Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«‎Иркутский государственный университет»  
(ФГБОУ ВО «‎ИГУ»)  
Институт математики и информационных технологий  
Кафедра алгебраических и информационных систем

**ОТЧЕТ**  
**ПО УЧЕБНОЙ ПРАКТИКЕ**

Студенты 2 курса очного отделения  
Группа 02271–ДБ  
Понаморевский Антон Алексеевич  
Брагин Илья Дмитриевич

Руководители:  
старший преподаватель Ильин Б. П.

**Иркутск 2022**

Оглавление

[Введение 3](#_Toc105409995)

[Раздел 1. Исследование предметной области 4](#_Toc105409996)

[**1.1. Описание предметной области** 4](#_Toc105409997)

[**1.2. Постановка задачи** 5](#_Toc105409998)

[Раздел 2. Обзор технологий разработки 7](#_Toc105409999)

[**2.1. Язык программирования C++** 7](#_Toc105410000)

[**2.2. Blender** 9](#_Toc105410001)

[**2.3. Unreal Engine** 11](#_Toc105410002)

[**2.4. Blueprints** 14](#_Toc105410003)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 17](#_Toc105410004)

Введение

Игровая индустрия образовалась совсем недавно, но ее стремительные темпы развития привели к тому, что она является одной из крупнейших в мире.

Мир игр представляет широкий спектр развлечений и обучающего контента.

В современных реалиях очень важно занимать свободное время чем-то полезным, при этом балансируя между сложными головоломками, которые требуют полной концентрации, и захватывающим сюжетом, уносящим с головой в мир тайн и мистики. В рамках практики была реализована компьютерная игра 12+, в которую можно погрузиться как в тихий вечер одному, так и устроить битву умов с друзьями.

Целью этого проекта будет свежий взгляд на жанр rogue like и привнесение в него тематики детектива вперемешку с мистикой.

# Раздел 1. Исследование предметной области

## **1.1. Описание предметной области**

**Индустрия компьютерных игр** (также [индустрия](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D0%B4%D1%83%D1%81%D1%82%D1%80%D0%B8%D1%8F) интерактивных развлечений) — [сектор экономики](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D0%BA%D1%82%D0%BE%D1%80_%D1%8D%D0%BA%D0%BE%D0%BD%D0%BE%D0%BC%D0%B8%D0%BA%D0%B8), связанный с разработкой, продвижением и продажей [компьютерных игр](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%B8%D0%B3%D1%80%D0%B0). В неё входит большое количество специальностей, по которым работают десятки тысяч человек по всему миру.

**Обзор**

Индустрия компьютерных игр зародилась в середине [1970-х](https://ru.wikipedia.org/wiki/1970-%D0%B5) годов как движение энтузиастов и за несколько десятилетий выросла из небольшого рынка в [мейнстрим](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D0%B9%D0%BD%D1%81%D1%82%D1%80%D0%B8%D0%BC) с годовой прибылью в 9,5 миллиардов долларов в США в [2007 году](https://ru.wikipedia.org/wiki/2007_%D0%B3%D0%BE%D0%B4) и 11,7 миллиардов в [2008 году](https://ru.wikipedia.org/wiki/2008_%D0%B3%D0%BE%D0%B4) (согласно ежегодным отчётам [ESA](https://ru.wikipedia.org/wiki/Entertainment_Software_Association)). На рынке работают как крупные игроки, так и небольшие фирмы и [стартапы](https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D0%B0%D1%80%D1%82%D0%B0%D0%BF), а также независимые разработчики и сообщества (напр. Kickstarter и др.).

Современные [персональные компьютеры](https://ru.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D1%81%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9_%D0%BA%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80) множеством новшеств обязаны игровой индустрии. К числу самых значимых относят [звуковые](https://ru.wikipedia.org/wiki/%D0%97%D0%B2%D1%83%D0%BA%D0%BE%D0%B2%D0%B0%D1%8F_%D0%BA%D0%B0%D1%80%D1%82%D0%B0) и [графические карты](https://ru.wikipedia.org/wiki/%D0%92%D0%B8%D0%B4%D0%B5%D0%BE%D0%BA%D0%B0%D1%80%D1%82%D0%B0), CD- и DVD-приводы и [центральные процессоры](https://ru.wikipedia.org/wiki/CPU).

[Звуковые карты](https://ru.wikipedia.org/wiki/%D0%97%D0%B2%D1%83%D0%BA%D0%BE%D0%B2%D0%B0%D1%8F_%D0%BA%D0%B0%D1%80%D1%82%D0%B0) изначально были разработаны для интегрирования качественного цифрового звука в компьютерные игры, и только потом звуковое оборудование было усовершенствовано под нужды меломанов.

[Графические карты](https://ru.wikipedia.org/wiki/%D0%92%D0%B8%D0%B4%D0%B5%D0%BE%D0%BA%D0%B0%D1%80%D1%82%D0%B0), которые на заре компьютерной эпохи эволюционировали в направлении увеличения количества поддерживаемых цветов, позже стали развиваться для аппаратной поддержки [графических интерфейсов пользователя](https://ru.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D1%84%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8F) ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) GUI) и игр. Для GUI требовалось увеличение [разрешения экрана](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D0%B7%D1%80%D0%B5%D1%88%D0%B5%D0%BD%D0%B8%D0%B5_(%D0%BA%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D0%BA%D0%B0)), а для игр — [ускорение трёхмерной графики](https://ru.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D1%84%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D1%83%D1%81%D0%BA%D0%BE%D1%80%D0%B8%D1%82%D0%B5%D0%BB%D1%8C).

Изначально [CD](https://ru.wikipedia.org/wiki/CD) и [DVD](https://ru.wikipedia.org/wiki/DVD) были разработаны как недорогой и достаточно надежный способ хранения и распространения любых данных. Впоследствии, когда эти технологии стали применяться в компьютерных играх, началось их развитие в сторону увеличения скорости чтения данных.

Современные игры — одни из самых требовательных приложений на ПК. Многие [мощные компьютеры](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B8%D0%B7%D0%B2%D0%BE%D0%B4%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%BE%D1%81%D1%82%D1%8C_%D0%BA%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%B0) покупаются геймерами, которые требуются для запуска новейших игр, в которых используются самые передовые технологии. Таким образом, игровая индустрия тесно связана с индустрией производства центральных процессоров и других компонентов ПК, так как игры зачастую требуют более высоких аппаратных мощностей, чем бизнес-приложения[1].

**Компью́терная игра**́ — [компьютерная программа](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0), служащая для организации [игрового](https://ru.wikipedia.org/wiki/%D0%98%D0%B3%D1%80%D0%B0) процесса ([геймплея](https://ru.wikipedia.org/wiki/%D0%93%D0%B5%D0%B9%D0%BC%D0%BF%D0%BB%D0%B5%D0%B9)), связи с партнёрами по игре, или сама выступающая в качестве партнёра.

В настоящее время, в ряде случаев, вместо термина компьютерная игра может использоваться [видеоигра](https://ru.wikipedia.org/wiki/%D0%92%D0%B8%D0%B4%D0%B5%D0%BE%D0%B8%D0%B3%D1%80%D0%B0), то есть данные термины могут употребляться как синонимы и быть взаимозаменяемыми. В компьютерных играх, как правило, игровая ситуация воспроизводится на экране дисплея или обычного телевизора (в этом случае компьютерные игры одновременно являются и [видеоиграми](https://ru.wikipedia.org/wiki/%D0%92%D0%B8%D0%B4%D0%B5%D0%BE%D0%B8%D0%B3%D1%80%D0%B0)), но в то же время компьютерная игра может быть [звуковой](https://ru.wikipedia.org/wiki/%D0%97%D0%B2%D1%83%D0%BA%D0%BE%D0%B2%D0%B0%D1%8F_%D0%B8%D0%B3%D1%80%D0%B0), [телетайповой](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BB%D0%B5%D1%82%D0%B0%D0%B9%D0%BF) и другой.

Компьютерные игры могут создаваться на основе [фильмов](https://ru.wikipedia.org/wiki/%D0%9A%D0%B8%D0%BD%D0%BE%D1%84%D0%B8%D0%BB%D1%8C%D0%BC) и [книг](https://ru.wikipedia.org/wiki/%D0%A5%D1%83%D0%B4%D0%BE%D0%B6%D0%B5%D1%81%D1%82%D0%B2%D0%B5%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B8%D0%B7%D0%B2%D0%B5%D0%B4%D0%B5%D0%BD%D0%B8%D0%B5); есть и обратные случаи. С 2011 года компьютерные игры официально признаны в [США](https://ru.wikipedia.org/wiki/%D0%A1%D0%A8%D0%90) отдельным [видом искусства](https://ru.wikipedia.org/wiki/%D0%98%D1%81%D0%BA%D1%83%D1%81%D1%81%D1%82%D0%B2%D0%BE).

Компьютерные игры оказали столь существенное влияние на общество, что в [информационных технологиях](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D1%8B%D0%B5_%D1%82%D0%B5%D1%85%D0%BD%D0%BE%D0%BB%D0%BE%D0%B3%D0%B8%D0%B8) отмечена устойчивая тенденция к [геймификации](https://ru.wikipedia.org/wiki/%D0%93%D0%B5%D0%B9%D0%BC%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D1%8F) для неигрового [прикладного программного обеспечения](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B8%D0%BA%D0%BB%D0%B0%D0%B4%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5).

Жанров компьютерных игр великое множество, от спокойных визуальных новелл с сюжетом, до симулятора Бога. Для каждого человека найдется что-то подходящее[2].

## **1.2. Постановка задачи**

Игра будет представлять из себя поле-карту с отмеченными на ней областями, по которым игрок сможет перемещаться и выполнять некоторые задачи, такие как поиск предметов, битвы с монстрами и т.п.

Начинается все с того, что управляемый персонаж появляется в виде фигурки на карте, отображающей маршрут путешествия с возможными развилками. Игрок выбирает одну из веток развития истории, после чего персонаж перемещается в соответствующую ей ячейку. Эта ячейка содержит определенное заранее событие, такое как внезапное происшествие, проклятие/благословление, нападение монстра, появление предмета. Эти события будут влиять на характер прохождения сюжетной линии. В ходе прохождения игрок зарабатывает специальные очки, которые он может потратить на улучшение способностей своего персонажа после его смерти.

События проклятия или благословления будут накладывать негативный или положительный эффект на персонажа соответственно, что будет сказываться на его способностях.

Система боя будет представлена в виде пошагового поединка по системе DnD с броском кубика, т.е. исход хода зависит от значения, которое выпадет в результате броска.

Найденные по ходу игры предметы, а также положительные и негативные эффекты будут влиять на результаты броска кубика, либо помогая в бою, либо ухудшая ситуацию.

Из описания вытекают следующие требования:

* наличие функциональной области-карты;
* возможность перемещения игрока по игровому пространству;
* система мета-прогресса;
* механики пошагового боя.

# Раздел 2. Обзор технологий разработки

## 2.1. Язык программирования C++

С++ — компилируемый, статически типизированный язык программирования общего назначения. Язык имеет богатую стандартную библиотеку, которая включает в себя распространённые контейнеры и алгоритмы, ввод-вывод, регулярные выражения, поддержку многопоточности и другие возможности. C++ сочетает свойства как высокоуровневых, так и низкоуровневых языков.

C++ широко используется для разработки программного обеспечения, являясь одним из самых популярных языков программирования. Область его применения включает создание [операционных систем](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0), разнообразных прикладных программ, [драйверов](https://ru.wikipedia.org/wiki/%D0%94%D1%80%D0%B0%D0%B9%D0%B2%D0%B5%D1%80) устройств, приложений для встраиваемых систем, высокопроизводительных серверов, а также игр.

Синтаксис C++ унаследован от языка C. Изначально одним из принципов разработки было сохранение совместимости с C. Тем не менее C++ не является в строгом смысле надмножеством C; множество программ, которые могут одинаково успешно транслироваться как компиляторами C, так и компиляторами C++, довольно велико, но не включает все возможные программы на C.

**Создание**

Язык возник в начале [1980-х годов](https://ru.wikipedia.org/wiki/1980-%D0%B5_%D0%B3%D0%BE%D0%B4%D1%8B), когда сотрудник фирмы [Bell Labs](https://ru.wikipedia.org/wiki/Bell_Labs) [Бьёрн Страуструп](https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D1%80%D0%B0%D1%83%D1%81%D1%82%D1%80%D1%83%D0%BF,_%D0%91%D1%8C%D1%91%D1%80%D0%BD) придумал ряд усовершенствований к языку [C](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8_(%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F)) под собственные нужды. Когда в конце [1970-х годов](https://ru.wikipedia.org/wiki/1970-%D0%B5_%D0%B3%D0%BE%D0%B4%D1%8B) Страуструп начал работать в Bell Labs над задачами теории очередей (в приложении к моделированию телефонных вызовов), он обнаружил, что попытки применения существующих в то время языков моделирования оказываются неэффективными, а применение высокоэффективных машинных языков слишком сложно из-за их ограниченной выразительности. Так, язык [Симула](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D0%BC%D1%83%D0%BB%D0%B0) имеет такие возможности, которые были бы очень полезны для разработки большого программного обеспечения, но работает слишком медленно, а язык [BCPL](https://ru.wikipedia.org/wiki/BCPL) достаточно быстр, но слишком близок к языкам низкого уровня и не подходит для разработки большого программного обеспечения.

Вспомнив опыт своей диссертации, Страуструп решил дополнить язык C (преемник BCPL) возможностями, имеющимися в языке Симула. Язык C, будучи базовым языком системы [UNIX](https://ru.wikipedia.org/wiki/UNIX), на которой работали компьютеры Bell, является быстрым, многофункциональным и переносимым. Страуструп добавил к нему возможность работы с классами и объектами. В результате практические задачи моделирования оказались доступными для решения как с точки зрения времени разработки (благодаря использованию Симула-подобных классов), так и с точки зрения времени вычислений (благодаря быстродействию C). В первую очередь в C были добавлены классы (с [инкапсуляцией](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D0%BA%D0%B0%D0%BF%D1%81%D1%83%D0%BB%D1%8F%D1%86%D0%B8%D1%8F_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5))), наследование классов, строгая проверка типов, inline-функции и [аргументы по умолчанию](https://ru.wikipedia.org/wiki/%D0%90%D1%80%D0%B3%D1%83%D0%BC%D0%B5%D0%BD%D1%82_%D0%BF%D0%BE_%D1%83%D0%BC%D0%BE%D0%BB%D1%87%D0%B0%D0%BD%D0%B8%D1%8E). Ранние версии языка, первоначально именовавшегося «C with classes» («Си с классами»), стали доступны с [1980 года](https://ru.wikipedia.org/wiki/1980_%D0%B3%D0%BE%D0%B4).

**Обзор языка**

Стандарт C++ состоит из двух основных частей: описание ядра языка и описание стандартной библиотеки.

Первое время язык развивался вне формальных рамок, спонтанно, по мере встававших перед ним задач. Развитию языка сопутствовало развитие [кросс-компилятора](https://ru.wikipedia.org/wiki/%D0%9A%D1%80%D0%BE%D1%81%D1%81-%D0%BA%D0%BE%D0%BC%D0%BF%D0%B8%D0%BB%D1%8F%D1%82%D0%BE%D1%80) [cfront](https://ru.wikipedia.org/wiki/Cfront). Новшества в языке отражались в изменении номера версии кросс-компилятора. Эти номера версий кросс-компилятора распространялись и на сам язык, но применительно к настоящему времени речь о версиях языка C++ не ведут. Лишь в 1998 году язык стал стандартизированным.

* C++ поддерживает как комментарии в стиле C (/\* **комментарий** \*/), так и однострочные (// **вся оставшаяся часть строки является комментарием**), где // обозначает начало комментария, а ближайший последующий символ новой строки, который не предварён символом \ (либо эквивалентным ему обозначением ??/), считается окончанием комментария. Плюс этого комментария в том, что его не обязательно заканчивать, то есть обозначать окончание комментария;
* Спецификатор inline для функций. Функция, определённая внутри тела класса, является inline по умолчанию. Данный спецификатор является подсказкой компилятору и может встроить тело функции в код вместо её непосредственного вызова;
* Квалификаторы const и volatile. В отличие от С, где const обозначает только доступ на чтение, в C++ переменная с квалификатором const должна быть инициализирована. volatile используется в описании переменных и информирует компилятор, что значение данной переменной может быть изменено способом, который компилятор не в состоянии отследить. Для переменных, объявленных volatile, компилятор не должен применять средства оптимизации, изменяющие положение переменной в памяти (например, помещающие её в регистр) или полагающиеся на неизменность значения переменной в промежутке между двумя присваиваниями ей значения. В многоядерной системе volatile помогает избегать [барьеров памяти](https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D1%80%D1%8C%D0%B5%D1%80_%D0%BF%D0%B0%D0%BC%D1%8F%D1%82%D0%B8) 2-го типа;
* [Пространства имён](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%81%D1%82%D1%80%D0%B0%D0%BD%D1%81%D1%82%D0%B2%D0%BE_%D0%B8%D0%BC%D1%91%D0%BD_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)#C++) (namespace)[3].

## 2.2. Blender

Blender — профессиональное [свободное и открытое программное обеспечение](https://ru.wikipedia.org/wiki/%D0%A1%D0%B2%D0%BE%D0%B1%D0%BE%D0%B4%D0%BD%D0%BE%D0%B5_%D0%B8_%D0%BE%D1%82%D0%BA%D1%80%D1%8B%D1%82%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5) для создания [трёхмерной компьютерной графики](https://ru.wikipedia.org/wiki/%D0%A2%D1%80%D1%91%D1%85%D0%BC%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D0%BA%D0%B0), включающее в себя средства моделирования, скульптинга, анимации, симуляции, [рендеринга](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BD%D0%B4%D0%B5%D1%80%D0%B8%D0%BD%D0%B3), постобработки и монтажа видео со звуком, компоновки с помощью «узлов» ([Node Compositing](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D0%BE%D0%B7%D0%B8%D1%82%D0%B8%D0%BD%D0%B3)), а также создания 2D-анимаций.

**Возможности**

Характерной особенностью пакета Blender выступает его небольшой размер по сравнению с другими популярными пакетами для 3D-моделирования. Документация в поставку не входит, но доступна онлайн. Демонстрационные сцены можно скачать на официальном сайте или на сайте открытых проектов «Blender Cloud».

Функции пакета:

* Поддержка разнообразных геометрических примитивов, включая [полигональные](https://ru.wikipedia.org/wiki/%D0%9C%D0%BD%D0%BE%D0%B3%D0%BE%D1%83%D0%B3%D0%BE%D0%BB%D1%8C%D0%BD%D0%B8%D0%BA) модели, систему быстрого моделирования в режиме [subdivision surface](https://ru.wikipedia.org/w/index.php?title=Subdivision_surface&action=edit&redlink=1) (SubSurf), [кривые Безье](https://ru.wikipedia.org/wiki/%D0%9A%D1%80%D0%B8%D0%B2%D1%8B%D0%B5_%D0%91%D0%B5%D0%B7%D1%8C%D0%B5), [поверхности NURBS](https://ru.wikipedia.org/wiki/NURBS), [metaballs](https://ru.wikipedia.org/wiki/Metaballs) (метасферы), [скульптурное моделирование](https://ru.wikipedia.org/wiki/%D0%A6%D0%B8%D1%84%D1%80%D0%BE%D0%B2%D0%B0%D1%8F_%D1%81%D0%BA%D1%83%D0%BB%D1%8C%D0%BF%D1%82%D1%83%D1%80%D0%B0) и [векторные](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%BA%D1%82%D0%BE%D1%80%D0%BD%D0%B0%D1%8F_%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D0%BA%D0%B0) шрифты;
* Универсальные встроенные механизмы [рендеринга](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BD%D0%B4%D0%B5%D1%80%D0%B8%D0%BD%D0%B3) и интеграция с внешними рендерерами [YafRay](https://ru.wikipedia.org/wiki/YafRay), [LuxRender](https://ru.wikipedia.org/wiki/LuxRender) и многими другими;
* Инструменты [анимации](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B8%D0%BC%D0%B0%D1%86%D0%B8%D1%8F_(%D0%B8%D1%81%D0%BA%D1%83%D1%81%D1%81%D1%82%D0%B2%D0%BE)), среди которых [инверсная кинематика](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D0%B2%D0%B5%D1%80%D1%81%D0%BD%D0%B0%D1%8F_%D0%BA%D0%B8%D0%BD%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0), [скелетная анимация](https://ru.wikipedia.org/wiki/%D0%A1%D0%BA%D0%B5%D0%BB%D0%B5%D1%82%D0%BD%D0%B0%D1%8F_%D0%B0%D0%BD%D0%B8%D0%BC%D0%B0%D1%86%D0%B8%D1%8F) и сеточная деформация, [анимация по ключевым кадрам](https://ru.wikipedia.org/wiki/%D0%9C%D1%83%D0%BB%D1%8C%D1%82%D0%B8%D0%BF%D0%BB%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D1%8F_%D0%BF%D0%BE_%D0%BA%D0%BB%D1%8E%D1%87%D0%B5%D0%B2%D1%8B%D0%BC_%D0%BA%D0%B0%D0%B4%D1%80%D0%B0%D0%BC), нелинейная анимация, редактирование весовых коэффициентов вершин, ограничители;
* Динамика мягких тел (включая определение коллизий объектов при взаимодействии), динамика твёрдых тел на основе физического движка [Bullet](https://ru.wikipedia.org/wiki/Bullet_Physics_Library);
* [Система частиц](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%87%D0%B0%D1%81%D1%82%D0%B8%D1%86) включающая в себя систему волос на основе частиц;
* Модификаторы для применения неразрушающих эффектов;
* Язык программирования [Python](https://ru.wikipedia.org/wiki/Python) используется как средство определения [интерфейса](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81), создания инструментов и прототипов, системы логики в играх, как средство импорта/экспорта файлов (например, [COLLADA](https://ru.wikipedia.org/wiki/COLLADA)), автоматизации задач;
* Базовые функции [нелинейного](https://ru.wikipedia.org/wiki/%D0%9D%D0%B5%D0%BB%D0%B8%D0%BD%D0%B5%D0%B9%D0%BD%D1%8B%D0%B9_%D0%BC%D0%BE%D0%BD%D1%82%D0%B0%D0%B6) видео и аудио монтажа;
* [Композитинг](https://ru.wikipedia.org/wiki/%D0%A6%D0%B8%D1%84%D1%80%D0%BE%D0%B2%D0%BE%D0%B9_%D0%BA%D0%BE%D0%BC%D0%BF%D0%BE%D0%B7%D0%B8%D1%82%D0%B8%D0%BD%D0%B3) видео, работа с [хромакеем](https://ru.wikipedia.org/wiki/%D0%A5%D1%80%D0%BE%D0%BC%D0%B0%D0%BA%D0%B5%D0%B9);
* [Трекинг](https://ru.wikipedia.org/wiki/%D0%A2%D1%80%D0%B5%D0%BA%D0%B8%D0%BD%D0%B3_(%D0%B2%D0%B8%D1%80%D1%82%D1%83%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%80%D0%B5%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D1%81%D1%82%D1%8C)) камеры и объектов;
* [Real-time](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D0%B5_%D0%B2%D1%80%D0%B5%D0%BC%D1%8F) контроль во время физической симуляции и рендеринга;
* [Процедурное](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%86%D0%B5%D0%B4%D1%83%D1%80%D0%BD%D0%BE%D0%B5_%D1%82%D0%B5%D0%BA%D1%81%D1%82%D1%83%D1%80%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) и node-based текстурирование, а также возможность рисовать текстуру прямо на модели;
* Grease Pencil — инструмент для 2D-анимации в полном 3D-пайплайне;
* [Blender Game Engine](https://ru.wikipedia.org/wiki/Blender_Game_Engine) — подпроект Blender, предоставляющий интерактивные функции, такие как [определение коллизий](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D0%BD%D0%B0%D1%80%D1%83%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5_%D1%81%D1%82%D0%BE%D0%BB%D0%BA%D0%BD%D0%BE%D0%B2%D0%B5%D0%BD%D0%B8%D0%B9), движок динамики и программируемая логика. Также он позволяет создавать отдельные [real-time](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D0%B5_%D0%B2%D1%80%D0%B5%D0%BC%D1%8F)-приложения начиная от архитектурной визуализации до видео игр. Удалён в версии 2.8.

**Дополнительные особенности**

* В программе Blender сущность, взаимодействующая с окружающим миром, и её данные (форма или функции объекта) разделяемы. Отношение Объект-Данные представляется отношением 1:n (термин, относящийся к теории баз данных, обозначает возможность нескольких объектов использовать одни и те же данные — один ко многим или [сюръекция](https://ru.wikipedia.org/wiki/%D0%A1%D1%8E%D1%80%D1%8A%D0%B5%D0%BA%D1%86%D0%B8%D1%8F));
* Внутренняя файловая система, позволяющая хранить несколько сцен в едином файле (называемом .blend-файл);
* Все «.blend»-файлы совместимы как с более старыми, так и с более новыми версиями Blender. Также все они переносимы с одной платформы на другую и могут использоваться как средство переноса созданных ранее работ;
* Blender делает резервные копии проектов во время всей работы программы, что позволяет сохранить данные при непредвиденных обстоятельствах;
* Все сцены, объекты, материалы, текстуры, звуки, изображения, [постпродакшн](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D1%81%D1%82%D0%BF%D1%80%D0%BE%D0%B4%D0%B0%D0%BA%D1%88%D0%BD)-эффекты могут быть сохранены в единый «.blend»-файл;
* Настройки рабочей среды могут быть сохранены в «.blend»-файл, благодаря чему при загрузке файла пользователь получит именно то, что сохранил в него. Файл можно сохранить как «пользовательский по умолчанию», и каждый раз при запуске Blender будет выдаваться необходимый набор объектов и подготовленный к работе интерфейс.

Тем не менее, внутреннее содержание «.blend»-файла менее похоже на структурированное описание объектов и их взаимоотношений, и более близко к прямому дампу области памяти программы. Это делает практически невозможным преобразование «.blend»-файлов в другие форматы. При этом следует заметить весьма продвинутый механизм экспорта в разнообразные форматы, такие как [obj](https://ru.wikipedia.org/wiki/Obj), [dxf](https://ru.wikipedia.org/wiki/DXF), [stl](https://ru.wikipedia.org/wiki/STL_%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82), [3ds](https://ru.wikipedia.org/w/index.php?title=3ds&action=edit&redlink=1) и прочие (список постепенно растёт)[4].

## 2.3. Unreal Engine

Unreal Engine — [игровой движок](https://ru.wikipedia.org/wiki/%D0%98%D0%B3%D1%80%D0%BE%D0%B2%D0%BE%D0%B9_%D0%B4%D0%B2%D0%B8%D0%B6%D0%BE%D0%BA), разрабатываемый и поддерживаемый компанией [Epic Games](https://ru.wikipedia.org/wiki/Epic_Games).

Написанный на языке [C++](https://ru.wikipedia.org/wiki/C%2B%2B), движок позволяет создавать игры для большинства [операционных систем](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0) и [платформ](https://ru.wikipedia.org/wiki/%D0%90%D0%BF%D0%BF%D0%B0%D1%80%D0%B0%D1%82%D0%BD%D0%B0%D1%8F_%D0%BF%D0%BB%D0%B0%D1%82%D1%84%D0%BE%D1%80%D0%BC%D0%B0): [Microsoft Windows](https://ru.wikipedia.org/wiki/Microsoft_Windows), [Linux](https://ru.wikipedia.org/wiki/Linux), [Mac OS](https://ru.wikipedia.org/wiki/Mac_OS) и [Mac OS X](https://ru.wikipedia.org/wiki/Mac_OS_X); [консолей](https://ru.wikipedia.org/wiki/%D0%98%D0%B3%D1%80%D0%BE%D0%B2%D0%B0%D1%8F_%D0%BF%D1%80%D0%B8%D1%81%D1%82%D0%B0%D0%B2%D0%BA%D0%B0) [Xbox](https://ru.wikipedia.org/wiki/Xbox), [Xbox 360](https://ru.wikipedia.org/wiki/Xbox_360), [Xbox One](https://ru.wikipedia.org/wiki/Xbox_One), [PlayStation 2](https://ru.wikipedia.org/wiki/PlayStation_2), [PlayStation 3](https://ru.wikipedia.org/wiki/PlayStation_3), [PlayStation 4](https://ru.wikipedia.org/wiki/PlayStation_4), PlayStation 5, [PSP](https://ru.wikipedia.org/wiki/PlayStation_Portable), [PS Vita](https://ru.wikipedia.org/wiki/PS_Vita), [Wii](https://ru.wikipedia.org/wiki/Wii), [Dreamcast](https://ru.wikipedia.org/wiki/Dreamcast), [GameCube](https://ru.wikipedia.org/wiki/Nintendo_GameCube) и др., а также на различных портативных устройствах, например, устройствах [Apple](https://ru.wikipedia.org/wiki/Apple) ([iPad](https://ru.wikipedia.org/wiki/IPad), [iPhone](https://ru.wikipedia.org/wiki/IPhone)), управляемых системой [iOS](https://ru.wikipedia.org/wiki/IOS) и прочих.

**Иерархия объектов**

Все элементы игрового движка представлены в виде [объектов](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)), имеющих набор характеристик, и [класса](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B0%D1%81%D1%81_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)), который определяет доступные характеристики. В свою очередь, любой класс является «дочерним» классом **object**. Среди основных классов и объектов можно выделить следующие:

* Актёр (**actor**) (жаргонное выражение, принятое в среде 3D-моделлеров, разработчиков, заменяющее выражение «действующий объект» или «субъект» — являющееся переводом [англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) actor — тот, кто действует — [англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) action) — родительский класс, содержащий все объекты, которые имеют отношение к игровому процессу и имеют пространственные координаты;
* Пешка (**pawn**) — физическая модель игрока или объекта, управляемого искусственным интеллектом. Название происходит от [англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) pawn — тот, кем манипулируют (или пешка, поэтому такой объект без какой-либо модели выглядит как [пешка](https://ru.wikipedia.org/wiki/%D0%9F%D0%B5%D1%88%D0%BA%D0%B0)). Метод управления описан специальным объектом, такой объект называется контроллером. Контроллер искусственного интеллекта описывает лишь общее поведение пешки во время игрового процесса, а такие параметры как «здоровье» (количество повреждений, после которых пешка перестаёт функционировать) или, например, расстояние, на котором пешка обращает внимание на звуки, задаются для каждого объекта отдельно;
* Мир, уровень (**world**, **game level**) — объект, характеризующий общие свойства «пространства», например, силу тяжести и туман, в котором располагаются все «актёры». Также может содержать в себе параметры игрового процесса, как, например, игровой режим, для которого предназначен уровень.

Для работы с простыми и, как правило, неподвижными элементами игрового пространства (например, стены) используется [двоичное разбиение пространства](https://ru.wikipedia.org/wiki/%D0%94%D0%B2%D0%BE%D0%B8%D1%87%D0%BD%D0%BE%D0%B5_%D1%80%D0%B0%D0%B7%D0%B1%D0%B8%D0%B5%D0%BD%D0%B8%D0%B5_%D0%BF%D1%80%D0%BE%D1%81%D1%82%D1%80%D0%B0%D0%BD%D1%81%D1%82%D0%B2%D0%B0) — всё пространство делится на «заполненное» и «пустое». В «пустой» части пространства располагаются все объекты, а также только в ней может находиться «точка наблюдения» при отрисовке сцены. Возможность полного или частичного помещения объектов в «заполненную» часть пространства не исключается, однако может привести к неправильной обработке таких объектов (например, расчёт физического взаимодействия) или неправильной отрисовки в случае помещения туда «точки наблюдения» (например, эффект «зала зеркал» на рисунке 1). Все пешки, попадающие в «заполненную» часть пространства, сразу «погибают».

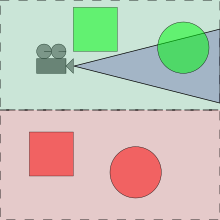


Рисунок 1. Эффект «зала зеркал»

Зонирование. В камеру не попадает ни один портал (пунктирная линия) красной зоны, поэтому объекты в ней не обрабатываются вовсе.

Поверхность (**surface**) является основным элементом двоичного дерева пространства. Эти элементы создаются на грани пересечения между «заполненной» и «пустой» частями пространства. Группа элементов двоичного дерева пространства называется нодом (**node**, [рус.](https://ru.wikipedia.org/wiki/%D0%A0%D1%83%D1%81%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) узел). Этот термин, как правило, употребляется в контексте node count — количество нодов на экране или в игровом пространстве вообще. Количество нодов, одновременно видимых на экране, влияет на производительность при прорисовке сцены. Если какой-то нод не попадает на экран или перекрывается целиком другими нодами, он не обсчитывается — это служит для повышения производительности, особенно в закрытых пространствах. Разбиение всего пространства на группы нодов называется зонированием.

Для этого иногда используются порталы — невидимые поверхности, которые служат для того, чтобы вручную разделить крупный нод на два меньших (в версии движка Unreal Engine 3 ввели поддержку аддитивной геометрии, что позволило отказаться от зонирования). Кроме порталов, используются антипорталы.

Описание «заполненных» и «пустых» частей пространства выполняется с помощью набора замкнутых трёхмерных объектов, составленных из непересекающихся поверхностей — брашей (**brush**, [рус.](https://ru.wikipedia.org/wiki/%D0%A0%D1%83%D1%81%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) кисть). Этот принцип построения пространства называется [конструктивной сплошной геометрией](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BD%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%82%D0%B8%D0%B2%D0%BD%D0%B0%D1%8F_%D1%81%D0%BF%D0%BB%D0%BE%D1%88%D0%BD%D0%B0%D1%8F_%D0%B3%D0%B5%D0%BE%D0%BC%D0%B5%D1%82%D1%80%D0%B8%D1%8F). Геометрия может быть «аддитивной» (всё пространство изначально «пустое») и «вычитательной» (изначально заполненное материей пространство).

Браши делятся на три типа:

* сплошные (**solid**) — полноценно участвуют в двоичном разбиении пространства;
  + аддитивные (**additive**) — «заполняют» двоичное пространство;
  + вычитательные (**substractive**) — «вырезают» объёмы в пространстве.
* полусплошные (**semi-solid**) — не влияют напрямую на двоичное дерево пространства, однако влияют на её физическую модель. Могут только «заполнять» пространство. Служат для создания «невидимых» препятствий, а также снижения числа полигонов и нодов;
* пустые (**non-solid**) — только создают поверхности, не влияют на двоичное дерево пространства. Используются преимущественно для создания объёмов (**volume**) — часть пространства, которая обладает свойствами, отличными от свойств игрового мира. Объёмы имеют приоритет, свойства объёма с большим приоритетом применяются к находящимся в нём актёрам. Игровой мир всегда имеет минимальный приоритет. При помощи объёмов можно изменить гравитацию, вязкость, туман и тому подобное. Объёмы, начиная с версии движка Unreal Engine 2, используются для создания воды (но не водной поверхности)[5];

## **2.4. Blueprints**

Система визуальных сценариев Blueprint в Unreal Engine — это полноценная система сценариев игрового процесса, основанная на концепции использования интерфейса на основе узлов для создания элементов игрового процесса из Unreal Editor. Как и многие распространенные языки сценариев, он используется для определения объектно-ориентированных (ОО) классов или объектов в движке.

Эта система чрезвычайно гибкая и мощная, поскольку она дает разработчикам возможность использовать практически весь спектр концепций и инструментов, обычно доступных только программистам. Кроме того, специфичная для Blueprint разметка, доступная в реализации C++ Unreal Engine, позволяет программистам создавать базовые системы, которые могут быть расширены дизайнерами.

Означает ли это, что Blueprints являются заменой UnrealScript? Да и нет. Программирование геймплея и все, для чего раньше использовался UnrealScript, по-прежнему можно обрабатывать с помощью кода на C++. В то же время, хотя Blueprints не предназначены для замены UnrealScript, они служат многим из тех же целей, что и UnrealScript, например:

* Расширение классов;
* Сохранение и изменение свойств по умолчанию;
* Управление экземплярами подобъектов (например, компонентов) для классов.

Ожидается, что программисты геймплея установят базовые классы, предоставляющие полезный набор функций и свойств, которые могут использоваться и расширяться в Blueprints, созданных на основе этих базовых классов.

**Типы чертежей**

Чертежи могут быть одним из нескольких типов, каждый из которых имеет свое собственное конкретное использование: от создания новых типов до сценариев уровня событий до определения интерфейсов или макросов, которые будут использоваться другими блупринтами.

**Класс чертежей**

Класс Blueprint, часто сокращенно Blueprint , — это актив, который позволяет создателям контента легко добавлять функциональные возможности поверх существующих игровых классов. Чертежи создаются внутри Unreal Editor визуально, а не путем ввода кода, и сохраняются как активы в пакете контента. По сути, они определяют новый класс или тип Актера, который затем может быть помещен в карты как экземпляры, ведущие себя как любой другой тип Актера.

**Схема уровня**

Чертеж уровня — это особый тип Чертежа , который действует как глобальный график событий уровня. Каждый уровень в вашем проекте имеет свою собственную схему уровня, созданную по умолчанию, которую можно редактировать в редакторе Unreal Editor, однако новые схемы уровней нельзя создавать через интерфейс редактора.

События, относящиеся к уровню в целом или конкретным экземплярам Актеров на уровне, используются для запуска последовательностей действий в форме вызовов функций или операций управления потоком. Те, кто знаком с Unreal Engine 3, должны быть хорошо знакомы с этой концепцией, поскольку она очень похожа на то, как Kismet работал в Unreal Engine 3.

Чертежи уровней также предоставляют механизм управления для потоковой передачи уровней и [Sequencer](https://docs.unrealengine.com/5.0/en-US/real-time-compositing-with-sequencer-in-unreal-engine) , а также для привязки событий к Актерам, размещенным на уровне[6].

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Wikipedia [Электронный ресурс]. — [Б. м. : б. и.]. — URL: https://ru.wikipedia.org/wiki/Индустрия\_компьютерных\_игр (дата обращения: 03.06.2022).
2. Wikipedia [Электронный ресурс]. — [Б. м. : б. и.]. — URL: https://ru.wikipedia.org/wiki/Компьютерная\_игра (дата обращения: 03.06.2022).
3. Wikipedia [Электронный ресурс]. — [Б. м. : б. и.]. — URL: https://ru.wikipedia.org/wiki/C%2B%2B (дата обращения: 03.06.2022).
4. Wikipedia [Электронный ресурс]. — [Б. м. : б. и.]. — URL: https://ru.wikipedia.org/wiki/Blender (дата обращения: 03.06.2022).
5. Wikipedia [Электронный ресурс]. — [Б. м. : б. и.]. — URL: https://ru.wikipedia.org/wiki/Unreal\_Engine (дата обращения: 03.06.2022).
6. Unreal Engine 5.0 Documentation [Электронный ресурс]. — [Б. м. : б. и.]. — URL: https://docs.unrealengine.com/5.0/en-US/overview-of-blueprints-visual-scripting-in-unreal-engine/ (дата обращения: 03.06.2022).