Увод

World of Warcraft е една от най-известните онлайн игри с така наречения „отворен свят“. Целта на този проект е да пресъзде най-базовите черти на оригиналната игра.

Основните цели и задачи са да се пресъздаде максимално много оригиналната игра, като се подържа:

1. от страна на потребителя:
   1. *combat* в реално време;
   2. удобен интерфейс за потребителя (играча), включващ „графики“ (използвайки само символи от *ASCII* таблицата), интерактивни прозорци за информация (*Info Box*-ове), плаващи надписи
   3. разнообразие от най-различни:
      1. предмети;
      2. противници;
      3. способности (*abilities*);
      4. класове, които потребителя може да играе;
   4. директен вход от клавиетура и мишка.
2. от страна на разработчика:
   1. диаграма, съдържаща нагледно йерархията използвана за направата на проекта;
   2. изчистеност и абстрактност на кода;
   3. избягване на така наречените „магически числа“

Обща архитектура

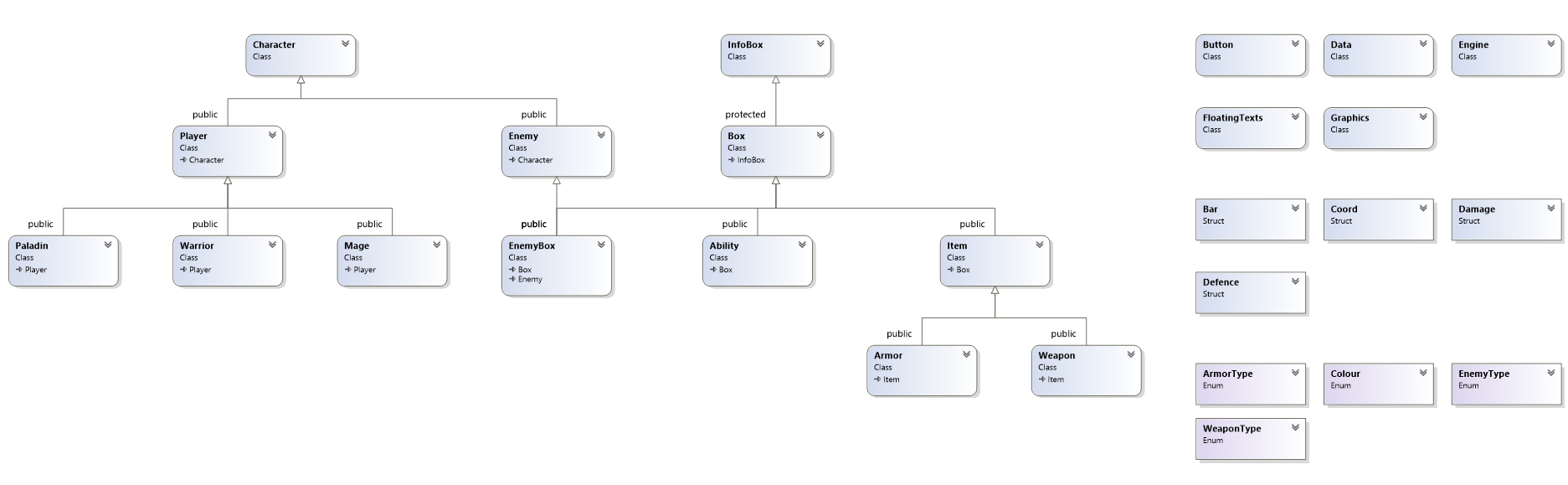
Проекта е организиран на 2 различни нишки (*thread*-а):

* **входна част**, която се грижи за входните данни;
* **обща част**, която използва горепосочената нишка като основно средство за изпълнението на функционалността си.

От своя страна **общата част** е разделена на:

* **основна**, която се грижи съответно за:
  + графиките;
  + двигателя (*engine*-а);
  + базата данни;
* **частна**, която се грижи за изискванията на съответния проект

За прегледност се предоставя тази диаграма на йерархията използвана за реализацията на проекта:



По подробна диаграма е предоставена в зип файла, в който се намира текущия документ.

Реализация

В тази секция (*section*) ще разгледаме как е реализирана йерахията от класове.

* **Входна част**:

Тя бива изпълнявана успоредно на **общата част** като отелна нишка (*thread*) thread inputThread(Engine::inputReader), която единствената ѝ работа е в случей че всички входни евенти са обработени от **общата част** да прочете чрез функцията BOOL ReadConsoleInput(HANDLE, PINPUT\_RECORD, DWORD, LPDWORD) (Microsoft), изискваща <windows.h>, и да запише във опашка всички входни евенти, които са възникнали от последното извикване на същата функция.  
Реализиране е и функция void Engine::resetEvents(), достъпна глобално, която в случей на нужда отбелязва евентите в опашката като прочетени, което от своя страна ще предизвика нишката, отговаряща за **входната част** да прочете и запише в опашката новия вход.

* **Обща част**:
  + Графики:

Реализирана е чрез статичен клас (*singleton*), като функционалността ѝ е разделена на абстрактна част, която да бъде използвана без излишни спесифики от **двигателя** и **частната част** и на „конкретна“ част, която използва абстрактната за оформянето на удобен *User Interface* на потребителя.

* + База данни:

Отново е реализирана чрез статичен клас (*singleton*), изпълняващ всички изисквания за енкапсулация на код поради простата причина, че защитата на данните е една от най-важните части на всяка програма.

* + Двигател:

Трети, но не по важност, статичен клас (*singleton*), който навръзва всички елементи на програма в едно цяло. В него се съдържат главните функционалности съответно: началното меню, инвентара, магазина, книгата за способности (*Ability book*-а), картата и самия *gameplay*.

Заключение

Всички основни цели по отношение на:

* *user experience*, поставени при започването на проекта, бяха успешно реализирани. Въпреки това много ключови моменти, които не бяха включени към основни цели и задачи, като например туториал и *in-game* информация за голяма част от функционалността липсва.
* *developer experience*, бяха предимно реализирани успешно. Въпреки положените усилия за липса на „магически числа“, поради ниската разолюция на игровия прозорец на места в програмата все още се намират такива „магически числа“ с цел центриране на изходния текст. Друг недостатък на кода е, че двигателя (*engine*-а) първоначално беше замислен като нещо напълно абстрактно, което да бъде използвано външно, но поради ограниченото време, беше взето решенеие вътре в самия клас, да бъде включена главната функционалност на програмата, което малко обезсмисли цялата идея.

Въпреки, че не всички цели бяха изпълнени, проекта като цяло дава усещане за завършеност. Поради изчистеността на кода и обектно ориентирания стил на писане, бъдещи разработки по кода, като например решения на гореизброените проблеми (изключвайки двигателя), ще бъдат изпълнявани с лекота.

# Bibliography

Turner, R., Niksa, M., & Wojciakowski, M. (n.d.). Retrieved from Microsoft: https://docs.microsoft.com/en-us/windows/console/readconsoleinput