**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 |  | Логинов Е.А. |
| Преподаватель |  | Филатов А.Ю. |

Санкт-Петербург

2020

**Цель работы.**

Разработать и реализовать набор классов правил игры.

**Ход работы.**

1. Было написано два класса правил игры Rule1 и Rule2 (файл Rules.h). Классы наследуются от класса RuleBase.

Оба правила предусматривают свободный бой игроков, заключающийся в поочередном ходе. Ход передается следующему игроку после создания или перемещения юнита.

Для первого правила проигравшим считается тот, у которого не осталось ни одного юнита на поле. Для второго правила проигравшим считается игрок, на базе которого не осталось ни одного юнита.

Каждое правило определяет начальные условия игры и создает в своем методе Field\* createField(Adapter\* adapter) поле, соответствующее своему правилу. В первом пра виле создается поле 8\*8, во втором 11\*11, также на поле создаются две базы (в соответствии с правилом). Метод возвращает созданное поле.

2) Каждое правило при помощи метода isLose() определяет условие победы игрока. Например, для правила 1 условие выигрыша – это отсутствие у противника всех юнитов на поле (возвращается номер проигравшего игрока, 0 или 1). Ничья будет в том случае, если на поле нет ни одного юнита из разных баз(возвращается 2).

Во втором правиле критерием проигрыша является отсутствие юитов на базе.

3) Для определения очередности хода игроков используется паттерн «Состояние». Класс State (файл State.h), имеет чисто виртуальный метод handle. От класса State наследуются классы State1 и State2, предназначенные для управления ходом первого и второго игрока соответственно. Каждый из классов переопределяет метод handle. В своих методах каждый из классов передает в обработчик цепочки команд chainOfResponsibility (chainOfResponsibility.h**)** строку с командой и номер базы, для которой нужно применить эту команду.

Для передачи ходов в классе Rules содержатся 2 различных экземпляра состояний и указатель currentState, на текущее состояние. В классе правил Rules есть метод changeState, который изменяет указатель currentState на экземпляр другого состояния. Каждое правило содержит метод handle(), который отвечает за передачу хода другому игроку. Каждое правило (Rules 1, 2) проверяет поступающую строку и в зависимости от ее значения вызывает или нет метод changeState(), который меняет текущее состояние.

4) Был реализован шаблонный класс игры GameFacade (файл GameFacade.h). В качестве шаблона он принимает правило Rule1 или правило Rule2.

C помощью метода правила createField() создается поле. В методе play() Запускается бесконечный цикл игры. В нем с помощью метода isLose() проверяется условие окончания игры, если игра не окончена, то считывается команда игрока и передается в обработчик handle() класса Rule. Иначе, если метод isLose() возвращает номер проигравшего игрока, выводится сообщение о проигрыше соответствующего игрока и происходит выход из бесконечного цикла игры.

5) Также была добавлена возможность начало новой игры, используется команда "newGame". Создается новое поле при помощи метода createField(), на котором размещаются две базы.

**Выводы.**

Был реализован набор классов правил игры и шаблонный класс игры, принимающий в качестве шаблона определенное правило.