ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
ФГАОУ ВО НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук
Образовательная программа «Прикладная математика и информатика»

Отчет о программном проекте на тему:

Игра на Unity

Выполнил студент:

группы БПМИ222 Круглов Андрей Юрьевич

Принял руководитель проекта:

Макаров Сергей Львович
Кандидат технических наук
Доцент ДПИ ФКН НИУ ВШЭ

Москва 2024

Аннотация

В этом проекте я разрабатываю игру “NeonBlock”, в которой совмещены жанры Тетрис и Roguelike. Игрок собирает набор из фигур с разными эффектами, которые потом используются в уровнях, а также различные полезные улучшения.

На данный момент есть рабочая версия игры с оформленным дизайном и большим числом различных фигур, уровней и реликвий. В ближайшие сроки я планирую полностью доделать игру, чтобы выложить её в открытый доступ на https://itch.io

Ключевые слова

Программирование, Игра, Unity, Тетрис, Roguelike, аркада, пиксель-арт, кибер-панк

Основные термины

Рогалик, Roguelike - жанр игр, предусматривающий случайную генерацию уровней и перманентную смерть, то есть при проигрыше игрок теряет весь прогресс.

Геймплей - игровой процесс

Игровой баланс - условное равновесие между игровыми механиками, никакие предметы, улучшения или навыки не дают значительное преимущество по сравнению с другими.

Блок - одна игровая клетка, каждый блок содержит отдельные свойства.

Фигура - объединение из нескольких блоков, которое может перемещать игрок.

Стакан - прямоугольное игровое поле шириной 10 блоков.

Очистка линии - Если в стакане есть строки, заполненные блоками, они очищаются.

Удаление линии - Если блоки находятся выше красной линии, несколько нижних линий удаляются, а игрок получает урон, см. раздел 3.3.

Пиксель - В данной игре является основной валютой, которую можно потратить в магазине на новые фигуры.

Реликвия - Постоянное улучшение, которое увеличивает какие-то показатели или меняет игровые свойства. Реликвии могут хорошо сочетаться с некоторыми блоками или с другими реликвиями.

Содержание

1 Введение

Roguelike - один из наиболее популярных жанров игр. Он состоит из двух основных частей: уровни в игре генерируется случайно, а игрок после смерти начинает игру сначала. Благодаря этому каждое прохождение становится уникальным и запоминающимся.

Тетрис - известная компьютерная игра 1984 года, в которой игрок расставляет падающие фигуры, чтобы заполнять линии. Эта игра стала знаменитой по всему миру, и сейчас у неё большое сообщество игроков. Она остаётся популярной благодаря международным соревнованиям и современным версиям игры.

Мой проект заключается в разработке игры на Unity и её последующем релизе. Я решил сделать игру, совмещающую жанр roguelike с Тетрисом. Позже было решено выполнить игру в пиксельном дизайне и в сеттинге киберпанка. Игра получила название "NeonBlock"(см. Рисунок 1). В этой игре нужно собирать различные фигуры с необычными свойствам, затем использовать их на уровнях, чтобы пройти как можно дальше и одолеть босса. Также игрок будет получать различные реликвии, которые могут повлиять на игру.

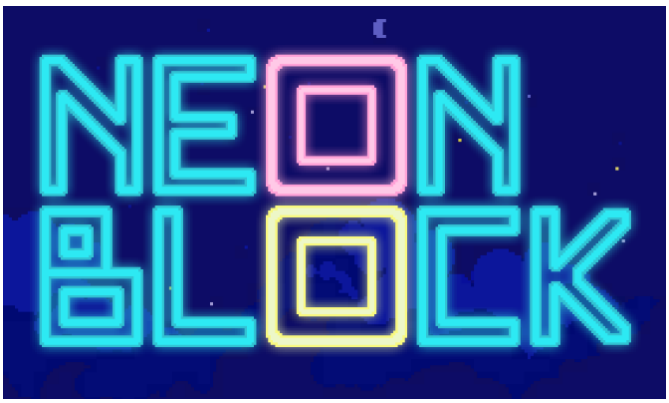

Рис. 1: Логотип игры в главном меню

На каждом уровне для победы надо набрать определённое количество энергии. Требуемое значение увеличивается с каждым уровнем. Чтобы получать энергию, надо очищать линии в стакане. Очистка нескольких линий за раз даёт больше энергии. Также энергию можно получить при активации некоторых блоков или реликвий. В середине стакана находится лазер, если поставить блоки выше него, то часть блоков исчезнет, чтобы освободить место, а игрок получит урон. Если здоровье упадёт до нуля, игра закончится, а прохождение придётся начать заново.

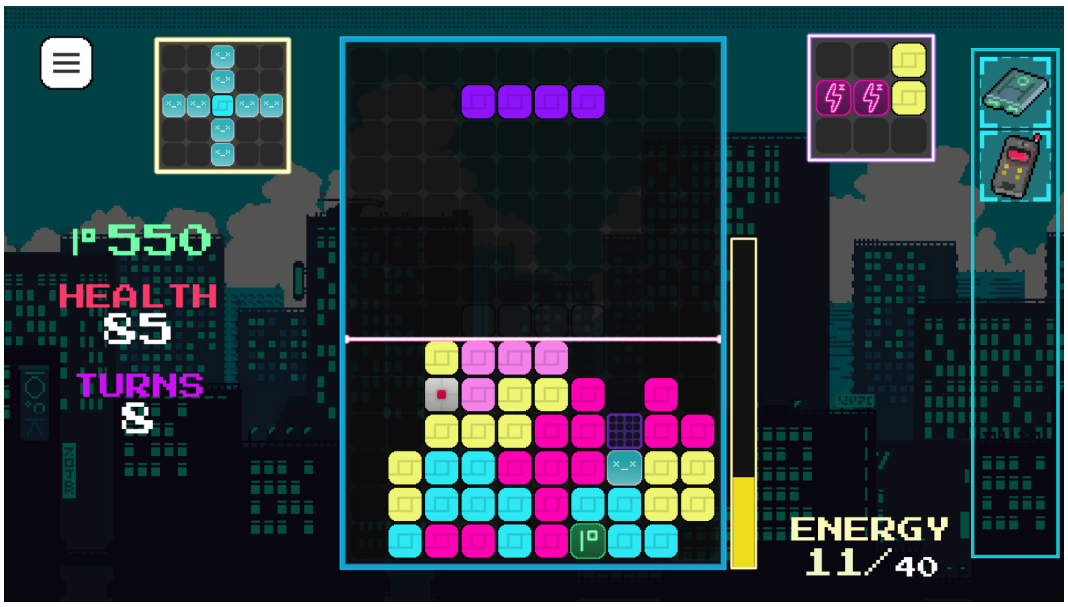

Рис. 2: Скриншот игрового процесса

Каждая фигура состоит из нескольких блоков. Блоки могут быть разного вида, каждый вид блоков обладает уникальными свойствами. На текущий момент в игре есть более 30 видов блоков. Эффекты блоков активируются при разных обстоятельствах в зависимости от типа блока, например при установке фигуры, при очистке линии или в конце хода. Взаимодействие всех блоков начинается после каждой установки фигуры и происходит по определённому алгоритму (см. раздел 4.2).

Сама игра делится на уровни, которые отображаются на карте. При передвижении по карте, игрок может выбирать направление и следующее событие, например уровень или магазин.

2 Анализ решений

Создавая игру, я опирался на многочисленные игры жанра roguelike: Peglin[1], Slay the Spire[2], Hades[3]. Сама идея совместить жанр с другой известной игрой появилась из игры Dungeons & Degenerate Gamblers[4], в которой совместили roguelike с блэкджеком.

Также я ориентировался на современные аналоги Тетриса, например Tetris® Effect: Connected[5]. Они отличаются от старых версий, в них есть много новых механик, такие как удержание фигуры, Lock delay и SRS[6] (Super Rotation System). От реализации SRS я решил отказаться, поскольку она требует определённую таблицу смещений для каждой фигуры и её восьми возможных поворотов, и её очень сложно сделать интуитивной для всех необычных вариантов фигур. Однако фигуры будут смещаться от стен при повороте. Описание реализации этих механик можно найти в спецификации The Tetris Company (компания с правами на Тетрис) - Tetris Guideline[7].

Игр, совмещающих эти два жанра, немного, одна из них - InfidHells[8]. В ней есть несколько персонажей с уникальными способностями и интересные механики противников, которые усложняют прохождение. Однако в игре по большей части используются только стандартные фигуры, а необычные механики редко взаимодействуют друг с другом. Геймплей в основном строится на быстрой очистке стакана и использовании способности персонажа и предметов.

Также есть игры, где механика Тетриса используется для построения уровня, например 4-Block Dungeon[9], в которой все тетрамино являются коридорами с монстрами, ловушками и предметами. Игрок строит подземелье из фигур, по которому перемещается герой. Чтобы пройти уровень, надо убить несколько монстров, очистить несколько линий и добраться до выхода. Такие игры отличаются от обычного тетриса, ограничиваясь лишь установкой коридоров из клеток на поле, а основной геймплей строится на исследовании подземелья.

Моя игра отличается от существующих, поскольку она содержит в себе механику колодостроя (вместо карт в игре используются фигуры), игрок должен собрать оптимальный набор из фигур и улучшений, сочетающихся друг с другом. Также в игре есть огромное разнообразие механик блоков и их взаимодействий между собой.

3 Игровой процесс

3.1 Карта

Игра начинается на глобальной карте, на которой можно выбрать направление движения. Каждая точка на карте представляет собой некоторое событие:

Уровень - обычный уровень, после прохождения игрок выбирает одну дополнительную фигуру и возвращается к карте.

Сложный уровень - или минибосс, или эпический уровень. Сложнее обычного уровня, но в конце игрок получит одну реликвию на выбор.

Босс - последний уровень, очень сложный и содержит необычные механики. После прохождения игра завершается.

Магазин - в магазине игрок может потратить пиксели, чтобы купить новые фигуры или восстановить здоровье.

Случайное событие - тут может быть несколько разных событий, например уровень, выбор новой фигуры или удаление фигуры из инвентаря.

События в точках выбираются случайно, хотя в некоторых точках я его сделал фиксированным (пустое событие на старте, обе следующих точки всегда являются уровнями, в конце босс и в середине несколько сложных уровней). Игрок обязан двигаться слева направо, в сторону финиша, и не может возвращаться на предыдущие точки.

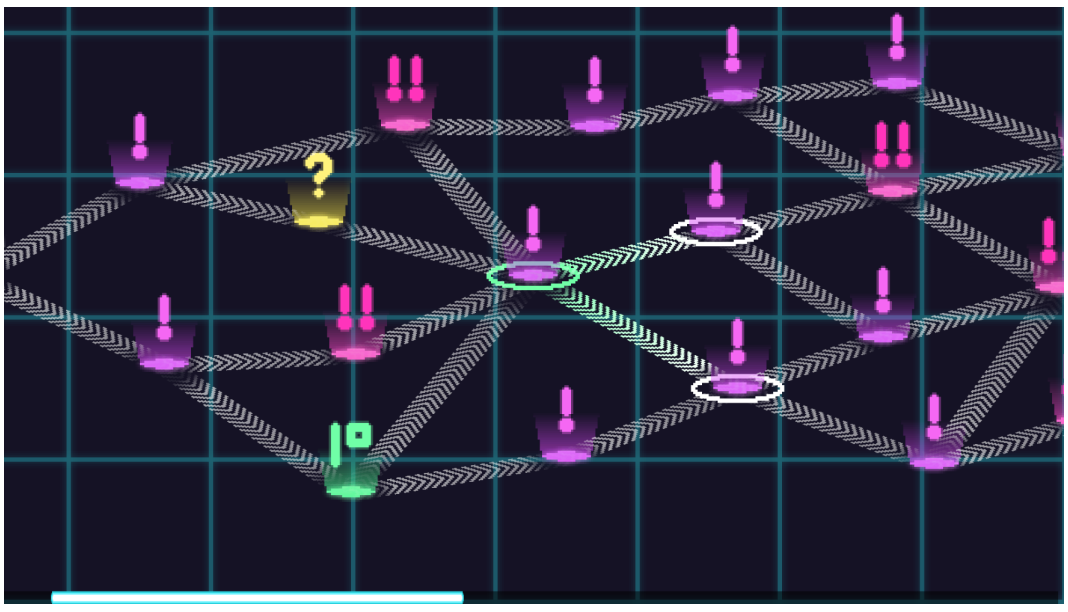

Рис. 3: Глобальная карта

3.2 Магазин, фигуры

При прохождении уровней игрок будет получать внутриигровую валюту, называемую пикселями. Эту валюту можно обменять в магазине (см. Рисунок 4) на восстановление здоровья или новую фигуру. На выбор даётся несколько фигур, можно купить больше одной. Также можно обновить выборку фигур за небольшую плату.

В игре есть очень много фигур, они состоят из разных блоков и имеют разную форму. Также фигуры в игре обладают различной редкостью, от обычной до легендарной. Чем выше редкость, тем удобнее форма фигуры и полезнее свойства её блоков, но у редких фигур выше стоимость. Собранные фигуры можно посмотреть в инвентаре, в меню паузы.

Рис. 4: Магазин

3.3 Уровень

Уровень представляет собой матрицу (стакан) шириной 10 блоков. Во время игры сверху будут падать фигуры, которые игрок может двигать и поворачивать. Нужно ставить фигуры так, чтобы заполнить горизонтальные линии. Такие линии очистятся и дадут энергию. Чтобы пройти уровень, нужно набрать определённое количество энергии, которое возрастает с каждым уровнем. Энергию также можно получить из некоторых блоков и реликвий.

В середине стакана расположен горизонтальный лазер (высота настраивается). Если блоки находятся над этой линией, игра удалит несколько линий так, чтобы блоки вмещались под лазером (см. Рисунок 5). При этом игрок получит урон, пропорциональный количеству

удалённых линий. Если здоровье опустится до 0, игрок проиграет. Каждые несколько ходов лазер будет смещаться на 1 блок вниз, поэтому игрок не может долго задерживаться на уровне и должен находить способы быстрее получать энергию.

Во время игры сбоку отображаются следующие фигуры (по умолчанию одна) и текущая удерживаемая фигура.

Перед началом уровня игрок может посмотреть на его конфигурацию. В уровне могут быть установлены блоки, лазер может быть смещён, а также уровень может содержать дополнительные фигуры, которые добавятся у инвентарю игрока до конца уровня.

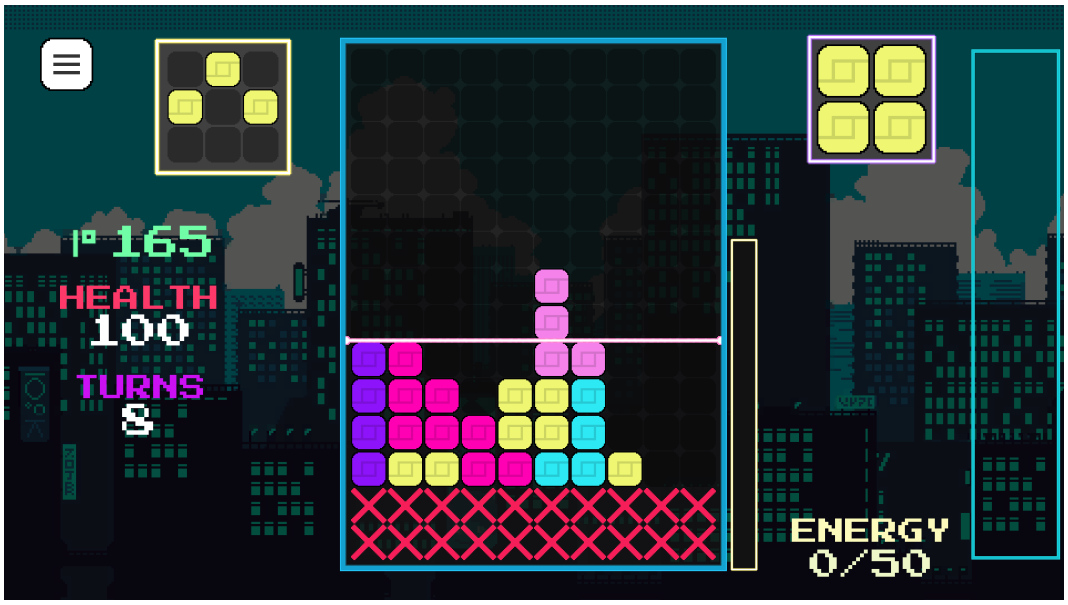

Рис. 5: Блоки установлены выше линии на два уровня, удаляется две нижних линии

3.4 Блоки

На данный момент в игре есть около 30 видов (типов) блоков. Некоторые из них обладают полезными свойствами, другие мешают игроку. У каждого вида блока есть название описание, которое отображается при наведении мышкой на блок. Также есть несколько специальных блоков, используемых в коде игры, которые не содержатся в фигурах. Ниже представлены некоторые виды блоков:

- Пустой
- Обычный
- Монета - даёт 5 монеты *при очистке*
- Энергия - даёт 5 энергии врагу *при очистке*

- Бомба - взрывает блоки в квадрате 3 на 3 *при очистке*
- Огненный - наносит 4 урона игроку *при очистке*.
- Песок - блок падает вниз *всегда*.
- Куст - при установке разрастается в 4 соседних клетки, если они пустые.
- Металл - нельзя очистить или взорвать
- Блок с цепями - фигуру с таким блоком нельзя вращать.
- Призрак - нетвёрдый блок, по свойствам похож на пустой блок, но учитывается при проверке готовых линий. Можно ставить в другие блоки или наоборот, при этом призрачный блок удаляется. Например, в него может упасть песок.
- Майнер - генерирует 1 пиксель *каждый ход*
- Замок - не даёт очищать линию, в которой находится. Эффект отменяется, если в линии есть блок типа "ключ".
- Временная энергия - даёт 5 энергии *при очистке*, *в конце хода* превращается в обычный блок.
- Вирус - заполняет вирусом закрытые области из пустых блоков *всегда*. Если область открыта (касается потолка), то она становится пустой.

3.5 Реликвии

Реликвии являются постоянными улучшениями, которые игрок может получить за прохождение сложных уровней. Некоторые реликвии изменяют какие-то параметры, например получаемую энергию или скорость фигур. Другие меняют игровые свойства, например позволяют видеть больше следующих фигур или влияют на эффекты определённых блоков. Игрок может получить больше одной копии реликвии, но для каждой из может стоять ограничение на максимальное количество. Вот некоторые из них:

- Батарея - немного больше энергии.
- Промокод - уменьшает стоимость фигур в магазине.
- Телескоп - позволяет видеть больше фигур в очереди.

- Урна - если фигура удерживается 5 ходов, она удаляется.
- Слот для батареи - гораздо больше энергии, но нельзя поменять удерживаемую фигуру.
- Транзистор - слегка больше энергии за очистку линий
- Лазерный модуль - при переполнении линии удаляются сверху, над лазером.
- Детонатор - даёт 1 энергии за каждый взорванный блок.
- Wi-fi - увеличивает получаемую энергию за каждый непустой блок в стакане.
- Таймер - увеличивает число ходов до понижения лазера.

Информацию о блоке или о реликвии можно узнать, если навестись на них указателем мыши (см. Рисунок 6). Эта функция отключена во время прохождения уровня, чтобы не отвлекать игрока.

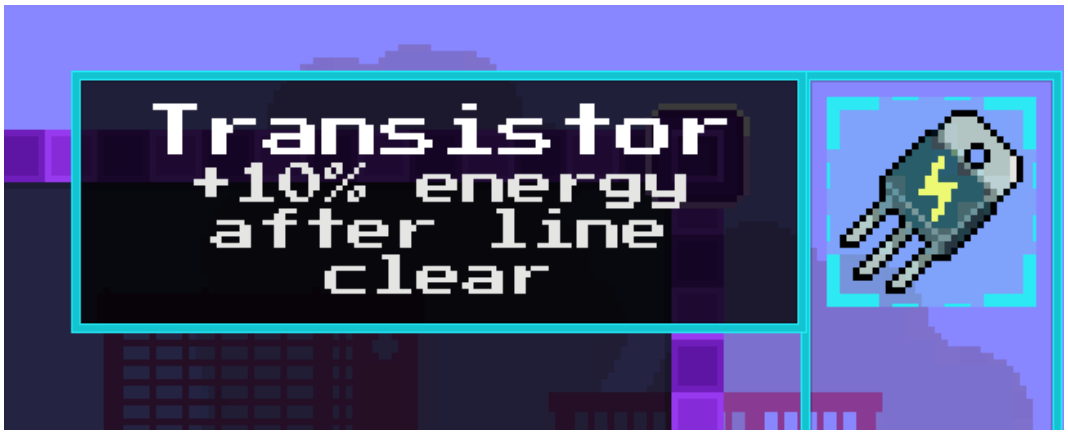

Рис. 6: Всплывающая подсказка для реликвии

4 Структура проекта

4.1 Управление

Создавая управление, я ориентировался на Tetris Guideline[7]. За управление фигурой отвечает класс Controller. Он принимает нажатия клавиш и вызывает нужные функции в других классах.

Игрок может двигать фигуру вниз, влево и вправо, поворачивать её или моментально бросить вниз. При вращении около стены фигура автоматически выталкивается от стены для удобства игрока. Это не получилось применить в общем случае, так как система смещений описана только для фигур из 4 блоков, для всех остальных форм она была бы сложной и не интуитивной для игрока.

Когда фигура касается блоков, запускается таймер в 0.5 сек., по истечении которого фигура фиксируется. Игрок может подвинуть или повернуть фигуру, тогда таймер сбросится, это позволяет лучше контролировать фигуру в сложных ситуациях. Таймер сбрасывается ограниченное число раз, после этого фигура в любом случае фиксируется.

4.2 Взаимодействие блоков

Блоки начинают взаимодействие, когда игрок ставит фигуру. Программа проверяет определённые условия и активирует нужные блоки. В процессе активации у игрока отображается анимация в виде небольшой задержки между взаимодействиями (в дальнейших версиях можно добавить более сложные анимации). Есть несколько вариантов активации: при установке, при очистке, всегда, в конце хода, каждый ход. Взаимодействие происходит в порядке, показанном на схеме (см. Рисунок 7). После каждого этапа стакан обновляется (обрабатываются блоки с условием активации “всегда”).

При каждой проверке все активированные блоки сортируются по приоритету, затем по типу и по координате. Например, взрывающиеся блоки активируются позже остальных. Все взаимодействия вызываются в зависимости от вида блока (см. раздел 3.4).

Энергия за очистку считается в зависимости от числа очищенных линий и некоторых блоков и реликвий.

При переполнении игрок получит урон за каждую удалённую линию. Эти линии не считаются за очищенные. За некоторые виды блоков при удалении наносится дополнительный урон.

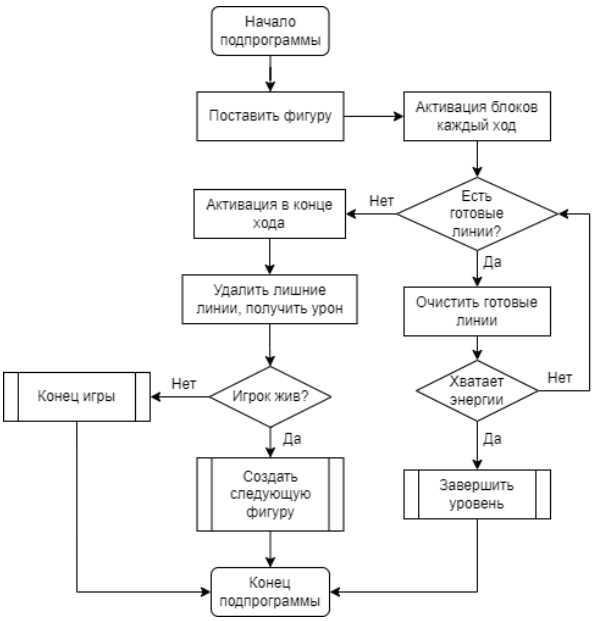

Рис. 7: Схема взаимодействия блоков

4.3 Хранение данных о видах блоков

Класс BlockDataManager является синглтоном и хранится в одном экземпляре. В него вносятся данные о каждом виде блока (см. рис. 8). Для этого я использовал сложную структуру, в которой для каждого ключа (типа блока) хранится название, описание, изображение, приоритет, и фиксированность цвета. Приоритет используется при активации блоков, блоки без фиксированного цвета окрашиваются в цвет фигуры, название и описание отображается при наведении мышкой на блок. Затем эти данные перемещаются в отдельные массивы и другие скрипты могут получить нужную информацию о типе блока, обратившись к статической переменной класса.

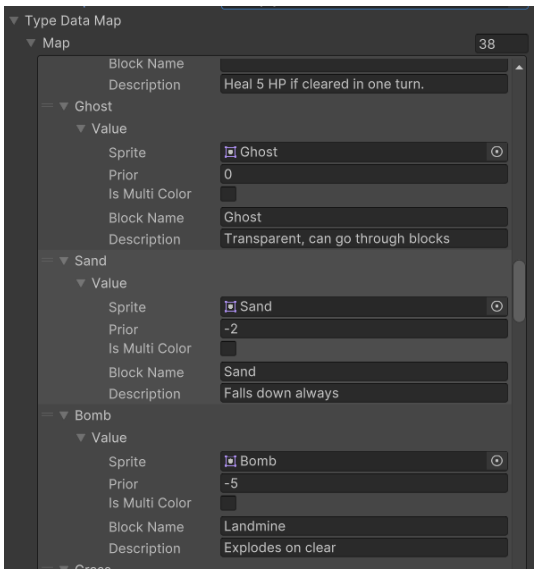

Рис. 8: Компонент BlockDataManager.

Такой подход позволяет легко добавлять и изменять блоки, изменяя значения только в префабе с классом BlockDataManager. Также этот класс отвечает за изменение типа блока у игровых объектов-блоков, за установку нового изображения и цвета.

4.4 Хранение фигур

Все фигуры в игре хранятся в классе PieceData, который тоже является синглтоном. Класс содержит массив из сотни структур, в каждой хранятся сама фигура, цвет и редкость. Для удобства фигуры добавляются через специальный конструктор, переменные можно вводить в любом порядке. Для самой фигуры используется дополнительная функция, преобразующая строки в матрицу блоков. С помощью строк можно намного быстрее изменять фигуры и добавлять новые. Для некоторых типов выделены специальные символы, остальные можно ввести с помощью цифр или символа * (равносильно 0). Каждой цифре можно сопоставить некоторый тип блока.

4.5 Конфигурация уровней

Уровни хранятся в классе LevelDataManager. При генерации уровня выбирается одна из множества конфигураций. Для боссов и сложных уровней есть отделные конфигурации. Уровень хранит в себе расположение блоков, высоту лазера, набор дополнительных фигур и дополнительные параметры (например, в некоторых уровнях линии по умолчанию удаляются сверху). Расположение блоков в стакане и дополнительные фигуры вводятся аналогично классу PieceData, через строки (см. Рисунок 9).

```
data.Add(new LevelData()
{
    glass = ReadLevel(new List<string> {
        ".#.#.#.#.#",
        "#.#.#.#.#.",
        ".#.#.#.#.#",
        "*.0.0.*.*.",
    }, new List<Block.Type>
    {
        Block.Type.Protector,
    }),
    laser = 8,
    task_extra = 80,
    turns_extra = 3,
    type = LevelType.boss,
});
```

Рис. 9: Пример уровня в LevelDataManager.

5 Дизайн

5.1 Графика

После завершения первого отчёта, я начала работать над графической составляющей игры. Я решил выполнить дизайн в стиле киберпанк, в пиксельной рисовке. Изображения создавались в программе для пиксель-артов - Aseprite[10]. Для проекта я выбрал определённую цветовую палитру и придерживался выбранных оттенков.

Некоторые ассеты были взяты из бесплатных источников (задний фон, некоторые реликвии, шрифт), но основная часть была сделана самостоятельно (блоки, элементы интерфейса, логотип).

Создание пиксельного дизайна имеет свои трудности. Одна из них - размытие изображения. Чтобы его избежать, все пиксельные спрайты должны отрисовываться точечным фильтром, желательно также правильно масштабировать изображения, чтобы пиксели не растягивались. Например, в игре исходные изображения блоков имеют размер 32 на 32, и почти всегда ширина объектов-блоков кратна 32 (кроме крупных фигур). Изображения расположены правильно в стандартном разрешении 1920 на 1080, но в других они слегка размыты, хотя при фиксированном соотношении сторон (16:9) интерфейс сохраняет исходное очертание.

Я создал несколько материалов, которые меняют оттенок изображения на определённое значение. Такой подход позволяет менять цвет изображения без умножения, не затемняя его светлые участки, как при стандартном изменении цвета. Также я реализовал простой шейдер, который генерирует анимированную линию. Эти линии используются для соединения точек на карте.

Рис. 10: Создание логотипа в Aseprite.

5.2 Аудио

В игре есть множество разных звуковых эффектов и несколько музыкальных треков. Звуки помогают передать взаимодействие пользователя с игрой, а музыка задаёт настроение. В игре заготовлено несколько музыкальных треков под разные ситуации, например для битвы с боссом или для главного меню. Звуки проигрываются с помощью специального класса, в которм указаны звуковые файлы. Звук воспроизводится через несколько микшеров (звуковых каналов), для музыки и эффектов есть разные микшеры. Громкость можно изменить в главном меню, в настройках (см. Рисунок 11). Параметры сохраняются при закрытии игры.

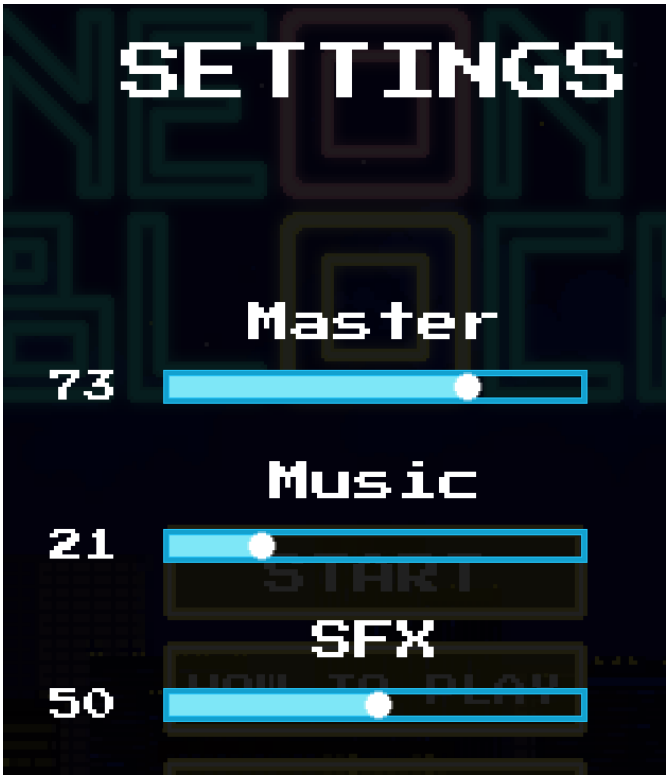

Рис. 11: Настройка громкости.

6 Тестирование

Поскольку самостоятельно бывает трудно понять, насколько интересная игра, и тяжело найти некоторые баги, я провёл небольшое тестирование игры, плейтест. Для этого в январе я разослал билд с прототипом игры нескольким знакомым. Полученные отзывы уже помогли улучшить игру. Например, некоторые игроки просили добавить возможность лечения в магазине. Ещё мне советовали улучшить управление фигурой, теперь в игре есть возможность вращать фигуру рядом со стеной и задержка перед установкой фигуры.

В тестовой версии присутствует опция тестового режима, которая делает видимыми некоторые скрытые элементы игры. Так можно включить определённые кнопки для игрока, например переход между уровнями, добавление монет или перевыбор блоков в магазине. Эта опция нужна для упрощённого тестирования, чтобы не нужно было много раз проходить уровни для проверки механики. Опция реализована с помощью класса TestModeObjectManager, он хранит список скрытых объектов и включает/отключает их при вызове определённого события.

Перед отправкой игры в интернет я планирую провести ещё одно тестирование, чтобы улучшить потенциальные недочёты с интерфейсом, дизайном или игровым балансом. После релиза игры можно продолжить тестирование, собирая информацию о прохождении через специальные аналитические программы, или опрашивать игроков через google-формы.

7 Заключение

За долгое время разработки мне удалось создать интересную и уникальную игру, у которой на данный момент нет аналогов. В ближайшее время игра будет готова к релизу.

Ссылка на репозиторий с игрой: https://github.com/Andrei-23/NeonBlock

После первой контрольной точки была реализована карта, на которой игрок перемещается по разным уровням и событиям, были добавлены реликвии и боссы, обновлён интерфейс. Теперь игрок может сразу получить информацию о блоке или реликвии.

В проекте есть 70 скриптов, которые суммарно содержат примерно 8000 строк кода. На данный момент в игре есть 34 вида блоков и 23 реликвии, каждое прохождение получается уникальным. Игрок может собрать разные интересные комбинации.

Мне удалось создать красивый дизайн в стиле пиксельного киберпанка. Было создано много пиксель-артов для проекта. Игра выглядит и звучит приятно.

Сейчас планируется доработать игру: добавить локализацию, улучшить звуки и музыку, добавить поддержку контроллера, провести тестирование и выложить игру на itch.io, а также подготовиться к защите проекта. В будущем, если игра понравится аудитории, я планирую дорабатывать её и выпускать обновления.

Список литературы

[1] Red Nexus Games Inc. *Peglin*. URL: https://store.steampowered.com/app/1296610/Peglin/ (дата обр. 15.12.2023).

[2] Mega Crit Games. *Slay The Spire*. URL: https://store.steampowered.com/app/646570/Slay_the_Spire/ (дата обр. 15.12.2023).

[3] Supergiant Games. *Hades*. URL: https://store.steampowered.com/app/1145360/Hades/ (дата обр. 15.12.2023).

[4] Purple Moss Collectors. *Dungeons and Degenerate Gamblers*. URL: https://store.steampowered.com/app/2400510/Dungeons__Degenerate_Gamblers/ (дата обр. 15.12.2023).

[5] Stage Games Monstars Inc. Resonair. *Tetris® Effect: Connected*. URL: https://store.steampowered.com/app/1003590/Tetris_Effect_Connected/ (дата обр. 10.01.2024).

[6] Blue Planet Software Inc. *Super Rotation System*. URL: https://tetris.wiki/Super_Rotation_System (дата обр. 06.01.2024).

[7] Blue Planet Software Inc. *Tetris Guideline*. URL: https://tetris.wiki/Tetris_Guideline (дата обр. 05.01.2024).

[8] Abject. *InfidHells*. URL: https://abject.itch.io/infidhells (дата обр. 18.12.2023).

[9] SquareDev. *4-Block Dungeon (prototype)*. URL: https://squaredev.itch.io/4-block-dungeon-prototype (дата обр. 18.12.2023).

[10] Igara Studio S.A. *Aseprite - Docs*. URL: https://www.aseprite.org/docs (дата обр. 20.02.2024).

[11] Tesseract. *20 Advanced Coding Tips For Big Unity Projects*. URL: https://www.youtube.com/watch?v=dLCLqEkbGEQ&t=854s (дата обр. 24.01.2024).

[12] Tarodev. *C# Events & Delegates*. URL: https://www.youtube.com/watch?v=6NRqwL3N5Go (дата обр. 25.01.2024).