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

ОТЧЕТ

по лабораторной работе №2

по дисциплине «Объектно-ориентированное программирование»

Студент гр. 4384 _____ Водолазко В.О.

Преподаватель _____ Жангиров Т.Р.

Санкт-Петербург

2025

Цель работы.

Изучить принципы объектно-ориентированного программирования. Написать программу на языке C++, которая будет прототипом пошаговой игры с перемещением персонажа и сражением с врагами.

Задание.

На 6/3/1 баллов:

1. Создать интерфейс карточки заклинания. Заклинание должно применяться игроком. На использование заклинания игрок тратит один ход.

2. Создать класс “руки” игрока, которая содержит все карточки заклинаний, которые игрок может применить в свой ход. Изначально рука игрока содержит только одно случайное заклинание. Реализовать возможность получать новые заклинание игроком, например, тратить очки на покупку или после уничтожения определенного кол-ва врагов. Размер “руки” должен быть ограничен и задается через конструктор.

3. Реализовать интерфейс заклинанием прямого урона. Это заклинание при использовании должно наносить урон врагу или вражескому зданию, если они находятся в достижимом радиусе. Если в качестве цели не выбран враг или вражеское здание, то заклинание не используется.

4. Реализовать интерфейс заклинания урона по площади. Это заклинание при использовании в допустимом радиусе наносит урон по области 2 на 2 клетки. Заклинание используется, даже если там нет никого.

На 8/4/1.5 баллов:

5. Реализовать интерфейс заклинания ловушки. Заклинание размещает на поле ловушку, если враг наступает на клетку с ловушкой, то ему наносится урон, и ловушка пропадает.

6. Создать класс вражеской башни. Вражеская башня размещается на поле, и если в радиусе ее атаки появляется игрок, то применяет ослабленную версию заклинания прямого урона. Не может применять заклинание несколько ходов подряд.

На 10/5/2 баллов:

7. Реализовать интерфейс заклинания призыва. Заклинание создает союзника рядом с игроком, который перемещается самостоятельно.

8. Реализовать интерфейс заклинание улучшения. Заклинание улучшает следующее используемое заклинание:

a) Заклинание прямого урона - увеличивает радиус применения
b) Заклинание урона по площади - увеличивает площадь
c) Заклинание ловушки - увеличивает урон
d) Заклинание призыва - призывает больше союзников
e) Заклинание улучшение - накапливает усиление, то есть при применении следующего заклинания отличного от улучшения, все улучшения применяются сразу

Примечания:

- Интерфейс заклинания должен быть унифицирован, чтобы их можно было единообразно использовать через интерфейс. Не должно быть методов в интерфейсе, которые не используются каким-то классом наследником.
- Избегайте явных проверок на тип данных.

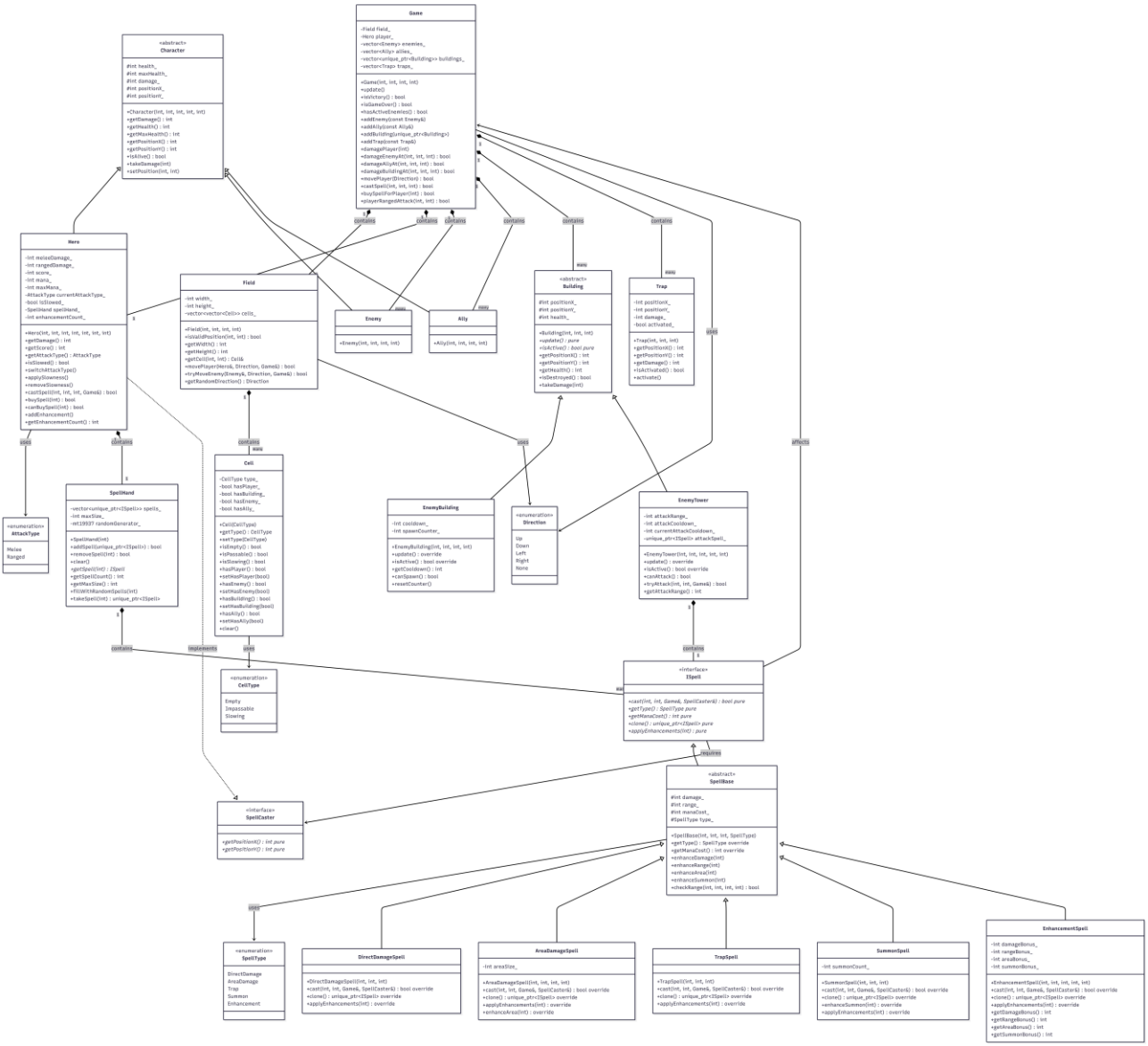
Выполнение работы.

Была реализована программа, содержащая все указанные в условие лабораторной работы классы и их поля и методы, а именно:

- Интерфейс карточки заклинания
- Класс руки персонажа
- Различные типы карт
- Класс вражеской башни

Архитектура программы.

В программе реализована иерархия классов, соответствующая принципам ООП.



Описание классов.

- SpellType

  Назначение: Определяет типы заклинаний в системе магии

  Значения:

  - DirectDamage - заклинание прямого урона
  - AreaDamage - заклинание урона по площади
  - Trap - заклинание установки ловушки
  - Summon - заклинание призыва союзников
  - Enhancement - заклинание улучшения

- Ally

Назначение: Класс союзных персонажей

Поля класса: Наследует все поля класса Character

Методы класса:

- Ally(int health, int maxHealth, int damage, int posX, int posY) - конструктор

- EnemyTower

Назначение: Башня с заклинаниями

Поля класса:

- attackRange_ - дальность атаки (private)
- attackCooldown_ - перезарядка атаки (private)
- currentAttackCooldown_ - текущая перезарядка (private)
- attackSpell_ - заклинание атаки (private)

Методы класса:

- EnemyTower(int x, int y, int health, int range, int cooldown) - конструктор
- update() - обновление состояния (атака игрока)
- isActive() - проверка активности башни
- canAttack() - проверка возможности атаки
- tryAttack(int targetX, int targetY, Game& game) - попытка атаки
- getAttackRange() - получение дальности атаки

- Trap

Назначение: Класс ловушки на поле

Поля класса:

- positionX_ - координата X (private)
- positionY_ - координата Y (private)
- damage_ - урон ловушки (private)
- activated_ - состояние активации (private)

Методы класса:

- Trap(int x, int y, int damage) - конструктор
- getPositionX() - получение координаты X

- getPositionY() - получение координаты Y
- getDamage() - получение урона
- isActivated() - проверка активации
- activate() - активация ловушки

- SpellCaster

Назначение: Интерфейс для объектов, способных применять заклинания

Методы класса:

- getPositionX() - метод получения координаты X
- getPositionY() - метод получения координаты Y

- ISpell

Назначение: Базовый интерфейс для всех заклинаний

Методы класса:

- cast(int targetX, int targetY, Game& game, SpellCaster& caster) - виртуальный метод применения заклинания
- getType() - виртуальный метод получения типа заклинания
- getManaCost() - виртуальный метод получения стоимости маны
- clone() - виртуальный метод клонирования
- applyEnhancements(int enhancementCount) - виртуальный метод применения улучшений

- SpellBase

Назначение: Базовая реализация общих свойств заклинаний

Поля класса:

- damage_ - урон заклинания (protected)
- range_ - дальность применения (protected)
- manaCost_ - стоимость маны (protected)
- type_ - тип заклинания (protected)

Методы класса:

- SpellBase(int damage, int range, int manaCost, SpellType type) - конструктор

- getType() - получение типа заклинания
- getManaCost() - получение стоимости маны
- enhanceDamage(int bonus) - улучшение урона
- enhanceRange(int bonus) - улучшение дальности
- enhanceArea(int bonus) - улучшение площади
- enhanceSummon(int bonus) - улучшение призыва
- checkRange(int casterX, int casterY, int targetX, int targetY) - проверка дальности

- DirectDamageSpell

Назначение: Заклинание прямого урона

Методы класса:

- DirectDamageSpell(int damage, int range, int manaCost) - конструктор
- cast(int targetX, int targetY, Game& game, SpellCaster& caster) - применение заклинания
- clone() - клонирование заклинания
- applyEnhancements(int enhancementCount) - применение улучшений

- AreaDamageSpell

Назначение: Заклинание урона по площади

Поля класса:

- areaSize_ - размер области поражения (private)

Методы класса:

- AreaDamageSpell(int damage, int range, int manaCost, int areaSize) - конструктор
- cast(int targetX, int targetY, Game& game, SpellCaster& caster) - применение заклинания
- clone() - клонирование заклинания
- applyEnhancements(int enhancementCount) - применение улучшений

- enhanceArea(int bonus) - улучшение площади поражения

- TrapSpell

Назначение: Заклинание установки ловушки

Методы класса:

- TrapSpell(int damage, int range, int manaCost) - конструктор
- cast(int targetX, int targetY, Game& game, SpellCaster& caster) - применение заклинания
- clone() - клонирование заклинания
- applyEnhancements(int enhancementCount) - применение улучшений

- SummonSpell

Назначение: Заклинание призыва союзников

Поля класса:

- summonCount_ - количество призываемых союзников (private)

Методы класса:

- SummonSpell(int damage, int range, int manaCost) - конструктор
- cast(int targetX, int targetY, Game& game, SpellCaster& caster) - применение заклинания
- clone() - клонирование заклинания
- enhanceSummon(int bonus) - улучшение призыва
- applyEnhancements(int enhancementCount) - применение улучшений

- EnhancementSpell

Назначение: Заклинание улучшения других заклинаний

Поля класса:

- damageBonus_ - бонус к урону (private)
- rangeBonus_ - бонус к дальности (private)
- areaBonus_ - бонус к площади (private)
- summonBonus_ - бонус к призыву (private)

Методы класса:

- EnhancementSpell(int manaCost, int damageBonus, int rangeBonus, int areaBonus, int summonBonus) - конструктор
- cast(int targetX, int targetY, Game& game, SpellCaster& caster) - применение заклинания
- clone() - клонирование заклинания
- applyEnhancements(int enhancementCount) - применение улучшений
- getDamageBonus() - получение бонуса урона
- getRangeBonus() - получение бонуса дальности
- getAreaBonus() - получение бонуса площади
- getSummonBonus() - получение бонуса призыва

- SpellHand

Назначение: Коллекция заклинаний игрока

Поля класса:

- spells_ - вектор заклинаний (private)
- maxSize_ - максимальный размер (private)
- randomGenerator_ - генератор случайных чисел (private)

Методы класса:

- SpellHand(int maxSize) - конструктор
- addSpell(unique_ptr<ISpell> spell) - добавление заклинания
- removeSpell(int index) - удаление заклинания
- clear() - очистка коллекции
- getSpell(int index) - получение заклинания по индексу
- getSpellCount() - получение количества заклинаний
- getMaxSize() - получение максимального размера
- fillWithRandomSpells(int count) - заполнение случайными заклинаниями
- takeSpell(int index) - извлечение заклинания

Выводы.

Была изучена парадигма объектно-ориентированного программирования. Была реализована программа на языке C++ содержащая основные классы игры с необходимыми полями и методами.