

DOKUMENTACJA TECHNICZNA PROJEKTU

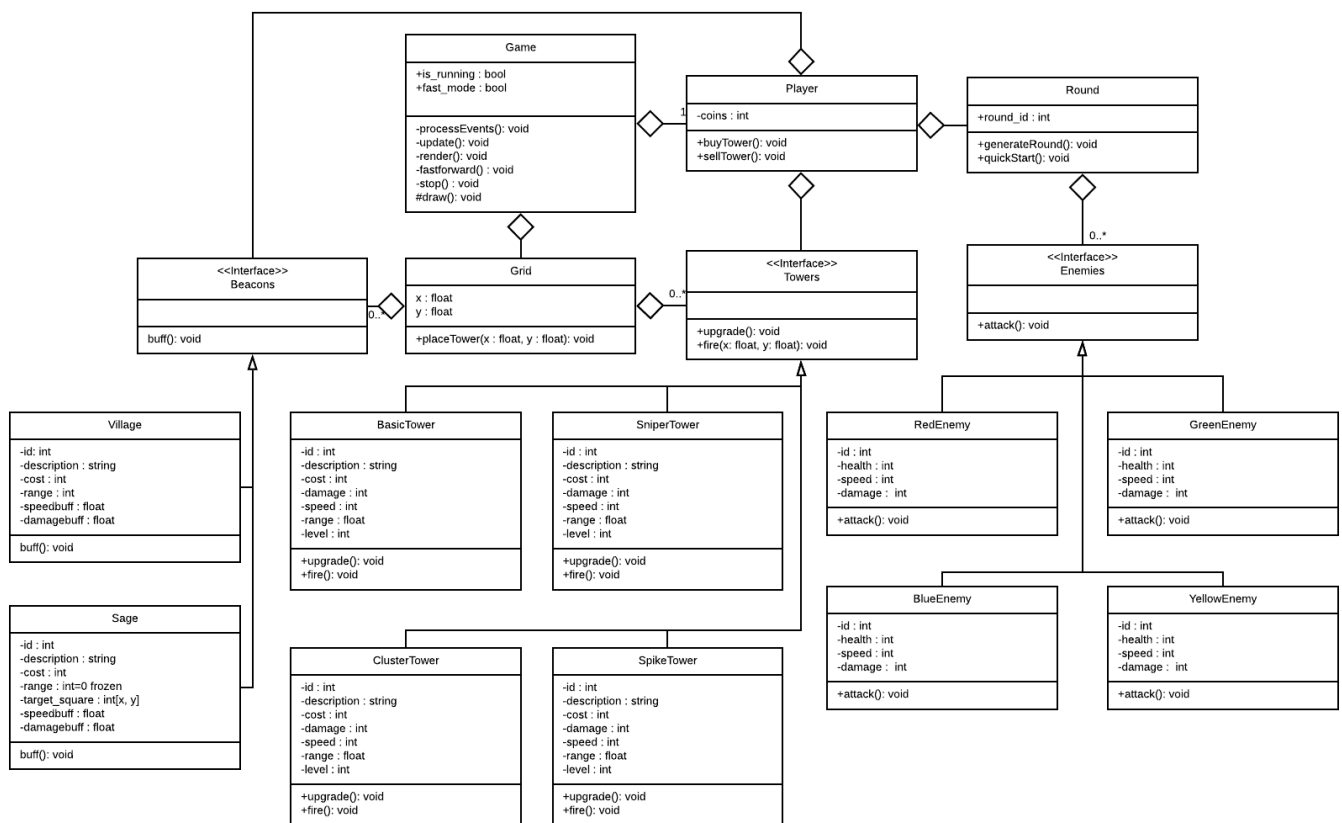
Gra typu Tower Defense [C++]

Przedmiot: PROGRAMOWANIE OBIEKTOWE

1 Zakres projektu

Projekt będzie zawierać instancję gry z 1 wbudowaną mapą. Będzie to gra oparta na gridzie 16x16 z możliwością budowania wież na kratkach, które nie są oznaczone jako "trasa" dla przeciwników. Samych wież będą 4 rodzaje i tyle samo tier-ów przeciwników, gdzie po "śmierci" przeciwnika na jego miejscu pojawia się jego poprzednik.

2 Diagram klas



3 Wymagania systemowe

3.1 Wymagania funkcjonalne

- Rendering - Renderowanie gui i samej planszy gry wraz ze strukturami oraz przeciwnikami.
- Placing Towers - Ustawianie struktur na odpowiednich slotach gridu.
- Shooting - Widoczny projectile, który przy styku z przeciwnikiem zadaje mu obrażenia.
- Hit Detection - Detektowanie czy przeciwnik (mob) otrzymał obrażenia.
- Wave Generation - Generowanie fali na podstawie predefiniowanej tablicy lub odpowiednich wartości.
- Selling Towers - Sprzedawanie i zwalnianie miejsca na inne struktury (Zwrot 70% wydanych coinsów)

3.2 Wymagania pozafunkcjonalne

- Łatwość modyfikacji - Przez obiektowe podejście dodawanie nowych funkcji jest ułatwione.
- Niezawodność - Wszystkie funkcje, które są zdefiniowane działają w instancji projektu.
- Funkcjonalne dopasowanie - funkcję uzupełniają się bez potrzeby powtarzania ich działania w innych.
- Łatwość nauki - sama gra będzie zawierać proste mechaniki, które są szybkie do nauki.
- Strukturalność - podejście strukturalne pozwala łatwiej zrozumieć działanie programu
- Samo-opisywalność - Nazwy funkcji mówią jaką funkcjonalność spełnia funkcja.

4 Realizacja projektu

- Projekt będzie realizowany z pomocą biblioteki SFML.
- Wieże będą miały określony range, dmg i upgrade'y (zmiana wartości dmg/range/attack speed).
- Przeciwnicy będą stworzeni w postaci takiej, że gdy giną (trafia ich wieża) z nich wypada przeciwnik o niższym poziomie.
- Na każdej kratce niebędącej trasą dla przeciwników będzie można postawić wieżę.

Realizacja samego projektu będzie przebiegała poprzez stworzenie samej planszy złożonej z kwadratów (grid), zrobienie 4 wież i 4 przeciwników.

5 Kryteria akceptacyjne

- Renderowanie planszy
- Generowanie fal
- Pokazywanie przeciwników i wież na planszy
- Stworzenie struktur, wież i przeciwników
- GUI dla ułatwienia wybierania wież i zmieniania opcji
- Zdefiniowanie działania wież i zachowania przeciwników