|  |  |
| --- | --- |
| Пермский филиал федерального государственного автономного  образовательного учреждения высшего образования  «Национальный исследовательский университет  «Высшая школа экономики» | |
| *Факультет экономики, менеджмента и бизнес-информатики* | |
| Шевчук Михаил Романович | |
| **ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ ИГРЫ С РАЗРУШАЕМЫМ ЛАНДШАФТОМ** | |
| *Выпускная квалификационная работа* | |
| по направлению подготовки *09.03.04 Программная инженерия* образовательная программа «Программная инженерия» | |
| Рецензент | Научный руководитель старший преподаватель кафедры информационных  технологий в бизнесе Г. И. Рустамханова  Консультант старший преподаватель кафедры информационных  технологий в бизнесе В. В. Лебедев |
| Пермь, 2020 год | |

# Аннотация

Оглавление

[Аннотация 2](#_Toc59659908)

[Введение 4](#_Toc59659909)

[Глава 1. Анализ предметной области 7](#_Toc59659910)

[1.1. Анализ существующих игр 7](#_Toc59659911)

[1.2. Анализ существующих методов генерации деформируемого ландшафта 7](#_Toc59659912)

[1.3. Выявление требований к разрабатываемой игре 8](#_Toc59659913)

[Глава 2. Проектирование игровых компонентов и механик 9](#_Toc59659914)

[2.1. Конструирование архитектуры разрабатываемой игры 9](#_Toc59659915)

[2.2. Проектирование механики генерации TD 9](#_Toc59659916)

[2.3. Проектирование геймплея игрока 9](#_Toc59659917)

[2.4. Проектирование пользовательского интерфейса игры 9](#_Toc59659918)

[Глава 3. Реализация игровых компонентов и механик 10](#_Toc59659919)

[Глава 4. Тестирование игрового процесса 11](#_Toc59659920)

[Заключение 12](#_Toc59659921)

[Библиографический список 13](#_Toc59659922)

# Введение

Рынок компьютерных игр с каждым годом набирает новые обороты. Всё больше крупных компаний, таких как BMW, Intel, Nike, Xiaomi, инвестируют в киберспорт миллионы долларов. В период пандемии COVID-19 индустрия компьютерных игр стала одной из немногих отраслей, которая не только существенно не пострадала от карантинных ограничений, но и показала рост прибыли и количества клиентов. К примеру, в Китае выручка рынка видеоигр за первые месяцы 2020 года выросла на 25,2% по сравнению с концом 2019, а прибыль с мобильных игр увеличилась на 37,6%, что говорит о том, что индустрия в ближайшее время будет только расширяться [1].

Особенностью компьютерных игр является разнообразие жанров и механик игрового процесса. Одной из таких механик является механика «разрушаемого окружения» (destructible environment, далее DE). Под DE подразумеваются игровые объекты, которые могут быть разрушены или деформированы в результате действий игрока. Это могут быть строения, предметы интерьера, растительность и другие искусственные, либо природные структуры. Способность игрока разрушить стену или препятствие в лабиринте и таким образом создать себе новый вариант пути является примером механики DE.

Существует отдельный подвид DE – «деформируемый ландшафт» (terrain deformation, далее TD), имеющий отличия и сложности в плане реализации. Если при разрушении объекта, находящего на ландшафте, на программном уровне достаточно просто заменить его текстуру или удалить из памяти, то при деформации самого ландшафта меняется его геометрия. К примеру, деформация исходного плоского полигона поверхности приводит к разбиению его на множество полигонов, больше не лежащих в его плоскости. При этом постоянное динамическое изменение ландшафта создаёт большую нагрузку на аппаратные ресурсы компьютера. И по этой причине механика TD меньше всего используется в самом динамичном жанре игр – «шутер».

В данной работе представлена разработка и реализация игры в жанре «шутер» с использованием механики TD.

**Объектом исследования** является игра в жанре «шутер» с механикой деформации ландшафта, а **предметом исследования** является процесс генерации деформируемого ландшафта для игр жанра «шутер».

Цель выпускной квалификационной работы – разработка 3D игры в жанре «шутер» с разрушаемым ландшафтом. Для достижения поставленной цели нужно выполнить следующие задачи:

1. Проведение анализа предметной области:
   1. Анализ существующих игр, использующих механику TD.
   2. Анализ существующих методов генерации TD.
   3. Описание механик разрабатываемой игры.
   4. Выявление функциональных требований.
2. Проектирование игровых механик и игрового интерфейса:
   1. Проектирование механики генерации TD.
   2. Проектирование механики передвижения игрока.
   3. Проектирование механики взаимодействия игрока с окружением.
3. Реализация игровых механик и игрового интерфейса:
   1. Выбор игрового движка для реализации.
   2. Реализация игровых механик.
   3. Разработка игровой графики и пользовательского интерфейса.
4. Тестирование игровых механик.

Степень разработанности проблемы способствует выполнению поставленной цели, так как существуют актуальные исследования на тему динамически изменяемых ландшафтов, а также существуют примеры игр, использующих данную механику. Практической значимостью игры является реализация метода с использованием нового для проблемы инструментального средства, что может привести к расширению спектра игровых механик и жанров на рынке в случае успеха продукта. Теоретической значимостью является доказательство сходимости алгоритмов и методов при решении проблемы генерации TD для динамических игр жанра «шутер».

В процессе работы предполагается использование следующих методов исследования:

1. Анализ – рассмотрение существующих игр с подобными механиками и рассмотрение существующих способов реализации генерации TD.
2. Дискретизация – один из методов генерации TD – представление его не как одной поверхности, а в виде совокупности примитивов – вокселей.
3. Визуализация – рендеринг полученного результата на мониторе игроков.
4. Анализ предметной области

В первой части данной главы описан анализ существующих решений – игр, которые используют механику TD. Затем, на основе полученных данных выделяются основные существующие методы реализации механики TD, приводится анализ исследований этих методов и других, которые ранее возможно в играх не использовались. Затем, на примере разобранных игр, выделяются необходимые механики и функциональные и нефункциональные требования к игре, и требования дополняются новыми механиками для реализуемой игры.

Список исследовательских вопросов:

1. Какие существуют игры, использующие механику TD? В каком жанре они созданы и как именно используют TD? Какие методы генерации TD при этом применяются?
2. Что из себя представляют методы TD? Как они реализуются? Какие имеют плюсы и минусы?
3. Какие обязательные механики должна включать в себя игра? Какие дополнительные механики дали бы ей преимущество перед другими играми?

## 1.1. Анализ существующих игр

Для разработки 3D игры в жанре «шутер», использующей механику TD, требуется рассмотрение существующих решений с целью выделения:

1. используемых способов создания разрушаемого ландшафта;

2. общих паттернов жанра;

Проводить анализ существующих решений будем по следующим параметрам:

1. Степень разрушаемости окружения.

2. Способы взаимодействия игрока с окружением.

3. Положение игровой камеры.

4. Геймплей игрока.

5. Особенности игрового ландшафта.

### 1.1.1. Worms 3D

## 1.2. Анализ существующих методов генерации деформируемого ландшафта

Как мы можем видеть, игры, в которых игрок может так или иначе деформировать ландшафт, используют для этого в основном два метода: изменение полигональной сетки ландшафта и генерация ландшафта с помощью вокселей. Рассмотрим оба этих подхода и как они работают с поставленной задачей.

### 1.2.1. Генерация ландшафта с помощью вокселей

Если пиксель является «элементом изображения» (picture element), то воксел является «элементом объёма» (volume element). Воксели можно сравнить с атомами, из которых состоит трёхмерное пространство, и их совокупность представляется в виде трёхмерной сетки. Каждый воксел является отдельным объектом и поэтому может хранить в себе необходимую для задачи информацию, такую как цвет, прозрачность (для симуляции таких вещей как дым, вода), плотность и тому подобное. Информация о местоположении вокселя не требуется, так как это определяется его индексом в сгенерированной сетке [5].

Основное отличие между полигонами и вокселями в том, что в то время, как полигоны покрывают пустое трёхмерное пространство, создавая полую фигуру, вокселями составляют объём этой фигуры. Другими словами, объём фигуры равен сумме вокселей внутри неё. Благодаря этому, решение проблемы деформации какого-либо объекта, состоящего из вокселей, сводится к задаче прибавления и вычитания этих вокселей в нужных местах [5].

Проблемой вокселей является больше потребление памяти, имеющую величину . То есть локация, размерностью в 1000 вокселей по каждой оси уже имеет размер в 1 000 000 000 блоков, что создаёт огромную нагрузку на память процессора. Однако, соседние воксели часто имеют схожие значения, а некоторые части сетки вообще могут быть пустыми. Поэтому одним из основных методов оптимизации воксельной сетки и построения воксельного мира является построение разряжённого воксельного октодерева (Sparse voxel octree, далее SVO). Такой подход использовали авторы игры Teardown, описанной выше [6], и такой подход использует физический движок PhysX, в данный момент принадлежащий компании NVIDIA [7].

Суть SVO состоит в следующем. Поместим объект в куб, и начнём делить этот куб на части, каждую из которых мы можем рекурсивно детализировать подобным образом. Таким образом получится дерево областей, которые содержат в себе какую-то часть нашего объекта. На рисунке ниже (указать рисунок) показано, как мы проделываем процедуру до тех пор, пока конечная область не будет содержать один элемент или будет пустой. Первоначальный куб является корнем дерева, конечные области являются листьями. Если же мы как-либо изменим один из листьев, то подобным образом мы разобьём его на меньшие части. То есть пока объект не тронут, он представляет собой один большой «куб», который меньше нагружает систему.

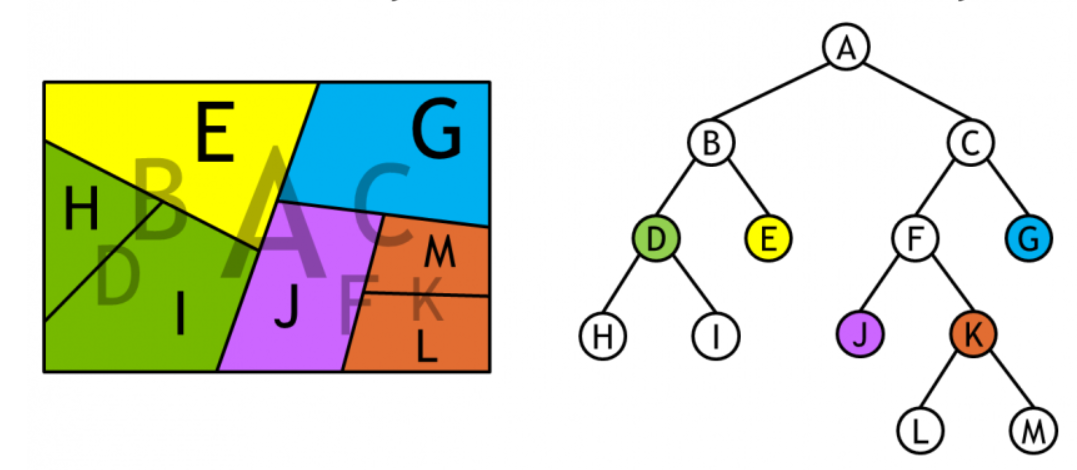


Рисунок . Пример построения октодерева

## 1.3. Выявление требований к разрабатываемой игре

Результатами данной главы являются:

1. Выделение общих (необходимых) механик для игры на основе просмотренных аналогов.
2. Добавление дополнительных механик, отличных от разобранных в аналогах. При этом важно, чтобы дополнительные механики вписывались в концепцию выбранного метода генерации TD.
3. Определение списка функциональных требований на основе определённых механик.
4. Определение списка нефункциональных требований на основе определённых механик.
5. Проектирование игровых компонентов и механик

Список исследовательских вопросов:

1. Из каких подсистем состоит разрабатываемая игра? Какова связь между ними? Как с этими подсистемами взаимодействует пользователь?
2. Как организована механика генерации TD? Как организован геймплей игрока, его передвижение по карте, взаимодействие с игровыми объектами и ландшафтом?

## 2.1. Конструирование архитектуры разрабатываемой игры

Результатом данной главы является:

1. Составление списка игровых подсистем
2. Построение схемы взаимодействий этих подсистем

Результат достигается путём декомпозиции основных игровых механик, выявления связей и отношений между ними и построения диаграммы последовательностей и диаграммы понятий.

## 2.2. Проектирование механики генерации TD

Результатом данной главы является модель генерации и поведения ландшафта в зависимости от действий игрока. Данный результат достигается с помощью построения модели структуры ландшафта и рассмотрения возможностей изменения ландшафта в зависимости от выбранного метода генерации.

## 2.3. Проектирование геймплея игрока

Результатом данной главы является модель возможного поведения игрока и взаимодействия его с игровыми объектами.

## 2.4. Проектирование пользовательского интерфейса игры

Результатом данной главы является описание интерфейса игры, описание главного меню и настроек, управления игрока, настройка игровой камеры.

1. Реализация игровых компонентов и механик

Список исследовательских вопросов: какой инструмент (игровой движок) выбирается для разработки игры? Как реализовывается алгоритм генерации деформируемого ландшафта? Как реализовывается геймплей игрока?

1. Тестирование игрового процесса

Список исследовательских вопросов: какие существуют баги в игре и насколько они критичны? Корректно ли приложение отвечает на действия пользователя?

# Заключение

# Библиографический список

1. Как растёт индустрия компьютерных игр и киберспорт в условиях пандемии [Электронный ресурс]. – Режим доступа: <https://www.kommersant.ru/doc/4325738> (дата обращения 06.12.2020).
2. Zhao, J. Physically based modeling and animation of landslides with MPM // The Visual Computer. – 2019. – Vol. 35. – P. 1223-1235.
3. Spuy, van der R. Advanced Game Design with Flash – NY.:Apress, 2010 – P. 305-366.
4. Gregory, J. Game Engine Architecture – 3rd ed. – FL.:CRC Press, 2018. – P. 525-544.
5. Akenine-Moller, T. Real-time Rendering – 4th ed. – FL.:CRC Press, 2018. – P.545-589.
6. Разрушаемость в Teardown: как совместить геймплей и технологии [Электронный ресурс]. – Режим доступа: <https://media-xyz.com/ru/articles/714-razrushaemost-v-teardown-kak-sovmestit-geimple> (дата обращения 22.01.2021).
7. NVIDIA Blast [Электронный ресурс]. – Режим доступа: <https://developer.nvidia.com/blast> (дата обращения 22.01.2021).