Слайд 1

Здравствуйте, уважаемая комиссия. Меня зовут Пугач Владислав Олегович. Я являюсь студентом группы ПКС-43 и представляю Вам дипломный проект на тему «Разработка программного продукта «Генератор игрового мира».

Слайд 2

На данный момент компьютерные игры очень популярны среди множества людей. Это связано с тем, что в игровой индустрии представлено большое количество жанров и стилей, хотя бы один из которых обязательно подойдёт конечному пользователю.

Крупная часть игровой индустрии представлена трехмерными проектами, в которых немаловажную роль играет игровой мир и его наполнение. Создание игрового мира - сложный и трудозатратный процесс, в связи с чем занимает большую часть времени в процессе разработки игры. Это приводит к тому, что разработчики нередко используют генераторы игровых миров, особенно когда в игре подразумевается большой или вовсе бесконечный открытый мир.

В открытом доступе подобных генераторов крайне мало, так как их разработка ещё более сложна, чем моделирование игрового мира вручную и, как правило, такие генераторы разрабатывают для конкретного проекта и вне их не распространяют.

Исходя из этого можно сказать, что выбранная тема актуальна.

Слайд 3

Исходя из описанной темы была обозначена цель разработать программный продукт «Генератор игрового мира».

Для достижения поставленной цели, были определены следующие задачи:

* исследование предметной области;
* проектирование пользовательского интерфейса;
* проектирование компонентов приложения;
* разработка пользовательского интерфейса;
* разработка приложения;
* разработка справочной системы.

Слайд 4

Для того, чтобы полностью сформулировать требования к приложению, были рассмотрены некоторые существующие разработки, являющиеся генераторами мира или играми с этими генераторами внутри.

Minecraft – компьютерная игра в жанре песочницы с условно бесконечным процедурно генерируемым миром, построенным на основе блоков, состоящих из вокселей. Это можно увидеть на рисунках, представленных на слайде. Игровой мир имеет высоту 384 блоков. Всего карта может содержать до 9,216×1017 (921 600 000 000 000 000). Поэтому в игре невозможно не использовать процедурную генерацию мира.

К достоинствам генератора мира игры можно отнести использование трехмерного шума Перлина, что, в совокупности с использованием блоков для построения мира, позволяет генерировать множество разнообразных форм без сильных затрат ресурсов устройства игрока. На слайде можно увидеть сгенерированные горы и лес, а также строение, заранее подготовленное из блоков.

К недостаткам можно отнести использование блоков, не подходящих для большинства игр.

Слайд 5

Следующей была рассмотрена игра No Man’s Sky. Она является космическим приключенческим боевиком. Мир игры состоит из 264 планет, каждая из которых имеет свои особенности рельефа, флоры, фауны, погоды.

Вручную такое количество разнообразных миров создать невозможно, из-за чего разработчики широко применяют различные алгоритмы генерации, чаще всего основанные на шуме Перлина, который будет рассмотрен далее.

К достоинствам генератора игры можно отнести следующее:

1. Большое разнообразие генерируемых форм, что достигается использованием множества взаимосвязанных между собой детерминированных алгоритмов (на слайде можно увидеть сгенерированные в игре пещеру, равнину, горы и причудливые формы, напоминающие змей);
2. Высокая детализация мира;

Недостатков в конкретном генераторе не выявлено.

Слайд 6

Также был рассмотрен набор инструментов для Unity «Gaia Pro 2021».

Gaia Pro 2021 – набор инструментов для Unity 3d, позволяющий генерировать детализированные ландшафты. С начала с помощью встроенных инструментов и предопределенных моделей создаётся рельеф, после чего система автоматически окрашивает и наполняет его в зависимости от выбранных параметров, таких как, например, погода.

Достоинства утилиты заключаются в большом наборе разнообразных инструментов, позволяющих пользователю по своему усмотрению гибко менять мир. Также немаловажным достоинством является автоматическое заполнение настроенного мира различными объектами, такими как растения и строения. Результат работы утилиты можно увидеть на текущем слайде.

Самым большим недостатком утилиты является необходимость пользователю самому менять один из заготовленных рельефов различными инструментами, то есть генерируется только наполнение мира, а не сам мир.

Слайд 7

После анализа предметной области и существующих разработок, к проекту были выдвинуты следующие требования:

* Генерация шестиугольной полигональной сетки (ячейки);
* Составление сетки из сгенерированных ячеек;
* Генерация рельефа путём изменения высоты вершин ячеек;
* Изменение типа поверхности в зависимости от её высоты;
* Симуляция водной эрозии;
* Симуляция роста деревьев;
* Размещение деревьев в игровом мире.

Слайд 8

Для разработки программного продукта была выбрана среда разработки компьютерных игр Unity.

Так как C# - единственный язык программирования, поддерживаемый Unity, он был выбран как язык программирования для разработки проекта.

Для редактирования C#-кода была выбрана интегрированная среда разработки Visual Studio.

В качестве инструмента проектирования интерфейса была выбрана утилита Adobe XD.

Слайд 9

Для обеспечения пользователю возможности взаимодействия с программным продуктом необходим пользовательский интерфейс. Для разработки интерфейса необходимо спроектировать его макет. На слайде представлен макет интерфейса главного меню генератора. Меню содержит поле ввода и 4 кнопки.

Поле ввода необходимо для возможности задать зерно генерации, которое будет использовано для получения псевдослучайных чисел в процессе работы генератора.

Кнопка в форме вопросительного знака позволяет открыть текст справки, описывающей возможные действия в открытом окне.

При нажатии на кнопку «Начать» запускается процесс генерации игрового мира на основе введенных данных.

При нажатии на кнопку «Выход» завершается работа программного продукта.

При нажатии на кнопку «Настройки мира» происходит переход к окну настроек мира.

Слайд 10

На данном слайде представлен макет окна настроек.

Окно настроек игрового мира содержит список настраиваемых параметров с бегунками, позволяющими изменять значения соответствующих им параметров, кнопку справки в форме вопросительного знака и кнопку «Назад».

Бегунки позволяют изменить необходимый параметр в пределах допустимых значений, что приведет к изменениям внешнего вида сгенерированного на их основе мира.

При нажатии на кнопку справки откроется текст справки, содержащий информацию о каждом имеющемся параметре.

При нажатии на кнопку «Назад» будет произведен переход в главное меню генератора.

Слайд 11

Также был спроектирован макет интерфейса меню паузы, которое может быть открыто во время перемещения по игровому миру. Макет показан на текущем слайде.

Меню паузы содержит две кнопки и информацию о зерне генерации.

При нажатии на кнопку «Продолжить» меню паузы будет закрыто и пользователь сможет продолжить свои действия в игровом мире.

При нажатии на кнопку «В главное меню» будет произведён переход в главное меню.

Поле «Зерно генерации» содержит зерно генерации текущего мира.

Слайд 12

Для понимания структуры программного продукта была спроектирована его структурная схема. Её можно увидеть на текущем слайде.

Блок «Работа генератора» подразумевает под собой работу всего программного продукта. Он объединяет множество описанных систем, каждая из которых выполняет свою, отличную от прочих, задачу.

Каждая система является совокупностью функций, предназначенных для цели, соответствующей названию системы.

Одной из наиболее важных систем является система хранения и вычисления информации о вершинах. Она хранит информацию о каждой из точек полигональных сеток, составляющих игровой мир, а также позволяет проводить различные математические вычисления, необходимые для генерации мира.

Слайд 13

На слайде показан реализованный на основе макетов интерфейс. Функционал интерфейса полностью соответствует описанному при проектировании.

Слайд 14

Основным инструментом, используемым для генерации игрового мира, является шум Перлина.

Шум Перлина — это градиентный шум, состоящий из набора псевдослучайных единичных векторов (направлений градиента), расположенных в определенных точках пространства и интерполированных функцией сглаживания между этими точками.

На левом рисунке можно увидеть как выглядит шум.

На правом схематично изображены единичные векторы, образующие подобное изображение.

Слайд 15

Для того, чтобы образовать рельеф, необходимо использовать значение шума в точках сетки как значение высоты, таким образом вытягивая плоскую форму в трехмерную.

Квадратная сетка проста в использовании и индексации, но имеет существенный изъян: при изменении высоты ряда точек в диагонали, могут появиться изломы, изображенные на рисунке слева.

В связи с этим была рассмотрена сетка равносторонних треугольников. Подобная сетка не имеет найденного в квадратной изъяна, но более сложна в индексации и обращении, так как состоит из двух типов ячеек, являющихся повернутыми на 180 градусов вариантами друг друга.

Так как треугольные сетки дают наиболее удовлетворительный результат, было решено использовать сетки шестиугольников, поскольку они состоят из равносторонних треугольников и имеют одинаковую форму.

Слайд 16

Для индексации и размещения шестиугольных ячеек сетки и точек в ячейках используется несколько систем координат.

Для индексации объектов используются кубические координаты и координаты смещения. Их схематическое представление изображено на слайде.

Кубические координаты представляют собой набор из 3х координат, сумма которых всегда равна нулю. Их удобно использовать для получения обычных декартовых координат. Это используется при размещении точек и ячеек в пространстве.

Координаты смещения напоминают декартовы координаты в двух измерениях, с одним отличием: каждая четная или нечетная строка или столбец, в зависимости от того, как располагаются соседние объекты) смещаются по соответствующей им оси. Для ячеек используется система чёт-c (четные колонки смещаются вверх), а для вершин – нечёт-r (нечетные строки смещены влево)

Это удобно использовать для нахождения объекта по его декартовым координатам.

Таким образом, имея для каждой ячейки и вершины список пар координат смещения и кубических координат, можно легко узнать где находится объект или по его положению найти его индексы.

Слайд 17

Для реализации систем генерации вершин полигональной сетки и объединения вершин полигональной сетки были разработаны следующие два C#-класса: HexVertex и HexChunk.

C#-класс HexVertex позволяет хранить и изменять данные о вершине полигональной сетки

C#-класс HexChunk содержит функционал, обеспечивающий генерацию вершин, их размещение на плоскости и объединение в полигоны. Генерация вершин происходит следующим образом:

1) в C#-классе VerticesContainer вычисляются направления осей кубических координат;

2) в C#-классе HexChunk генерируется множество вершин с кубическими координатами, каждая из которых находится в пределах от отрицательного заданного значения, до положительного;

3) каждой вершине задаются декартовы координаты, исходя из её кубических координат

Слайд 18

Для реализации системы генерации сетки игрового мира был разработан C#-класс TerrainMeshGenerator. Он предоставляет функционал, позволяющий составить из множества полигональных сеток одну шестиугольную сетку. Для этого используется алгоритм, схожий с алгоритмом распределения вершин, но оси кубических координат наклоняются на 30 градусов, что позволяет расставить ячейки таким образом, чтобы они соприкасались друг с другом только своими гранями.

Результат работы системы представлен на слайде. На нем можно увидеть, что карта мира состоит из множества шестиугольных ячеек напоминающих собой соты.

Слайд 19

Для реализации системы генерации рельефа был разработан C#-класс HexNoises. C#-класс включает в себя набор методов, позволяющих деформировать сетку вершин в соответствии с определенными детерминированными алгоритмами.

Среди алгоритмов, реализованных в C#-классе, наиболее важным является многооктавный шум Перлина.

Многооктавный шум Перлина – несколько шумов Перлина с масштабом, меняющимся в геометрической прогрессии в зависимости от его порядка, наложенных друг на друга. Подобный подход позволяет сделать шум более шероховатым, что делает поверхность мира, сгенерированную на его основе, более естественной.

На слайде можно увидеть gif-анимацию с результатами применения шума Перлина с шестью октавами

Слайд 20

Для того, чтобы рельеф выглядел более правдоподобным, решено было использовать симуляцию водной эрозии.

Водная эрозия почв – это разрушение почвенного покрова под действием поверхностных водных потоков: дождевых, талых и поливных.

Симуляция происходит следующим образом. В произвольном месте на сетке ячеек создается капля. В зависимости от наклона поверхности капля начинает своё движение по склону. Пока капля движется и её объем достаточен для вмещения в себя осадка, она вбирает в себя часть поверхности в некотором радиусе (тем самым уменьшая её высоту). Если капля по инерции движется вверх, остановилась или вобрала в себя максимальное количество осадка, она оставляет некоторое количество осадка ближайшим вершинам, увеличивая их высоту.

Внешний вид рельефа побережья до применения эрозии и после показан на слайде.

Слайд 21

Для заполнения мира растительностью было решено разработать генератор деревьев на основе симуляции транспорта питательных веществ. Симуляция происходит следующим образом:

* создаётся корневая ветка;
* далее в нее поступает некоторое количество питательного вещества;
* ветка расширяется и, если не имеет потомков, удлинняется;
* если количество полученного питания превышает некий порог, ветка разветвляется;

Также, на концах веток генерируется икосфера, имитирующая крону дерева.

Слайд 22

На данном слайде можно увидеть видеоматериал, позволяющий увидеть процесс роста деревьев с разными параметрами.

Слайд 23

Для того, чтобы пользователь мог в полной мере использовать возможности генератора, необходима справочная система.

В программном продукте она реализована в виде подсказок, вызываемых пользователем по необходимости.

На слайде показана справочная система главного меню. Она вызывается нажатием на вопросительный знак в верхнем правом углу.

Слайд 24

Подобным образом вызывается и выглядит справка окна настроек.

Для обеспечения работы справочной системы был разработан C#-класс HintButton. Его функционал позволяет привязать к кнопке множество объектов пользовательского интерфейса, которые будут скрываться и раскрываться по нажатию этой кнопки.

Слайд 25

В связи с тем, что генерация мира занимает несколько минут, на слайде представлен видеоматериал, позволяющий увидеть работу программного продукта.

Слайд 26

В процессе дипломного проектирования был спроектирован и разработан программный продукт «Генератор игрового мира». Для достижения полученного результата в полной мере были выполнены поставленные задачи:

* исследование предметной области;
* проектирование пользовательского интерфейса;
* проектирование компонентов приложения;
* разработка пользовательского интерфейса;
* разработка приложения;
* разработка справочной системы;