

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»
Тема: Создание классов заклинаний

Студентка гр. 4381

Гайнутдинова З. Р.

Преподаватель

Жангиров.Т.Р.

Санкт-Петербург

2025

Цель работы.

Создать классы заклинаний и класс хранения заклинаний.

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.

ОСНОВНАЯ СТРУКТУРА КЛАССОВ

Система заклинаний представляет собой иерархическую структуру классов, спроектированную для реализации магической механики в игре. Основой системы является абстрактный класс `Spell`, который определяет общий интерфейс для всех типов заклинаний. Данный класс содержит защищенные поля для хранения названия заклинания, его описания и стоимости маны, а также чисто виртуальные методы `cast()` и `requiresTarget()`. Метод `cast()` отвечает за применение заклинания к игровому полю и принимает параметры: ссылку на игровое поле, ссылку на игрока и координаты цели. Метод `requiresTarget()` возвращает булево значение, указывающее, требует ли данное заклинание выбора конкретной цели для применения.

Наследниками абстрактного класса `Spell` являются два конкретных класса заклинаний: `DirectDamageSpell` и `AreaDamageSpell`. Класс `DirectDamageSpell` реализует заклинания прямого урона, предназначенные для атаки одиночных целей. Он содержит поля для хранения величины урона, дальности применения и флага усиления заклинания. Конструктор класса позволяет задавать параметры урона и дальности, а также наследует базовые параметры названия, описания и стоимости маны. Метод `cast()` в этом классе проверяет возможность применения заклинания (наличие достаточного количества маны у игрока и допустимость расстояния до цели), после чего наносит урон врагу в указанных координатах.

Класс `AreaDamageSpell` реализует заклинания площадного поражения, которые наносят урон всем врагам в определенной области. Помимо параметров урона и дальности, данный класс содержит поле `areaSize`, определяющее размер области поражения. Метод `cast()` в этом классе выполняет проверки аналогично прямому урону, но затем вычисляет все клетки в заданном радиусе от точки применения и наносит урон всем врагам в этой области. Оба класса содержат метод `enhance()`, позволяющий усиливать параметры заклинания, однако конкретная реализация усиления отличается: для прямого урона увеличивается `damage` и `range`, а для площадного - `damage` и `areaSize`.

Для управления коллекцией заклинаний игрока реализован класс `SpellHand`, который выполняет функции инвентаря заклинаний. Класс содержит вектор умных указателей на объекты `Spell` и ограничение на максимальное количество заклинаний. Основные методы класса включают: `addSpell()` для добавления нового заклинания в коллекцию (с проверкой на превышение лимита), `removeSpell()` для удаления заклинания по индексу, `getSpell()` для получения доступа к конкретному заклинанию, а также вспомогательные

методы для получения текущего количества заклинаний и проверки заполненности коллекции.

За координацию работы с заклинаниями отвечает служебный класс `SpellCasting`, содержащий исключительно статические методы. Ключевой метод `castSpell()` инкапсулирует всю логику применения заклинания: проверку наличия заклинания в коллекции игрока, проверку достаточности маны, применение эффекта заклинания и вычитание соответствующего количества маны у игрока. Дополнительные методы класса предоставляют функциональность для получения случайного стартового заклинания, проверки наличия заклинаний у игрока, отображения информации о заклинаниях и мане, а также обработки покупки новых заклинаний.

Интеграция системы заклинаний с другими компонентами игры осуществляется через класс `Player`, который содержит экземпляр `SpellHand` и поля для хранения текущего и максимального запаса маны. Класс `Player` предоставляет методы для управления маной (установка значения, восстановление), проверки возможности произнесения конкретного заклинания, а также методы для применения заклинаний через делегирование к `SpellCasting`. Такая архитектура обеспечивает четкое разделение ответственности: `Player` отвечает за хранение состояния, `SpellHand` - за управление коллекцией заклинаний, а `SpellCasting` - за логику их применения.

ВЗАИМОДЕЙСТВИЕ КЛАССОВ

Взаимодействие классов в системе заклинаний построено по принципу разделения ответственности, где каждый класс выполняет свою четкую роль. Процесс начинается с игрока (`Player`), который является центральной фигурой в системе заклинаний. У игрока есть два ключевых компонента: коллекция заклинаний в виде `SpellHand` и запас маны. Когда игрок решает применить заклинание, он обращается к своему инвентарю `SpellHand`, который работает

как умный контейнер - он не только хранит заклинания, но и следит за тем, чтобы их количество не превышало лимит, а также предоставляет доступ к конкретным заклинаниям по индексу.

После выбора заклинания управление передается служебному классу SpellCasting, который действует как координатор всего процесса применения магии. Этот класс проверяет основные условия: достаточно ли у игрока маны для каста, существует ли выбранное заклинание, и требует ли оно цель. Если все проверки пройдены, SpellCasting запускает механизм применения заклинания, обращаясь к конкретной реализации выбранного заклинания.

На этом этапе в работу вступают конкретные классы заклинаний - DirectDamageSpell для точечных атак или AreaDamageSpell для площадных воздействий. Каждое заклинание имеет свою логику применения. DirectDamageSpell вычисляет расстояние до цели и если оно в пределах досягаемости, наносит урон конкретному врагу. AreaDamageSpell работает сложнее - он определяет зону поражения вокруг выбранной точки и наносит урон всем врагам в этой области. Оба типа заклинаний после применения сообщают о результате операции - успешно ли было применено заклинание.

Важную роль в этом процессе играет игровое поле (GameField), которое предоставляет информацию о расположении всех объектов. Через GameField заклинания получают доступ к данным о врагах, их позициях и состоянии. После применения заклинания GameField обновляет состояние пораженных врагов - уменьшает их здоровье или удаляет их, если они погибли.

Завершающий этап - обновление состояния игрока. Если заклинание было успешно применено, игрок теряет соответствующее количество маны. Весь этот процесс обеспечивает гибкую и расширяемую систему, где новые заклинания можно легко добавлять, просто создавая новые классы-наследники от базового

Spell, не меняя при этом основную логику взаимодействия между компонентами системы.

ИГРОВОЙ ПРОЦЕСС

Игровой процесс представляет собой пошаговую стратегию на клеточном поле, где игрок управляет персонажем, сражаясь с врагами и используя магические способности. В начале игры игрок размещается на поле и получает стартовое заклинание. Каждый ход игрок может выполнить одно основное действие: переместиться на соседнюю клетку, атаковать врага в ближнем или дальнем бою, либо применить заклинание из своей коллекции. Система заклинаний предлагает два основных типа - заклинания прямого урона для точечной атаки отдельных врагов и заклинания площадного поражения для удара по группе противников. После хода игрока наступает очередь врагов - они перемещаются по полю, пытаясь приблизиться к игроку и атаковать его. Особенностью игры являются вражеские здания, которые периодически порождают новых противников, что требует от игрока стратегического подхода - либо уничтожать здания, либо быстро устранять появляющихся врагов. По мере победы над врагами игрок зарабатывает очки, которые можно тратить на покупку новых заклинаний или улучшение существующих. Игра продолжается до тех пор, пока игрок не погибнет от атак врагов или не выполнит поставленные цели.

UML диаграмма: