**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**

Студент гр. 4381 _____ Ишамчурин Д.И.

Преподаватель _____ Жангиров.Т.Р.

Санкт-Петербург

2026

**Цель работы.**

Расширить функциональность игры, добавив систему заклинаний с интерфейсами для прямого урона и урона по площади, а также реализовать руку игрока с ограниченным количеством карт.

**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.

**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**

Во второй лабораторной работе программа получила серьезное расширение за счет добавления системы заклинаний. Появился новый абстрактный класс Spell, который стал базовым для всех заклинаний в игре. В нем хранятся название заклинания, стоимость использования и радиус действия, а также объявлен чисто виртуальный метод cast, который отвечает за применение заклинания. Этот метод принимает игровое поле, игрока и координаты цели, что позволяет единообразно работать с разными типами заклинаний.

От него отпочковались два конкретных класса заклинаний. DirectDamageSpell добавляет величину урона и при применении ищет врага на указанной клетке если цель в радиусе действия, то враг получает урон. AreaDamageSpell тоже хранит урон, но вдобавок имеет размер области поражения, при применении оно проходит по всем клеткам в этой области и наносит урон каждому врагу, который там окажется, даже если область пустая, заклинание все равно срабатывает.

Для управления заклинаниями появился класс PlayerHand, то есть рука игрока. Внутри у него вектор уникальных указателей на заклинания, благодаря чему можно хранить разные типы заклинаний в одном месте. Рука имеет ограничение по размеру, которое задается через конструктор, также там считается количество убитых врагов, чтобы со временем выдавать игроку новые заклинания. Через методы можно добавить заклинание, применить его по индексу и удалить после использования, показать список доступных заклинаний на экране.

Еще добавился класс GameController, который теперь управляет всей игрой. В нем собраны вместе игровое поле, игрок и рука с заклинаниями, а также основной игровой цикл. Конструктор настраивает начальное состояние, размещает игрока и врагов, выдает первое случайное заклинание. Дальше в методе run запускается цикл, где на каждом ходу отображается поле и список

заклинаний, обрабатывается ввод пользователя либо перемещение, либо применение заклинания с запросом координат, а потом ходят враги. В конце каждого хода проверяется, не умер ли игрок и не убиты ли все враги.

Класс GameField тоже немного доработали, добавили методы, чтобы получать доступ к ширине и высоте поля, к списку врагов, к конкретной клетке по координатам. Это понадобилось для того, чтобы заклинания могли взаимодействовать с полем и находить врагов. Также появился метод removeDeadEnemies, который убирает с поля мертвых врагов после применения заклинаний.

Для удобства создания заклинаний сделали отдельный класс SpellFactory с набором статических методов, которые возвращают готовые заклинания, например, файербол или взрыв. Это позволяет не разбрасывать создание объектов по разным местам программы.

Ну и main.cpp теперь совсем короткий просто создает контроллер и запускает игру, вся логика убрана в отдельные классы.

**ВЗАИМОДЕЙСТВИЕ КЛАССОВ**

Взаимодействие классов во второй лабораторной работе строится вокруг новой системы заклинаний, которая тесно вплетается в существующую архитектуру. Главным координатором выступает класс GameController, который содержит в себе игровое поле, игрока и руку с заклинаниями. Когда игрок решает применить заклинание, он вводит соответствующую цифру, и GameController обращается к классу PlayerHand с просьбой использовать заклинание под указанным индексом. PlayerHand берет нужное заклинание из своего вектора и вызывает его виртуальный метод cast, при этом передавая туда ссылки на игровое поле, игрока и координаты цели, которые предварительно запросил контроллер. В зависимости от того, какой именно объект хранится в руке DirectDamageSpell или AreaDamageSpell, вызывается соответствующая реализация метода cast. Внутри этого метода заклинание обращается к игровому полю через его новые геттеры, чтобы получить доступ к вектору врагов, и проходит по врагам, проверяя их координаты. Если враг попадает под

действие заклинания, ему наносится урон через его метод takeDamage. После применения заклинания PlayerHand удаляет его из руки. Когда враг погибает, GameField при следующем вызове removeDeadEnemies убирает его с поля, а GameController через руку вызывает метод onEnemyKilled, который увеличивает счетчик убийств и при достижении порога может добавить новое заклинание через SpellFactory. Таким образом, все классы работают в связке: контроллер управляет процессом, рука хранит и выдает заклинания, заклинания взаимодействуют с полем и врагами, а фабрика создает новые экземпляры по мере необходимости.

## ИГРОВОЙ ПРОЦЕСС

Игровой процесс после добавления системы заклинаний стал более разнообразным и теперь включает не только перемещение и атаку при столкновении, но и возможность применять магические способности. В начале игры у игрока в руке уже есть одно случайное заклинание, например файербол для прямого урона или взрыв по области. На каждом ходу игрок видит на экране не только игровое поле с противниками и свое здоровье, но и список доступных заклинаний с номерами для выбора. Теперь у игрока есть выбор либо переместиться по полю с помощью клавиш WASD, либо применить одно из заклинаний, нажав соответствующую цифру. Если игрок решает использовать заклинание, программа запрашивает координаты цели, и после ввода проверяет, находится ли цель в радиусе действия. Для заклинания прямого урона ищется враг на указанной клетке, и если он там есть, ему наносится урон. Для заклинания по области урон наносится всем врагам в квадрате два на два клетки вокруг указанной точки, даже если там никого нет, заклинание все равно считается использованным. После успешного применения заклинание исчезает из руки, и на его место со временем можно получить новое за убийство определенного количества врагов. После хода игрока, будь то перемещение или заклинание, наступает ход врагов они двигаются случайным образом и атакуют игрока, если оказываются рядом. Игра продолжается до тех пор, пока не будут убиты все враги или пока не погибнет игрок, при этом

убитые враги считаются, чтобы в нужный момент наградить игрока новым заклинанием.

**UML диаграмма:**