**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра МО ЭВМ**

**ОТЧЕТ**

**по лабораторной работе №6**

**по дисциплине «Объектно-ориентированное программирование»**

**Тема: Шаблонные классы**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Студент гр. 8303 | |  | | Кибардин А.Б. | |
| Преподаватель |  | |  | | Филатов Ан.Ю. |

Санкт-Петербург

2020

**Цель работы.**

Разработка и реализация набора классов правил игры.

**Задание.**

Основные требования:

* Правила игры должны определять начальное состояние игры
* Правила игры должны определять условия выигрыша игроков
* Правила игры должны определять очередность ходов игрока
* Должна быть возможность начать новую игру

**Ход работы.**

1) Для реализации правил игры были написаны классы Rule1 и Rule2, отнаследованные от класса Rule(файлы Rule.h/.cpp).

Каждый из классов задает начальные размеры поля и расположение баз(в первом правиле) и юнитов(во втором правиле).

Первый класс реализует правило защита базы. Правило заключаются в том, что каждый из игроков должен уничтожить базу соперника. У каждого игрока в свой ход есть возможность создать нового юнита, если их число не превышает заданного, перемещать юнитов и атаковать. После ход передается следующему игроку. Победа наступает в момент, когда один из игроков разрушит вражескую базу.

Второй класс реализует правило титаны. Правило заключается в том, что кажды из игроков должен уничтожить юнита-титана другого игрока. Каждый игрок на старте выбирает класс юнита-титана, у которого будут увеличенные характеристики, по сравнению с базовыми юнитами. В данном правиле игроки имеют возможность перемещать своего юнита и атаковать вражеского. Победа наступает в случае смерти одного из юнитов.

2) Для определения выигрыша игроков в классах правил определены методы loser(). В первом правиле происходит проверка на наличие базы на поле. Если на поле база отстутствует — база была разрушена, то возвращается номер проигравшего игрока. Во втором правиле производится проверка на наличие юнитов на поле. Если юнит отсутствует, то, аналогично методу loser в первом правиле, возвращается номер проигравшей стороны.

3) Для определения очередности ходов игрока был написан виртуальный класс State(State.h) методом handle. Для реализации хода каждого из игроков были написаны классы State1, State2, State3, которые реализуют метод handle для хода каждого из игроков. Каждый из методов передает команду пользователя, его номер в обработчик цепочки команд. Для передачи хода следующему игроку в классе Rule содержатся указатели на классы состояний(State1, State2, State3) и указатель на текущее состояние, а так же метод nextPlayer, который вызывается если пользователь вводит команду, соответствующую ходу одного из игроков. Внутри метода nextPlayer производятся проверки на равенство одного из трех состояний и текущего состояния. При обнаружении совпадения, текущее состояние меняется на состояния для следующего игрока.

4) Для возможности начать новую игру в методе play() шаблонного класса Game производится проверка на ввод команды «newGame». При вводе данной команды пользователь сможет начать новую игру в данном правиле игры.

5) Написан шаблонный класс игры Game(Game.h), поддерживающий разное количество игроков. В качестве параметров передаются конкретные правила игры (Rule1 или Rule2). Для этого от базового класса GameBase отнаследован класс Game c входным параметром правил игры. Далее отдельно специализированы классы для разного количества игроков.

6) Класс игры один единственный и создается паттерном синглтон. Для реализации данного паттерна конструктор класса Game сделан приватным. В приватном поле класс Game хранит указатель на себя. В статическом методе получения объекта getInstance() производится проверка был ли создан экземпляр класса. Если экземпляр не был создан, то вызывается конструктор класса. Возвращается экземпляр класса Game. С помощью данного паттерна гарантируется единственность экземпляра класса.

**Выводы.**

В ходе выполнения работы были изучены принципы реализации правил игры с помощью шаблонных классов. Так же реализованы паттерны Состояние для передачи хода между игроками и Синглтон для реализации класса игры.