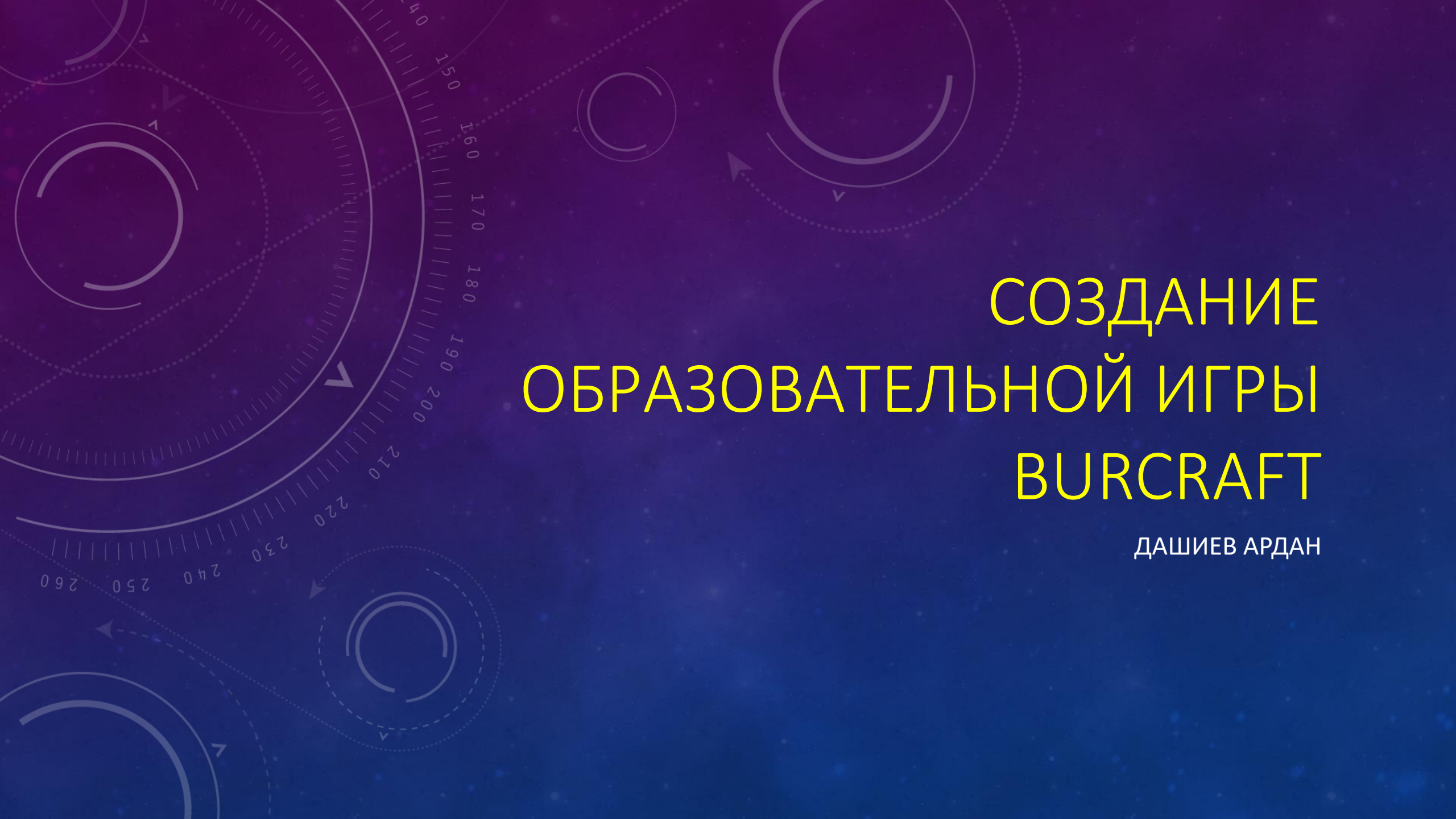




СОЗДАНИЕ ОБРАЗОВАТЕЛЬНОЙ ИГРЫ BURSCRAFT

ДАШИЕВ АРДАН

ВВЕДЕНИЕ В ПРОЕКТ BURCRAFT

Актуальность. С быстрым развитием цифровых технологий традиционные методы обучения уже не всегда удовлетворяют потребности школьников. Все больше учащихся предпочитают интерактивные и увлекательные форматы обучения. Исследования показывают, что геймификация повышает вовлеченность и эффективность учебного процесса.

Цель проекта – разработать интерактивный образовательный продукт Burcraft в качестве дополнительного средства обучения к традиционному учебнику бурятского языка.

Основными задачами проекта стало:

- Изучение механик Minecraft
- Введение процедурной генерации
- Разработка игровых механик
- Оптимизация
- Локализация
- Разработка и интеграция игрового ИИ (искусственного интеллекта).

Роль видеоигр в образовании

- Геймификация повышает вовлеченность учащихся, делая процесс обучения более увлекательным.
- Исследования показывают, что игровые элементы увеличивают интерес к учебным дисциплинам и способствуют лучшему усвоению материала.
- Игры, такие как Minecraft, используются для изучения математики, логического мышления и программирования.

Перспективы Burcraft

- Дальнейшее развитие игры включает создание миров, отражающих реальную действительность, где игроки смогут пройти этапы развития человечества, симулируя научно-технический прогресс (от каменных орудий до разработки своего искусственного интеллекта).
- Планируется добавить многопользовательский режим, позволяющий носителям бурятского языка создавать серверы для общения и практики языка. Интеграция текстового и голосового чата поддержит изучение письменной и устной речи, помогая игрокам быстрее осваивать новый лексический материал.
- Для освоения таких предметов, как физика, инженерия, электроника, химия, биология, география и программирование будут созданы обновления игры.

ОСОБЕННОСТИ MINECRAFT КАК КЛЮЧЕВЫЕ МЕХАНИКИ ПРОЕКТА

Процедурная генерация миров

- Процедурная генерация в Minecraft — это способ создания игровых миров с помощью алгоритмов, а не вручную. Каждый мир состоит из блоков размером 1 кубометр, и его размеры колоссальны, примерно равные площади всей поверхности Земли, умноженной на 7. Это приводит к возникновению бесконечного разнообразия ландшафтов, биомов и структур. Каждое новое создание мира уникально из-за различных параметров генерации.

Блочная структура Minecraft

- Блочная структура — основа Minecraft, определяющая уникальный игровой процесс. Весь мир состоит из кубических блоков размером 1x1x1 метра, что упрощает ориентацию и строительство. Свобода строительства развивает креативность, позволяя игрокам создавать уникальные архитектурные решения и решать нестандартные задачи.

ОТЛИЧИЯ КОНЦЕПЦИЙ BURSCRAFT И MINECRAFT

- В Minecraft акцент сделан на приключенческой и фэнтезийной составляющей, что проявляется в новых механиках и контенте, увеличивающих разнообразие и реиграбельность. Игроки сталкиваются с элементами средневековья и фэнтези, такими как магия и вымышленные существа.
- На сегодня игра находится на стадии применения в качестве интерактивной платформы для расширения словарного запаса учащихся по бурятскому языку.
- В будущем мой проект направлен на реализацию в игре прогресса по имеющимся технологиям и научным достижениям. Он предложит учащимся возможность исследовать и применять реальные концепции, обогащая их знания и формируя интерес к наукам.
- Таким образом, проект рассматривается как дополнение к образовательному процессу, направленное на обучение школьным предметам.

ИСПОЛЬЗОВАНИЕ UNREAL ENGINE 5

1. Выбор движка

Для разработки игры Bircraft был выбран Unreal Engine 5, который предоставляет мощные инструменты для создания игр и упрощает процесс разработки.

2. Прототипирование через Blueprints

На начальном этапе разработки использовались Blueprints — визуальный язык программирования, что позволило быстро создать прототип игры.

3. Переход на C++

После успешного прототипирования я перешел на C++ для повышения производительности, что позволило оптимизировать игру и улучшить общую эффективность кода.

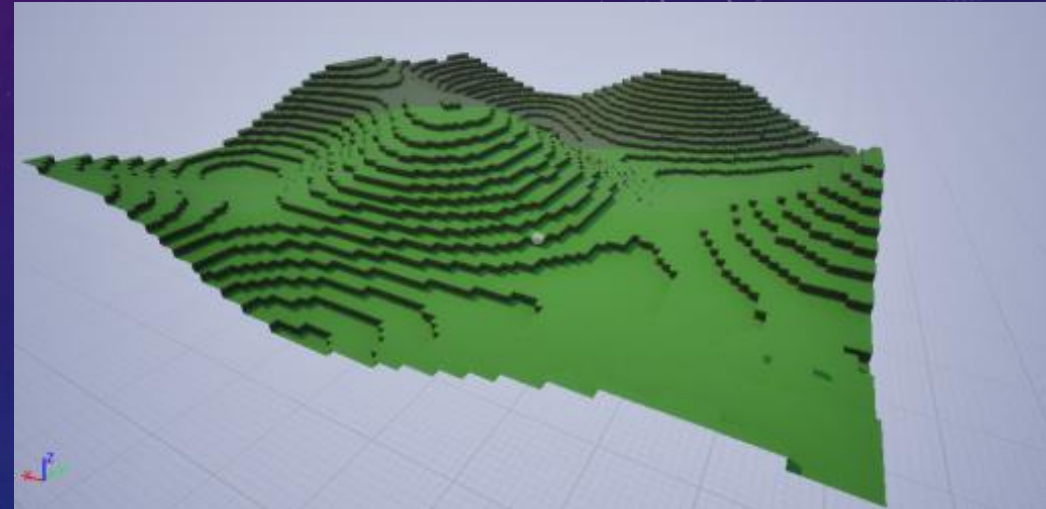
4. Процедурная генерация мира

Используя возможности Unreal Engine 5, была реализована процедурная генерация мира, что позволило создавать уникальные игровые локации и практически бесконечные миры.

ПРОТОТИПИРОВАНИЕ НА BLUEPRINTS

Я разработал эффективную систему процедурной генерации игрового мира с акцентом на оптимизацию производительности. Реализовано было создание чанков — небольших участков мира, чтобы уменьшить нагрузку на систему, и создана система хранения информации о блоках с помощью одномерного массива.

Введена проверка наличия блоков и ограничена генерация новых кубов в окружении существующих блоков, что значительно улучшило производительность. Также я оптимизировал процесс, создавая только видимые грани и заменил bool переменные на enum, чтобы учитывать разные типы блоков. Используя алгоритм шума Перлина, создал органичный ландшафт, а моя функция Terrain Initialisator обеспечивает разнообразие высот, добавляя реалистичность в мир. В результате получился оптимизированный чанк, обладающий рельефом.



Чанки, расположенные вплотную друг к другу

ПЕРЕХОД НА C++

Я перенёс код на C++, установив плагин для Visual Studio и разработав класс ChunkSystem, который улучшил производительность, устранив постоянную генерацию геометрии — теперь она происходит один раз после заполнения массивов. Размер чанка 16x16x255 блоков, и добавил переменную BlockScale для масштабирования. Также создал массив нормалей для улучшения освещённости и функции для добавления треугольников и нормалей.

Разработал класс World System для автоматической генерации чанков вокруг игрока и создал структуру FChunkData, оптимизирующую генерацию, исключая невидимые блоки. Радиусы инициализации настроены с использованием теоремы Пифагора. Генерация чанков осуществляется с помощью таймера и алгоритма спиральной схемы.

```
switch (Direction)
{
    case EDirection::None:
        YOffset -= CHUNKWIDTH;
        Direction = EDirection::Backward;
        break;
    case EDirection::Backward:
        XOffset -= CHUNKLENGTH;
        if (XOffset == -XStep * CHUNKLENGTH)
        {
            Direction = EDirection::Right;
            YStep++;
        }
        break;
    case EDirection::Right:
        YOffset += CHUNKWIDTH;
        if (YOffset == YStep * CHUNKWIDTH - CHUNKWIDTH)
        {
            Direction = EDirection::Forward;
            XStep++;
        }
        break;
    case EDirection::Forward:
        XOffset += CHUNKLENGTH;
        if (XOffset == XStep * CHUNKLENGTH - CHUNKLENGTH)
        {
            Direction = EDirection::Left;
        }
        break;
    case EDirection::Left:
        YOffset -= CHUNKWIDTH;
        if (YOffset == -YStep * CHUNKWIDTH)
        {
            Direction = EDirection::Backward;
        }
        break;
}

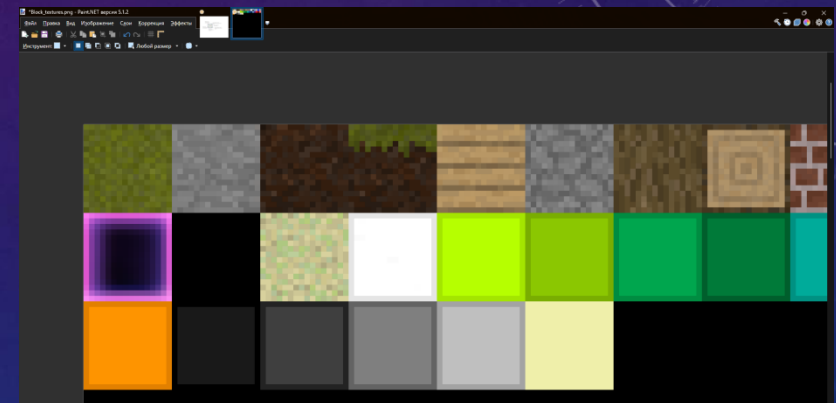
ChunkPosition.X = PlayerChunkPosition.X + XOffset;
ChunkPosition.Y = PlayerChunkPosition.Y + YOffset;
```

Код спиральной генерации чанков

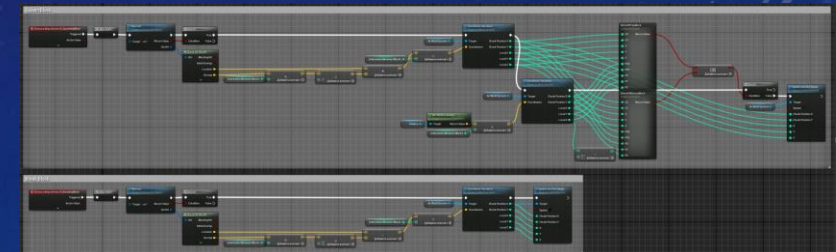
УЛУЧШЕНИЕ ПРОЦЕДУРНОЙ ГЕНЕРАЦИИ, ТЕКСТУРИРОВАНИЕ, МНОГОПОТОЧНОСТЬ, РАЗРУШЕНИЕ И УСТАНОВКА БЛОКОВ

Я интегрировал библиотеку FastNoiseLite для процедурной генерации, реализовал функцию AddUvs для вычисления UV-координат блоков и создал механики для разрушения и установки блоков с помощью луча.

Для повышения скорости исполнения кода добавил многопоточность с классом FChunkCalculator, который вычисляет геометрию чанков в отдельном потоке. Результаты возвращаются в основной поток для обновления мира без конфликтов. Параметры помещены в структуру MeshData для передачи в очередь мешей. Каждый кадр проверяется очередь, и функция SpawnChunk создаёт актор чанка и вызывает SetChunkMesh для его создания.



Нарисованные текстуры для блоков



*Схема логики разрушения и установки
блоков*

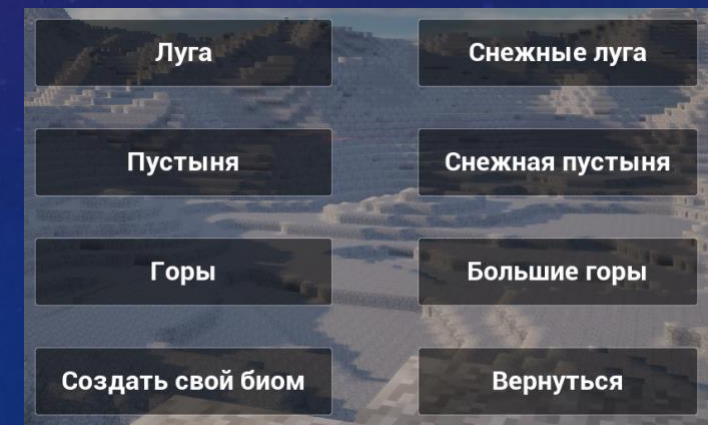
UX/UI РАЗРАБОТКА ИНТЕРФЕЙСОВ

Задачи, которые были выполнены на данном этапе:

- Создание интуитивно понятного интерфейса для улучшения взаимодействия пользователя с игрой.
- Реализация главного меню с возможностью выбора языка (бурятский, русский, английский).
- Внедрение системы инвентаря для хранения различных блоков и управления ими.
- Возможности настройки дистанции появления чанков и сид мира.
- Гибкая настройка процедурной генерации, позволяющая игрокам создавать уникальные игровые миры.



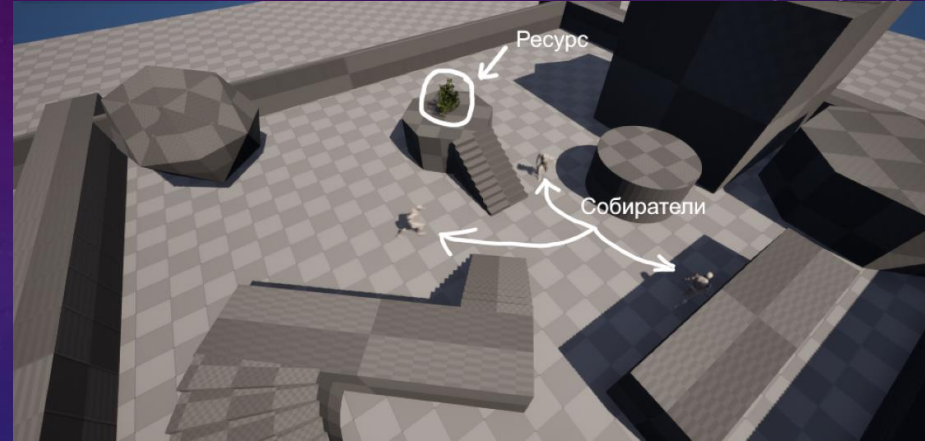
Интерфейс инвентаря



Выбор биомов для генерации мира

СОЗДАНИЕ И ВВЕДЕНИЕ ИИ В ИГРУ

- В рамках проекта Burcraft была разработана модель поведения ИИ, названная моделью собирателя, которая имитирует инстинкты живых существ. Интеллектуальный агент (ИИ) собирает ресурсы, аналогично животным-собирателям, ведя поиск и соревнуясь с другими агентами за доступные ресурсы. Когда агент находится в состоянии поиска, он выбирает случайную позицию в радиусе своей досягаемости и движется к ней, пока не обнаружит ресурс, после чего стремится к нему, ускоряясь при наличии конкуренции



Тестирование ИИ на специальном уровне

Планируется внедрение новых моделей поведения, таких как травоядные и хищники, что создаст дополнительные стратегии выживания и реакции самосохранения у собирателей и травоядных. Эти элементы добавят интерес и сложность в игровую динамику, включая охоту как способ пропитания для игроков. Хотя текущая модель примитивна, существует потенциал для её развития за счет генетических характеристик и новых состояний поведения. Проект Burcraft будет не только имитировать живую природу, но и служить образовательной платформой для изучения биологии и генетики.

ЛОКАЛИЗАЦИЯ И БИЛД ИГРЫ

Игра Burcraft обучает новым словам через названия блоков и предметов. Текст игры переведён с английского на русский и бурятский (с помощью онлайн-переводчиков и корректировок учителя бурятского языка). Планируется многопользовательский режим с серверами для общения на бурятском, с текстовым и голосовым чатом для практики письменной и устной речи.

Таблица переводов

#	A	B	C
1	Английский язык	Русский язык	Бурятский язык
2	Air	Воздух	Агаар
3	Aquamarine block	Аквамариновый блок	Аквамарин блок
4	Black block	Чёрный блок	Хара блок
5	Blue block	Синий блок	Хүхэ блок
6	Bricks	Кирпичи	Хирпиис
7	Brown block	Коричневый блок	Хүрэн блок
8	Cobblestone	Булыжник	Хабтагай шулуун
9	Dark blue block	Тёмно-синий блок	Хара хүхэ блок
10	Dark gray block	Тёмно-серый блок	Хара саарал блок
11	Dark green block	Тёмно-зелёный блок	Хара ногоон блок
12	Dirt	Земля	Газар
13	Error	Ошибка	Алдауу
14	Glass	Стекло	Шэл
15	Grass	Трава	Ногоон
16	Gray block	Серый блок	Саарал блок
17	Green block	Зелёный блок	Ногоон блок
18	Ice	Лёд	Мүлхэн
19	Light blue block	Светло-синий блок	Санхир блок
20	Light gray block	Светло-серый блок	Сагаан саарал блок
21	Lime block	Лаймовый блок	Шара ногоон блок
22	Orange block	Оранжевый блок	Улаан шара блок
23	Pink block	Розовый блок	Ягаан блок
24	Red block	Красный блок	Улаан блок
25	Sand	Песок	Элхэн
26	Snow	Снег	Саһан
27	Snowy grass	Заснеженная земля	Саһатай газар
28	Stone	Камень	Шулуун
29	Tree trunk	Ствол дерева	Модоной үндэһэн
30	Violet block	Фиолетовый блок	Ягаан хүхэ блок
31	White block	Белый блок	Сагаан блок
32	Wood	Дерево	Модон
33	Yellow block	Жёлтый блок	Шара блок
34	Quit to main menu	Выйти в главное меню	Гол меню руу гараха
35	Resume	Продолжить	Үргэлжэхэ
36	Exit	Выйти	Гарагты
37	Snowy plains	Снежные луга	Саһатай нуганууд
38	Plains	Луга	Нуганууд
39	Octaves	Октавы	Октаванууд
40	Settings	Настройки	Тохиргоонууд
41	Create new world	Создать новый мир	Шэнэ дэлхэй байгуулха
42	Language	Язык	Хэлэн
43	Desert	Пустыня	Губи
44	Mountains	Горы	Ууланууд
45	Create own biome	Создать свой биом	Өөрынгөө биом бүтээхэ
46	Domain warp	Доменное искажение	Доменнэ гажангир
47	Underground height :	Высота подземного слоя :	Газар доорохи хүрһэнэй үеын үндэр :
48	Frequency :	Частота :	Частота :
49	Ground :	Земной слой :	Газарай дабхар :
50	Chunk render distance :	Дистанция отрисовки :	Зайе харуулха :
51	Create new octave	Создать новую октаву	Шэнэ октава зохиохо
52	Noise type :	Тип шума :	Шууяанай түхэл :
53	Amplitude :	Амплитуда :	Амплитуда :
54	Save and back	Сохранить и выйти	Хадагалжа гараха
55	Seed :	Сид :	Сид :
56	Base height :	Базовая высота :	Үндэһэн үндэр :
57	Surface :	Поверхность :	Дэбисхэр :
58	Underground :	Подземный слой :	Газар доорохи хүрһэнэй үе :

ЗАКЛЮЧЕНИЕ



Создана игра Burcraft с образовательными целями, основанная на добровольном вовлечении игрока. Получен ценный опыт разработки игр на C++ и Blueprints в Unreal Engine 5. Задачи, поставленные в начале работы, выполнены. В перспективе - многопользовательский режим и создание миров, моделирующих развитие человечества через научно-технический прогресс.