

Grzegorz Malinowski [147086]

Krzysztof Brzeziński [148155]

DOKUMENTACJA TECHNICZNA PROJEKTU

Guns'n'Jets

Przedmiot: PROGRAMOWANIE OBKIEKTOWE

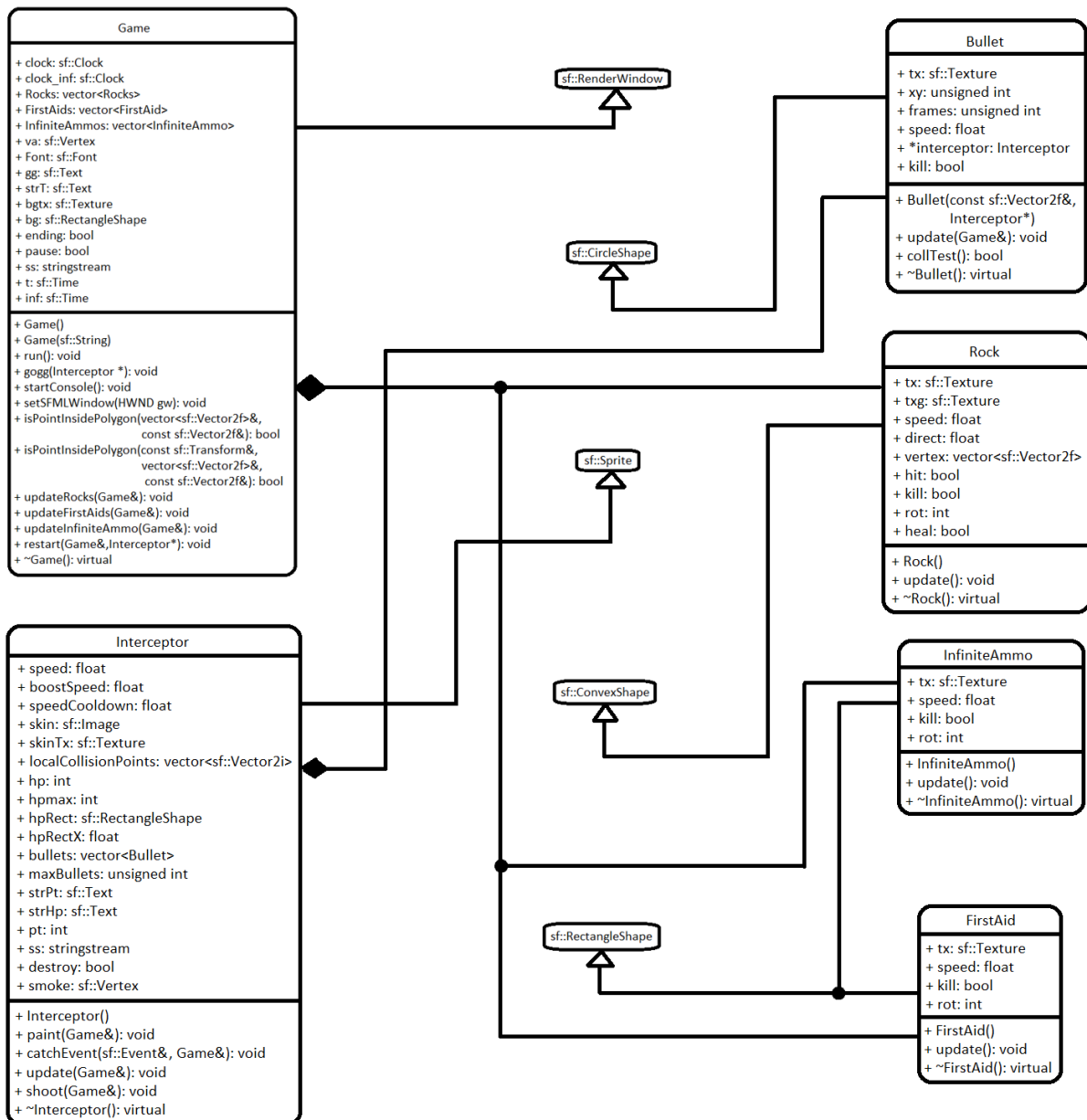
1. Zakres projektu

Program wykorzystywany do odbywania rozgrywki w trybie single player 2D. Cel gry polega na zdobyciu jak największej ilości punktów przy określonych warunkach brzegowych jak upływający czas i zmniejszający się poziom życia podczas kolizji obiektu sterowalnego z przeszkodami.

Program skierowany jest do wszystkich odbiorców ceniących samodzielną rozgrywkę typu akcja/zręczność, opartą o prostotę celu oraz wykonaną w klimacie lat 90.

Gra powstała w celu nauki współpracy języka c++ z biblioteką SFML oraz sposobów jej użycia i implementacji w kodzie.

2. Diagram klas



3. Wymagania systemowe

3.1 Wymagania funkcjonalne

a) Sterowanie obiektem- naciśnięcie [STRZAŁKI W DÓŁ] powoduje przesunięcie obiektu o określoną ilość pikseli w dół, analogicznie naciśnięcie [STRZAŁKI W GÓRĘ] powoduje przesunięcie obiektu o określoną ilość pikseli w górę. Naciśnięcie klawisza [SPACJA] powoduje wystrzelenie pocisku z obiektu sterowalnego.

Przytrzymanie klawisza [A] zmienia tryb strzału na wolny.

Przytrzymanie klawisza [D] zmienia tryb strzału na szybki.

b) Generowanie przeszkód – w określonej jednostce czasu generują się pseudolosowo nowe przeszkody.

c) Generowanie elementów wspomagających gracza tzw. Power-up – pojawienie się wspomagacza jest losowe w danej jednostce czasu, ponadto tuż przed pojawieniem się wspomagacza losowany jest jego typ. Typy wspomagaczy:

- * apteczka – przywrócenie 30% HP gracza

- * nieskończona amunicja – liczba pocisków staje się nieograniczona na 5 sekund

c) System kolizji – zderzenie przeszkody z obiektem sterowalnym powoduje utratę punktów życia, zderzenie pocisku z przeszkodą powoduje jej destrukcję lub zmniejszenie. Zderzenie z wyróżnionym elementem (zielona skała) powoduje dodanie punktów życia oraz zwiększenie ilości posiadanych pocisków.

d) Pauza– wciśnięcie klawisza [P] powoduje pauzę gry (lub jej wystartowanie w przypadku uruchomienia programu).

e) Czas i Punkty – podczas rozgrywki wyświetlane są zdobyte przez gracza punkty oraz pozostały czas do zakończenia rozgrywki.

f) Restart – po zakończonej rozgrywce wciśnięcie klawisza [R] powoduje rozpoczęcie gry na nowo

3.2 Wymagania pozafunkcjonalne

- a) Dynamiczne wyświetlanie rezultatów punktowych.
- b) Ochrona przed błędami wywołanymi wciśnięciem pojedynczych klawiszy nieobsługiwanych przez aplikację.
- c) Prostota obsługi i zrozumienia celu wykorzystania programu.
- d) Bezpieczeństwo rezultatów w postaci braku identyfikacji gracza oraz możliwości sprawdzenia jego poprzednich wyników.
- e) Dynamiczne odnawianie poziomu HP oraz ilości pocisków po zdobyciu odpowiedniego przedmiotu
- f) Płynność ruchu obiektów

4. Realizacja projektu

- Instalacja i wstępne zrozumienie działania biblioteki SFML
- Stworzenie koncepcji aplikacji
- Stworzenie klas
- Opracowanie mechanizmu kolizji
- Stworzenie funkcji odpowiedzialnych za punktację
- Zaprojektowanie mechanizmu wspomagaczy
- Stworzenie grafik
- Realizacja GUI – dodanie tła, zarządzanie umiejscowieniem elementów statycznych jak np. licznik

5. Kryteria akceptacyjne

- Uruchomienie programu
- Wciśnięcie klawisza [P], aby wystartować grę
- W zgodzie z umiejętnościami próba omijania przeszkód przy użyciu strzałek [GÓRA][DÓŁ] oraz klawisza [SPACJA] w celu wystrzelenia pocisków i destrukcji przeszkód, używanie klawiszy [A][D], które zmieniają tryb strzelania
- Po zakończeniu gry zrestartowanie gry klawiszem [R]