

Projekt Galaga

Opis funkcjonalny

Ogólne zasady gry :

- Galaga to gra typu shooter. Podczas rozgrywki gracz wciela się w pilota samolotu lecącego w przestrzeni kosmicznej. Jego celem jest niszczenie nadlatujących przeciwników - również samolotów. Gra polega na zdobyciu jak największej ilości punktów.
- Ilość zdobytych punktów zależy od ilości zestrzelonych przeciwników, skuteczności (ilość celnych strzałów do ilości zużytej amunicji, przy czym lepsze pociski zużywają więcej amunicji) oraz od ilości przeciwników wpuszczonych za „linię frontu” (za koniec okna gry).
- Gracz przegrywa w momencie zderzenia się z przeszkodą, przeciwnikiem lub którymkolwiek z pocisków.
- Przeszkody mają formę nadlatujących asteroid różnej wielkości. Niszczenie przeszkód nie jest możliwe.
- Są trzy typy przeciwników : zwykły (niebieski), strzelający (szary) oraz próbujący uciec przed pociskiem (czerwony). Przeciwnik zostaje zniszczony gdy trafi w niego jakikolwiek pocisk.
- Są również trzy typy broni : zwykły pocisk lecący prosto, pocisk potrójny składający się z jednego pocisku lecącego prosto, drugiego lecącego lekko w prawo, trzeciego lecącego lekko w lewo. Trzecim typem broni jest pocisk naprowadzany, który leci w kierunku najbliższego przeciwnika.

Moja wersja gry, w menu, pozwala użytkownikowi na :

- rozpoczęcie nowej rozgrywki
- wczytanie zapisanej gry
- wyświetlenie rekordów rozgrywek
- wyświetlenie zasad
- wyjście ;(

Natomiast podczas rozgrywki gracz może :

- sterować samolotem w pionie i poziomie w zakresach okna gry za pomocą strzałek
- strzelać, w tym :
 - pojedynczym pociskiem (spacja), zużywającym jeden nabój
 - potrójnym pociskiem (L ctrl + spacja), zużywającym 3 naboje
 - pociskiem naprowadzanym (L shift + spacja), zużywającym 3 naboje
- zapauzować rozgrywkę
- zapisać rozgrywkę
- ustanawiać nowe rekordy
- wyświetlić statystyki na koniec rozgrywki

Działanie i wykonanie programu

Program napisałem w języku C++ w standardzie C++11. Wykorzystałem środowisko Code::Blocks. Użyłem platformę SMFL (2.3.2), wspomagającą tworzenie prostych gier.

Program działa w oparciu o standardowe rozwiązanie. Stale realizuje on pętlę główną gry, podczas której wyświetlane jest okno gry, przyjmowane i rozpoznawane są akcje, które wykonuje gracz, obsługiwany jest cały mechanizm, ogólnie w pętli głównej dzieje się wszystko. Program został napisany obiektowo, w związku z tym zawiera klasy takie jak : okno, przeciwnik, przeszkoda, pocisk, samolot, zapis, zderzenie. Wykorzystane przeze mnie dziedziczenia po klasach pocisk i przeciwnik są publiczne.

Opiszę teraz po krótko funkcje i działanie każdej z klas.

- Okno : klasa odpowiedzialna za wyświetlanie okna gry (w każdym przejściu pętli głównej gry okno wyświetlane jest na nowo), rozporządza ona tym co ma być obecnie wyświetlane w oknie („gdzie jesteśmy w grze”), czyli np. menu, rozgrywka, zasady, pauza, etc. Zanim cokolwiek pojawi się na ekranie musi zostać wywołana jedna z metod klasy okno odpowiedzialna za wyświetlenie „tego czegoś”. Ta klasa spełnia również rolę „moderatora” gry, tzn. rozporządza czasem, kontroluje wynik, kontroluje tworzenie się przeciwników, przeszkód etc, rozpoznaje akcje, które wykonuje gracz.
- Samolot : klasa obsługuje samolot gracza, ładuje jego grafikę, przechowuje położenie, porusza nim, tworzy nowy pocisk w przypadku strzału.

- Przeciwnik : klasa obsługuje zachowanie się pojedynczego przeciwnika. Podobnie jak w przypadku klasy samolot, klasa przeciwnik łąduje odpowiednią grafikę, przechowuje położenie, porusza obiektem. Po tej klasie dziedziczą klasa przeciwnik strzelający i przeciwnik uciekający. Przeciwnik strzelający zawiera dodatkową metodą odpowiedzialną za tworzenie pocisków (strzelanie), natomiast przeciwnika uciekającego wyróżnia rozbudowana funkcja porusz.
- Przeszkoda : klasa jest analogiczna do klasy przeciwnik, jedyną różnicą jest fakt, że ta jedna klasa odpowiada za tworzenie „różnych” przeszkód – różnych pod względem rodzaju grafiki . W klasie przeciwnik, są klasy dziedziczące po niej i ładowanie innych grafik odbywa się w tych „młodszych” klasach., tutaj odbywa się to w ramach jednej klasy.
- Pocisk : jest to najprostsza klasa w programie, obsługuje ona obiekt pocisk analogicznie do tego jak klasa przeciwnik obsługuje obiekt przeciwnika. Klasa pocisk jest klasą bazową dla trzech innych klas, które rozróżnia sposób poruszania się. Pocisk skręcający oprócz podstawowego ruchu „do góry” skręca w prawo lub w lewo, pocisk naprowadzany wyszukuje najbliższego przeciwnika i podąża w jego stronę, a pocisk przeciwnika porusza się „w dół”.
- Zderzenie : klasa odpowiedzialna za sprawdzanie czy nie doszło do zderzenia dwóch obiektów podczas rozgrywki i podjęcie odpowiedniej akcji w przypadku zderzenia.
- Zapis : klasa odpowiedzialna za zapisywanie rekordów, wczytywanie ich, zapisywanie rozgrywki, wczytywanie zapisanych rozgrywek. Zapis odbywa się od dwóch plików tekstowych : jeden z wynikami i rekordami, drugi z zapisanym stanem gry (położeniami przeciwników, przeszkód, samolotów etc.)

Schemat dziedziczenia

