МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра математического обеспечения и применения ЭВМ

ОТЧЕТ

по лабораторной работе №2

по дисциплине «Объектно-ориентированное программирование»

Тема: Интерфейсы классов, взаимодействие классов, перегрузка операций

Студент гр. 8383 _____ Костарев К.В.

Преподаватель _____ Жангиров Т.Р.

Санкт-Петербург

2020

Цель работы.

Продолжить работу над игрой, изучить интерфейсы классов, их взаимодействие и перегрузку операций.

Постановка задачи.

Разработать и реализовать набор классов:

- Класс базы
- Набор классов ландшафта карты
- Набор классов нейтральных объектов поля

Класс базы должен отвечать за создание юнитов, а также учитывать юнитов, относящихся к текущей базе. Основные требования к классу база:

- База должна размещаться на поле
- Методы для создания юнитов
- Учет юнитов, и реакция на их уничтожение и создание
- База должна обладать характеристиками такими, как здоровье, максимальное количество юнитов, которые могут быть одновременно созданы на базе, и.т.д.

Набор классов ландшафта определяют вид поля. Основные требования к классам ландшафта:

- Должно быть создано минимум 3 типа ландшафта
- Все классы ландшафта должны иметь как минимум один интерфейс
- Ландшафт должен влиять на юнитов (например, возможно пройти по клетке с определенным ландшафтом или запрет для атаки определенного типа юнитов)
- На каждой клетке поля должен быть определенный тип ландшафта

Набор классов нейтральных объектов представляют объекты, располагаемые на поле и с которыми могут взаимодействие юнитов. Основные требования к классам нейтральных объектов поля:

- Создано не менее 4 типов нейтральных объектов

- Взаимодействие юнитов с нейтральными объектами, должно быть реализовано в виде перегрузки операций
- Классы нейтральных объектов должны иметь как минимум один общий интерфейс

Выполнение работы.

Так как преследуется цель разработать градостроительный симулятор, пришлось переосмыслить назначение базы в игре: базой будет считаться мэрии города (муниципалитеты). Мэрии аналогично базе имеют возможность создавать (строить) юнитов (здания), аналогично районам города в реальной жизни. Мэрии имеют аналог здоровья: коэффициент благосостояния (оппозиции), если он положителен, то в базе (мэрии) все стабильно, и наоборот.

class Base:public IUnit

Класс базы, описан в файлах Base.h, Base.cpp.

База размещается на клетке игрового поля и имеет со зданиями один интерфейс IUnit, так как тоже является зданием.

int opposition, maxBuildings, numOfBuildings – характеристики базы: индекс оппозиции, максимальное количество зданий, которое может контролировать база, количество зданий, принадлежащих базе.

vector <IUnit*> buildings – здания, которые контролирует база. Необходимо для компоновщика и статистики.

void addBuild(int x, int y, Builds type, Board* board) – метод для создания юнитов (строительства зданий).

class Statistics

Компоновщик характеристик зданий для базы, описан в файлах Statistics.h, Statistics.cpp.

Statistic(Base* base) – конструктор для базы base.

class ILandscape, class FabricOfLandscape, class Proxy:public ILandscape

Интерфейс ландшафтов карты, фабрика ландшафтов, и паттерн взаимодействия ландшафта с юнитом (зданием), описаны в файлах ILandscape.h, ILandscape.cpp.

virtual Landscapes typeOfLandscape() возвращает тип ландшафта, все типы описаны в enum Landscapes в файле Naming.h

От интерфейса ILandscape унаследованы следующие классы-типы ландшафтов: Hill (холм), Lake (озеро), Plain (равнина).

Proxy содержит ландшафт landscape, с которым необходимо взаимодействовать, и функцию doSomething(IUnit* build), которая определяет, как landscape влияет на build. Так, на холму может располагаться только лишь электростанция и наоборот, так как важно соблюдать экологию и поэтому все станции в игровом мире – ветровые (единственно исключение в виде нейтрального объекта Gaz, см. ниже). На озере может быть только Водоканал, и наоборот.

class INeutral, class FabricOfNeutral

Интерфейс нейтральных объектов карты и фабрика, описаны в файлах INeutral.h, INeutral.cpp.

virtual Neutrals typeOfNeutral() возвращает тип нейтрального объекта, все типы описаны в enum Neutrals в файле Naming.h.

virtual IUnit* operator += (IUnit* build) перегрузка оператора += для реализации взаимодействия с юнитами.

От класса унаследованы следующие классы нейтральных объектов: Chernozem (плодородная почва), Relics (реликвии), Radiation (радиация), Gas (природный газ). Почва действует только на Ferma и увеличивает количество вырабатываемой ей пищи. Реликвии действуют на Office (увеличивает доходность) и на School (школы становятся исследовательскими центрами, и поэтому увеличивается расходность). Радиация действует на любое здание: опустошает их (сбрасывает все характеристики до 0, а сальдо уменьшает до -

100). Природный газ действует только на Powerhouse и увеличивает выработку электроэнергии.

Нейтральные объекты располагаются на клетке игрового поля с вероятностью 25%. При этом, они не отображаются на карте и игрок узнает о них только тогда, когда они начинают действовать на здание.

Демонстрация работы.

1.

Начало работы. Отрисовка карты (~ - озеро, вода; ^ - гора; _ - равнина).

```
Enter width and height:
5 5
Buildings on board: 0
Coin: 2000
_ ~ ^ ^ _
^ ~ _ _ ^
_ ~ ^ _ ^
^ ~ ~ ~ _
~ ~ _ _ _
You want to know more about the individual cell? (1 - yes, 0 - no)
0
```

Сначала необходимо построить мэрию (первая база).

```
ADD CITY HALL, Enter x, y:
5 4
```

Построим здание, для начала нужно определить базу, к которой оно будет относиться. Попробуем построить дачу на озере, например в (2,2):

```
Select a base (city hall), Enter x, y:
5 4
Add Build, Enter x, y and type of Building (D-0, H-1, F-2, O-3, P-4, S-5, V-6):
2 2 0
```

Дача не построилась. Ожидаемо.

```
Buildings on board: 0
Coin: 1700
_ ~ ^ ^ _
^ ~ _ _ C
_ ~ ^ _ ^
^ ~ ~ ~ _
~ ~ _ _ _
You want to know more about the individual cell? (1 - yes, 0 - no)
```

Попробуем вывести характеристики нашей базы:

```
You want to know more about the individual cell? (1 - yes, 0 - no)
1
Enter x, y:
5 4
CITY HALL.
Sum Saldo: 0
Sum Employment: 0
Sum Education: 0
Sum Energy: 0
Sum Water: 0
Sum Satiety: 0
```

Логично, так как дача так и не построилась.

2.

Попробуем построить дачу, электростанцию и ферму.

При строительстве даче оказалось, что мы ее построили на месторождении газа, а могли бы построить там электростанцию (в этом и смысл скрытия нейтральных объектов на карте).

```
Enter width and height:
5 5
Buildings on board: 0
Coin: 2000
~ _ ^ ~ ~
_ ^ _ ~ ^
_ ~ ~ ^ _
^ ^ ~ ~ ~
^ _ _ ^ _
You want to know more about the individual cell? (1 - yes, 0 - no)
0
ADD CITY HALL, Enter x, y:
2 1
Select a base (city hall), Enter x, y:
2 1
Add Build, Enter x, y and type of Building (D-0, H-1, F-2, O-3, P-4, S-5, V-6):
3 1 0
Neutral object on cell: 3
(0 - chernozem, 1 - relics, 2 - radiation, 3 - gas)
```

Больше при строительстве ни один нейтральный объект не попался. При окончании строительства видно, что городу не хватает только лишь школы.

```
Coin: 1700
~ _ ^ ~ ~
_ ^ _ ~ ^
_ _ ~ ~ ^
_ ~ ~ ^ _
^ ^ ~ ~ ~
P C D ^ F
You want to know more about the individual cell? (1 - yes, 0 - no)
1
Enter x, y:
2 1
CITY HALL.
Sum Saldo: 62
Sum Employment: 14
Sum Education: -1
Sum Energy: 1072
Sum Water: 1159
Sum Satiety: 26
```

3.

Покажем, что нейтральные объекты действительно влияют на здание. Для этого на время изменим программу так, что на всех клетках находится радиация. Попробуем построить хрущевку.

```
ADD CITY HALL, Enter x, y:
2 1
Select a base (city hall), Enter x, y:
2 1
Add Build, Enter x, y and type of Building (D-0, H-1, F-2, O-3, P-4, S-5, V-6):
3 1 0
Neutral object on cell: 2
(0 - chernozem, 1 - relics, 2 - radiation, 3 - gas)
Buildings on board: 2
Coin: 1700
^ _ ~ ~ ~
^ ~ ~ _ ^
^ ^ ^ _ ~
^ _ ~ _ ~
~ C D ~ ~
You want to know more about the individual cell? (1 - yes, 0 - no)
1
Enter x, y:
3 1
Saldo: -100
Energy: 0
Water: 0
Eat: 0
Workmans: 0
Students: 0
Kids: 0
```

Видно, что радиация негативно повлияла на хрущевку.

Выводы.

В ходе выполнения данной лабораторной работы были изучены интерфейсы классов, их взаимодействие и перегрузка операций.