry na wzór gry z lat 80 pod tytułem "River Raid" . Gra polega na sterowaniu samolotem, którym musimy niszczyć wrogie statki oraz zdobywać beczki z paliwem aby uzupełnić paliwo.

Wymagania systemowe (*requirements*)

- Windows 7 lub nowszy
- Karta graficzna z obsługą OpenGL powyżej wersji 2.0
- Minimum 1GB pamięci RAM
- 200MB wolnej przestrzeni na dysku

Funkcjonalność (*functionality*)

Do funkcjonalności gry możemy zaliczyć:

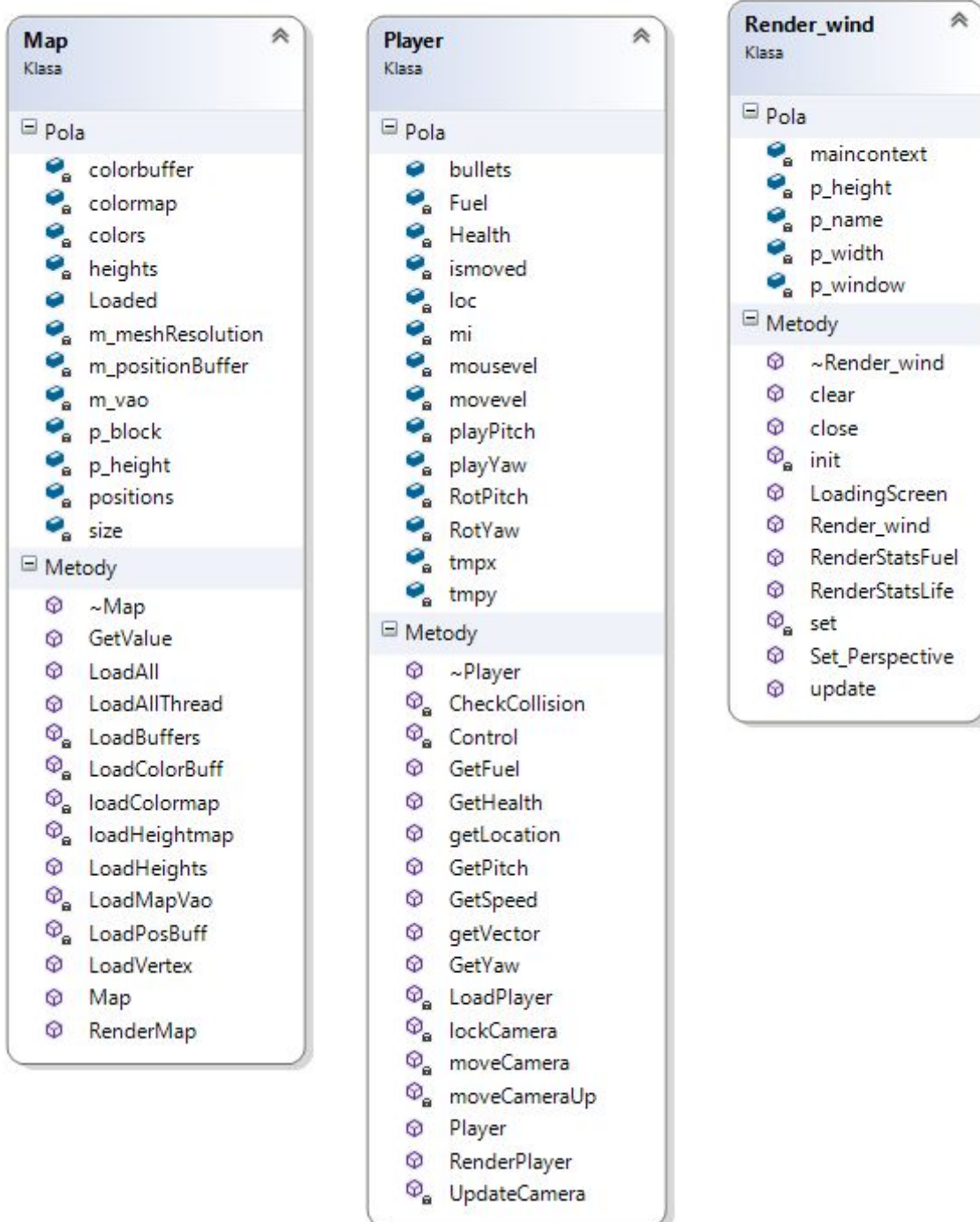
- Możliwość sterowania samolotem
- Możliwość strzelania
- Możliwość rozbicia się o elementy terenu, przez co tracimy punkty życia
- Ciągły pobór paliwa

Analiza problemu (*problem analysis*)

- **Tworzenie terenu:** Należy zaimplementować klasę reprezentującą mapę terenu, która będzie tworzyła otwarty świat za pomocą pliku opisującego teren.
- **Stworzenie samolotu:** Samolot powinien móc latać, strzelać oraz powinien być podatny na uszkodzenia związane z możliwymi zderzeniami z terenem świata
- **Sterowanie samolotem:** Sterowanie odbywa się za pomocą myszki oraz przycisku W i S, które odpowiednio zwiększają oraz zmniejszają prędkość samolotu. Ruch myszki natomiast determinuje w którą stronę samolot będzie leciał. Przyciśnięcie LPM powoduje wystrzał pocisku z samolotu.
- **Utrata punktów życia:** W wyniku zderzenia się z powierzchnią terenu samolot traci punkty życia, które się nie regenerują wraz z trwaniem gry

Projekt techniczny (*technical design*)

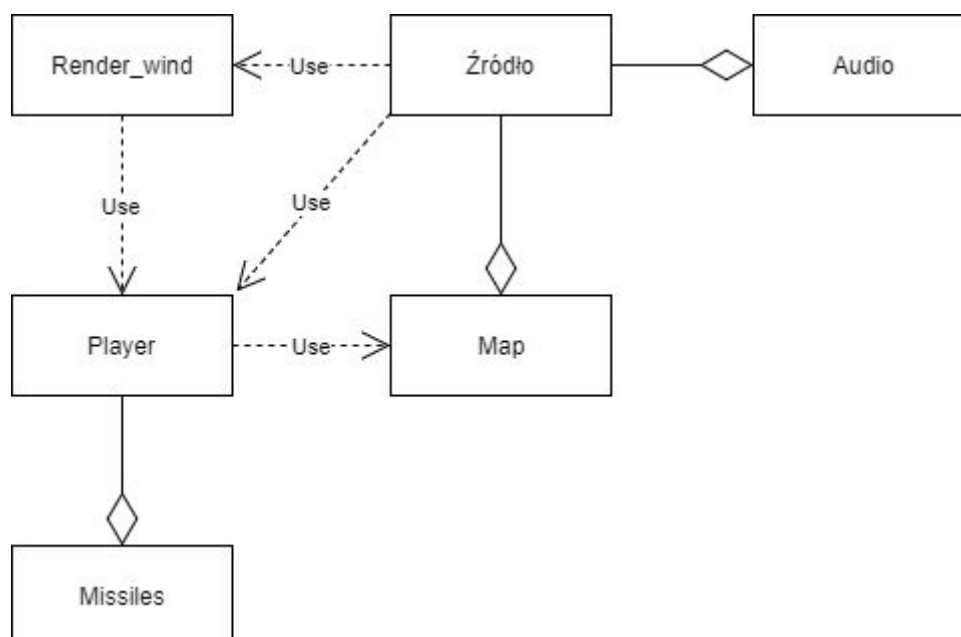
Opisy klas



The image displays three panels, each representing a class hierarchy in a software development environment. Each panel has a title bar with the class name and a 'Klasa' label, and a collapse icon (two upward-pointing arrows).

- vector3d** (Klasa):
 - Pola** (Attributes):
 - x
 - y
 - z
 - Metody** (Methods):
 - change (+ prze...)
 - changeX
 - changeY
 - changeZ
 - crossProduct
 - dotProduct
 - length
 - normalize
 - operator-
 - operator!=
 - operator*
 - operator*=
 - operator/
 - operator/=
 - operator+
 - operator+=
 - operator-=
 - operator==
 - vector3d (+ prz...
- Audio** (Klasa):
 - Pola** (Attributes):
 - music
 - p_path
 - Metody** (Methods):
 - ~Audio
 - Audio
 - IsPlaying
 - loadMusic
 - playMusic
 - playMusicInf
 - SetMusicVolume
 - SetUp
 - stopMusic
- Missiles** (Klasa):
 - Pola** (Attributes):
 - ammunition
 - factor
 - frequency
 - infinite
 - p_bullets
 - timer
 - Metody** (Methods):
 - ~Missiles
 - Add
 - Missiles
 - remove
 - render
 - update
 - Zagnieżdżone typy** (Nested types):
 - SingleBullet** (Struktura)

Diagram klas



Opis realizacji (*implementation report*)

Program został napisany w Visual Studio 2015 wykorzystując dodatkowe biblioteki takie jak SFML 2.1 oraz SDL2. Do zrealizowania warstwy graficznej użyłem interfejsu OpenGL. Do bardziej skomplikowanych rozwiązań jakie użyłem w programie można zaliczyć:

- **Wczytywanie terenu:** Operacja wytworzenia terenu wewnątrz programu polega na wczytaniu dwóch plików, jednego odpowiadającego za rozkład wysokości oraz drugiego odpowiadającego za kolory w danych punktach. Obie mapy bitowe mają rozdzielczość 1081 x 1081 px. Po załadowaniu plików zostają one użyte do stworzenia VBO, które przechowuje dane potrzebne do wyrenderowania terenu w pamięci RAM. Metoda ta jest o wiele szybsza niż poprzednia, która polegała na ciągłym tworzeniu i renderowaniu terenu, przez co można było poszerzyć obszar tworzonego świata do 3600 km² (świat to kwadrat 60km x 60km).
- **Tworzenie i usuwanie pocisków:** Pociski są tworzone w klasie Missiles, po wcześniejszym dodaniu pocisku (w klasie Player) do przechowującego je wektora *p_bullets*. Pocisk jest tworzony po naciśnięciu LMP w lokalizacji samolotu, wraz z jego kątem położenia w przestrzeni. Dzięki temu zabiegowi można z łatwością wystrzelić pocisk w kierunku w którym w danej chwili znajdował się samolot. Usuwanie pocisku następuje automatycznie w momencie gdy przeleci on zadaną mu odległość w jednej z 3 płaszczyzn (x,y lub z).

Opis wykonanych testów (*testing report*) - lista buggów, uzupełnień, itd.

Kod usterki	Data	Autor	Opis	Stan

Podręcznik użytkownika (*user's manual*)

Opis gry

Gra odbywa się na kwadratowej mapie 3D, reprezentującej wycinek realnego obszaru Norwegii wokół miasta Alesund o wymiarach 60km x 60km. W trakcie gry mamy nieograniczoną możliwość eksploracji terenu lecącym samolotem. Samolot może się rozбивać o elementy terenu, tracąc przy tym punkty życia. Ponadto wraz z poruszaniem się samolotu, jego poziom paliwa stale maleje. Oprócz tego mamy jeszcze możliwość strzelania nieograniczoną liczbą pocisków.

Na ekranie wyświetlają nam się informacje na ekranie odnośnie pozostałych punktów życia (po prawej stronie) oraz pozostałego paliwa (po lewej stronie).



Sterowanie

- W - zwiększenie prędkości samolotu
- S - zmniejszenie prędkości samolotu
- Ruch myszki - sterowanie samolotem
- LPM - wystrzał pocisku

Metodologia rozwoju i utrzymania systemu (*system maintenance and deployment*)

- Dodanie przeciwników oraz beczek paliwa uzupełniających paliwo samolotu
- Dodanie menu głównego oraz menu pauzy podczas gry
- Dodanie kolejnych poziomów
- Dodanie poziomów trudności (np. NPC którzy po wykryciu samolotu zaczęli by go gonić i ostrzeliwać)
- Dodanie efektów graficznych takich jak dynamiczna kamera lub efekty wybuchu pocisków podczas zderzenia