

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
МОЭВМ

ОТЧЕТ
по практической работе № 3
по дисциплине «ООП»
Тема: Логическое разделение классов

Студент гр. 8383

Степанов В.Д.

Преподаватель

Жангиров Т. Р.

Санкт-Петербург

2020

Цель работы.

Изучение и применение паттернов программирования для разработки игры на языке C++.

Постановка задачи.

Разработать и реализовать набора классов для взаимодействия пользователя с юнитами и базой. Основные требования:

- Должен быть реализован функционал управления юнитами
- Должен быть реализован функционал управления базой

Выполнены все основные требования к взаимодействию
Добавлен функционал просмотра состояния базы
Имеется 3+ демонстрационных примера
*Реализован паттерн “Фасад” через который пользователь управляет программой
*Объекты между собой взаимодействуют через паттерн “Посредника”
*Для передачи команд используется паттерн “Команда”
*Для приема команд от пользователя используется паттерн “Цепочка обязанностей”

Ход работы.

1. В классе базы был реализован метод `spawn`, который создает нового юнита, по требованию игрока (метод реализован в фалах `Base.hpp` и `Base.cpp`). За всю игру может быть создано игроком или базой 4 юнита каждого типа, кроме типа `Smoker`. Чтобы создать юнита можно ввести команду `<“spawn”><‘space’><тип юнита><‘space’><номер игрока>`. Были реализованы методы атак для каждого юнита. Визуализация атаки для каждого типа представлена на рисунках 1 – 3. Для того чтобы атаковать

нужно ввести команду <направление><'h'><'space'><тип юнита<номер юнита> (Пример: wm !1).

2. Каждый раз, когда происходит какое либо изменение юнитов, в консоль печатаются игровое поле и состояние всех юнитов, находящихся на карте. Пример вывода представлен на рисунке 4. Реализую методов можно увидеть в файлах Smoker.cpp, Shooter.cpp и Runner.cpp.

		!1		

Рисунок 1 – Атака юнита Runner (ah !1)

Рисунок 2 – Атака юнита Shooter (ah <1)

		1 \$		

Рисунок 3 – Атака юнит Smoker (ah \$1)

	BA		<1		B1						
	HP	((!5		\$1						
	##	WW	!3		<3						
	6!PC		PC	!1							
	2>		4!WW								
	2\$		2!		BA						
	B2		4>		HP						

\$1	h = 15	a = 15	p = 1	2\$	h = 15	a = 15	p = 1				
<1	h = 5	a = 10	p = 4	2>	h = 5	a = 10	p = 4				
<3	h = 5	a = 10	p = 4	4>	h = 5	a = 10	p = 4				
!1	h = 8	a = 13	p = 5	2!	h = 8	a = 13	p = 5				
!3	h = 8	a = 13	p = 5	4!	h = 8	a = 13	p = 5				
!5	h = 8	a = 13	p = 5	6!	h = 8	a = 13	p = 5				

Рисунок 4 – Вывод игрового поля и состояния юнитов

3. Был реализован паттерн “Фасад”. Был создан класс Facade, который имеет методы печати поля printField и вывода характеристик юнитов printPoints. Реализацию можно увидеть в файлах Facade.cpp и Facade.hpp.

4. С помощью паттерна “Посредник” была реализована специальная способность каждого игрока. Каждый игрок может раз в три хода отнять от здоровья любого юнита число, равное атаке юнита Smoker, принадлежащего игроку. Реализацию можно посмотреть в файлах Mediator.hpp и Mediator.cpp.

5. Был реализован паттерн “Команда”. Был создан интерфейс Command для реализации команд. Из данного интерфейса были созданы классы команды передвижения MoveCommand и взаимодействия юнита HitCommand. Был создан класс CommandHandler, который содержит и вызывает выше перечисленные команды. Реализации представлены в файлах Command.hpp, MoveCommand.hpp, MoveCommand.cpp, HitCommand.hpp, HitCommand.cpp, CommandHandler.hpp, CommandHandler.cpp.

6. Был реализован паттерн “Цепочка обязанностей” для проверки корректного ввода команд. Реализация представлена в файлах Handler.hpp, AbstractHandler.hpp, AbstractHandler.cpp, SpawnHandler.cpp,

SpawnHandler.hpp, ExitHandler.cpp, ExitHandler.hpp, MotionHandler.hpp, MotionHandler.cpp, SpecHandler.cpp, SpecHandler.hpp. Применения паттерна находится в файле Game.cpp, в методе action.

Примеры работы.

```

-----
|  BA##  <1  B1 |
| HP      !5  $1 |
|          WW!3  |
|        6!PC  PC!1 |
| 2>  4!WW      <3 |
|  2$  2!      BA |
| B2  4>      HP |
|-----
$1  h = 15  a = 15  p = 1  2$  h = 15  a = 15  p = 1
<1  h = 5   a = 10  p = 4  2>  h = 5   a = 10  p = 4
<3  h = 5   a = 10  p = 3  4>  h = 5   a = 10  p = 4
!1  h = 8   a = 13  p = 5  2!  h = 8   a = 13  p = 5
!3  h = 8   a = 13  p = 5  4!  h = 8   a = 13  p = 5
!5  h = 8   a = 13  p = 5  6!  h = 8   a = 13  p = 5
ah <3
I'm shooting at you
-----
|  BA##  <1  B1 |
| HP      !5  $1 |
|          WW!3  |
|        6!PC  PC!1 |
| 2>  4!WW      <3 |
|  2$  2!      BA |
| B2  4>      HP |
|-----
$1  h = 15  a = 15  p = 1  2$  h = 15  a = 15  p = 1
<1  h = 5   a = 10  p = 4  2>  h = 5   a = 7   p = 4
<3  h = 5   a = 10  p = 3  4>  h = 5   a = 10  p = 4
!1  h = 8   a = 13  p = 5  2!  h = 8   a = 13  p = 5
!3  h = 8   a = 13  p = 5  4!  h = 8   a = 10  p = 5
!5  h = 8   a = 13  p = 5  6!  h = 8   a = 13  p = 5

```

Рисунок 5 – Пример атаки юнита

```

-----
| BA## <1 B1 |
| HP !5 |
| WW!3$1 |
| 6!PC PC!1 |
| 2> 4!WW <3 |
| 2$ 2! BA |
| B2 4> HP |
-----

$1 h = 15 a = 15 p = 1 2$ h = 15 a = 15 p = 1
<1 h = 5 a = 10 p = 4 2> h = 5 a = 7 p = 4
<3 h = 5 a = 10 p = 3 4> h = 5 a = 10 p = 4
!1 h = 8 a = 13 p = 5 2! h = 8 a = 13 p = 5
!3 h = 8 a = 13 p = 5 4! h = 8 a = 10 p = 5
!5 h = 8 a = 13 p = 5 6! h = 8 a = 13 p = 5
spawn ! 1

-----
| BA## <1 B1 |
| HP !5 !7 |
| WW!3$1 |
| 6!PC PC!1 |
| 2> 4!WW <3 |
| 2$ 2! BA |
| B2 4> HP |
-----

$1 h = 15 a = 15 p = 1 2$ h = 15 a = 15 p = 1
<1 h = 5 a = 10 p = 4 2> h = 5 a = 7 p = 4
<3 h = 5 a = 10 p = 3 4> h = 5 a = 10 p = 4
!1 h = 8 a = 13 p = 5 2! h = 8 a = 13 p = 5
!3 h = 8 a = 13 p = 5 4! h = 8 a = 10 p = 5
!5 h = 8 a = 13 p = 5 6! h = 8 a = 13 p = 5
!7 h = 8 a = 8 p = 7

```

Рисунок 6 – Пример создания юнита игроком

