**IT-Колледж “Сириус”**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**ДОКЛАД**

по дисциплине “Введение в специальность”

на тему “Сравнение и обзор движков для создания игр”

Выполнил:  
Студент группы

1.9.7.1

Осипов Владислав Антонович

Принял:

Старший преподаватель  
Тенигин Альберт Андреевич

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

IT-Колледж “Сириус”  
2022

**Содержание**

[I. Введение 3](#_Toc118123496)

[II. Основная часть 5](#_Toc118123497)

[1. Unity 5](#_Toc118123498)

[2. Unreal Engine 4 7](#_Toc118123499)

[3. GameMaker Studio 10](#_Toc118123500)

[4. Godot 12](#_Toc118123501)

[III. Заключение 14](#_Toc118123502)

[Список литературы 16](#_Toc118123503)

# I. Введение

Какой игровой движок вы вспоминаете первым при слове GameDev? Скорее всего, это Unity или Unreal Engine. Но знали ли вы о других движках, таких как GameMaker, Godot? И какой из них всё-таки выбрать? Об этом я сегодня и хочу вам рассказать.

Вообще, для начала разберёмся с понятием игрового движка. Игровой движок — это программный комплекс, который упрощает разработку игр, предоставляя вам набор необходимых для разработки инструментов. Из этого следует несколько простых фактов. Во-первых, движок совершенно необязателен, игру можно написать и без него на голом языке программирования. Во-вторых, движок не сделает крутую игру за вас. Но с ним работа пойдёт в десятки раз быстрее, так что лучше не писать велосипеды, а использовать готовое [1].

Обобщённо говоря, игровой движок ответственен за организацию и поведение игровых объектов, а также за их отображение на экране. Ваша же задача — выбрать, как они будут выглядеть и как себя вести. Для этого движок предоставит вам возможность создавать и удалять объекты, задавать их параметры, добавлять логику и управлять ресурсами.

Касательно **внутреннего устройства** игровые движки делятся на:

1. Полноценное ПО — это такая специальная программа, которая может собрать игру воедино за вас, а вам предоставляет полноценный редактор.
2. Фреймворк — это набор полезных классов и функций, которые вы можете использовать при создании своей игры с нуля. Игру вам придётся собирать самостоятельно, но зато у вас будет больше контроля за процессом.

Игровые движки могут предоставлять или не предоставлять различные **возможности**, такие как:

* Система уровней
* Физика
* Система частиц
* Навигация
* Двумерная анимация
* Трёхмерная анимация
* Карты тайлов
* Редактор местности (terrain)
* Воспроизведение звука
* Источники освещения и тени
* Шейдеры
* Диалоговая система [6].

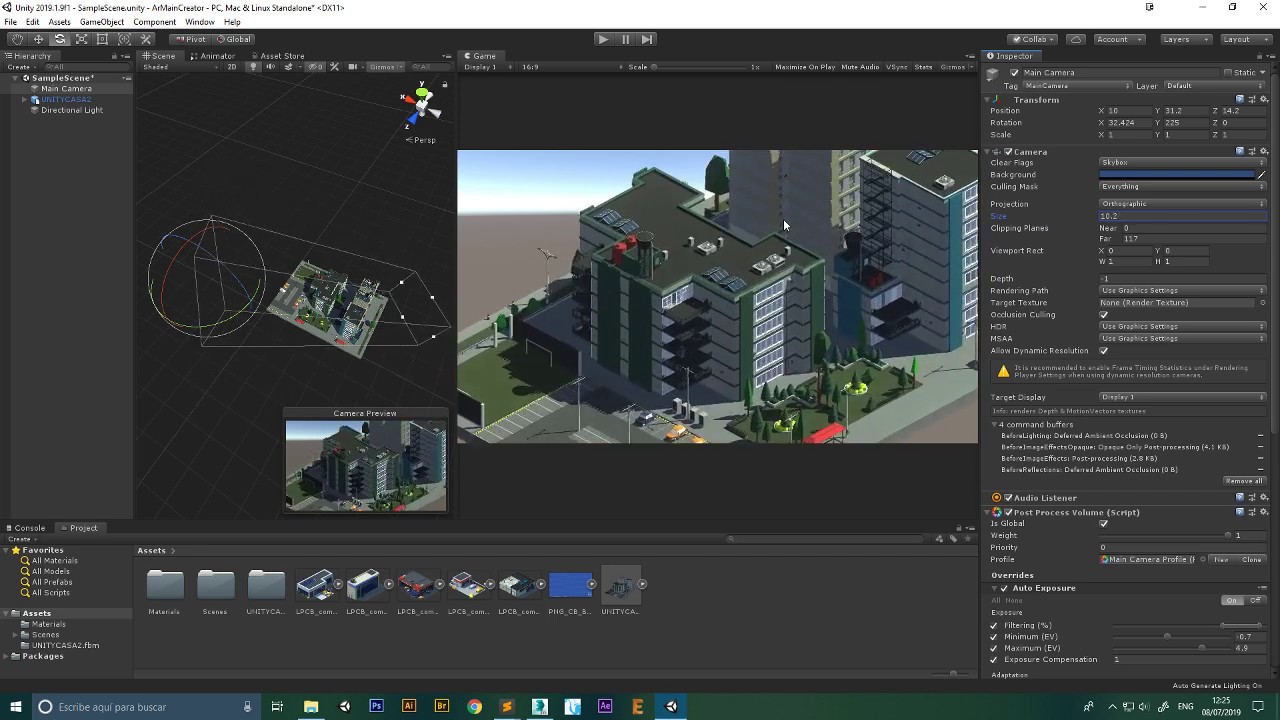
А теперь я хотел бы рассказать о популярных игровых движках.

# II. Основная часть

## **1. Unity**

Начнём, пожалуй, с Unity. Это самый популярный кросс-платформенный игровой движок на сегодняшний день. Он подходит для разработки 2D и 3D игр любого жанра и формата.

Появился Unity в 2005 году. Большое распространение получил с выходом версии Unity3D. На этом движке сделано большое количество мобильных игр и игр в Steam, в том числе и инди. Например, на Unity написаны такие популярные игры, как Heartstone, Pokemon GO, Cuphead, Subnautica, Hollow Knight, Escape From Tarkov и т. д.



*Рисунок 1. Окно программы «Unity»*

У Unity самое большое количество разработчиков, у большинства компаний процессы заточены именно под Unity (и под используемый там язык C#). Кроме того, готовых ассетов у движка примерно столько же, сколько суммарно у всех других рассматриваемых в этом докладе. Ассеты - это готовые решения: 3d модели, картинки, уровни и даже целые игры, которые можно использовать в своей игре [2].

Но при этом это и самый критикуемый движок. Причём соотношение критикуемых выше среднего по другим движкам.

На Unity можно сделать игры практически под любую платформу. Но из-за огромного объёма всего, что там есть, вылезают ошибки. И правят их не очень быстро. В приоритете премиальный сегмент пользователей [4].

**Плюсы Unity:**

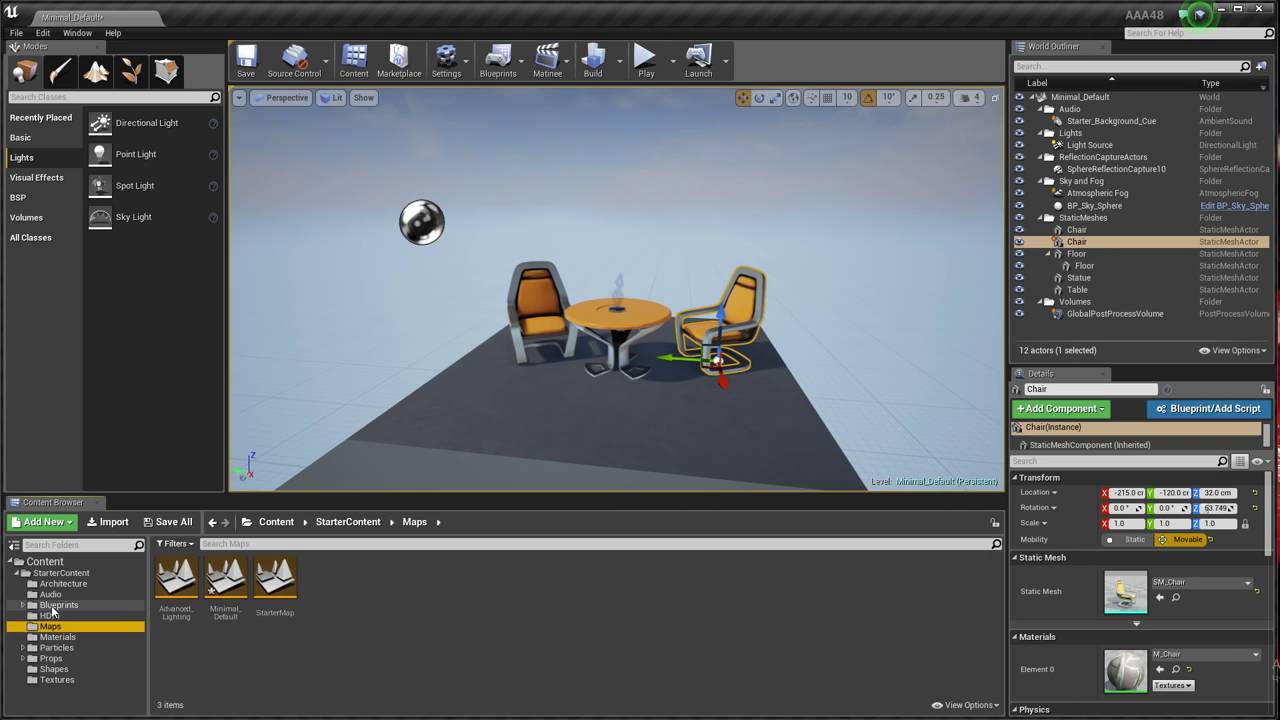
* Гибкий и расширяемый движок. Множество компонентов для работы из коробки;
* Бесплатная версия, если оборот не превышает 100 000 $;
* Огромная база знаний, самое большое комьюнити;
* Большое количество шаблонов, ассетов и примеров работ;
* Самый большой выбор обучающих материалов, курсов и книг;
* Простота в разработке за счет C# и Bolt, доступных и понятных;
* Движок имеет большие возможности как в 2d, так и в 3d;
* Мультиплатформенность;
* Огромное количество успешных кейсов;
* Встроенные реклама и аналитика от самой Unity.

**Минусы Unity:**

* Для разработки хорошей игры нужно глубоко погрузиться в аспекты движка;
* Даже при наличии инструмента визуального программирования Playmaker и большого количества ассетов, для новичков движок будет слишком сложный;
* Много сырых версий и ошибки, которые иногда тянутся из версии в версию;
* Отсутствие открытого кода для небольших команд разработчиков;
* Файлы на выходе весят много. Из-за чего его мало используют в вебе;
* В приоритете разработки обновлений движка - мобильные платформы;
* Модель поддержки Unity нацелена на бизнес, а не на удобство для разработчиков;
* 3d графика неплоха, но все-таки прилично уступает Unreal Engine [3,4].

## **2. Unreal Engine 4**

Следующим я хотел бы разобрать не менее популярный игровой движок – Unreal Engine 4. Его история длится десятилетиями, однако сейчас - он кажется достиг своего величия. Движок разрабатывается компанией Epic Games и является передовым решением для создания крупных AAA-игр [3].



*Рисунок 2. Окно программы «Unreal Engine 4»*

Легендарный движок повлиявший на всю индустрию шутеров впервые был использован в 1998 году в игре Unreal. Впоследствии много раз использовался для игр ААА-класса: Deus Ex, Clive Barker's Undying, Unreal, Tom Clancy's Splinter Cell, Gears of War, BioShock, Mass Effect, серия Batman: Arkham, Medal of Honor, Mortal Kombat, Street Fighter 5, Tekken 7, XCOM, Fornite и др.

С 2015 года бесплатный для использования. Если игра приносит более 3000$ за квартал, производители берут 5% прибыли.

Язык программирования - C++, который считается сложным в изучении и в работе. Но и одним из самых производительных. Но помимо него есть система визуального программирования Blueprints из коробки, и можно создавать игры, не написав и строчки кода. Этим и занимаются в основном инди-разработчики на Unreal, которые не хотят связываться с C++. Однако надо учесть только проблемы с производительностью. Unreal Engine предоставляет красивую графику, шикарные эффекты, отличное моделирование и дизайн уровней. Движок популярен у 3d-художников, моделлеров и левел-дизайнеров. Кроме того, отличный набор готовых ассетов и туториалы на самые разные темы.

Ругают его обычно за низкую производительность, высокий порог входа и периодические баги. Используют в основном для 3d игр на компьютеры и игровые приставки. Но с ростом технологий часто стали встречаться игры на этом движке и на мобильных. Для 2d-игр есть специальная система 2d Paper [4].

**Плюсы Unreal Engine:**

* Готовый к AAA-проектам из коробки
* Есть встроенный скриптинг Blueprints;
* Большое комьюнити и большое количество туториалов;
* Мощный редактор на все случаи жизни;
* Гибкая архитектура игрового движка;
* Хорошие инструменты для моделирования уровней;
* Отличный набор готовых ассетов. Меньше чем у Unity, но все очень качественные;
* Игровой движок разрабатывается в том числе для игр создателя. Поэтому он в первую очередь нацелен на других разработчиков, а не на бизнес, как в случае с Unity;
* Отличная графика;
* Бесплатный;
* Кросс-платформенный.

**Минусы Unreal Engine:**

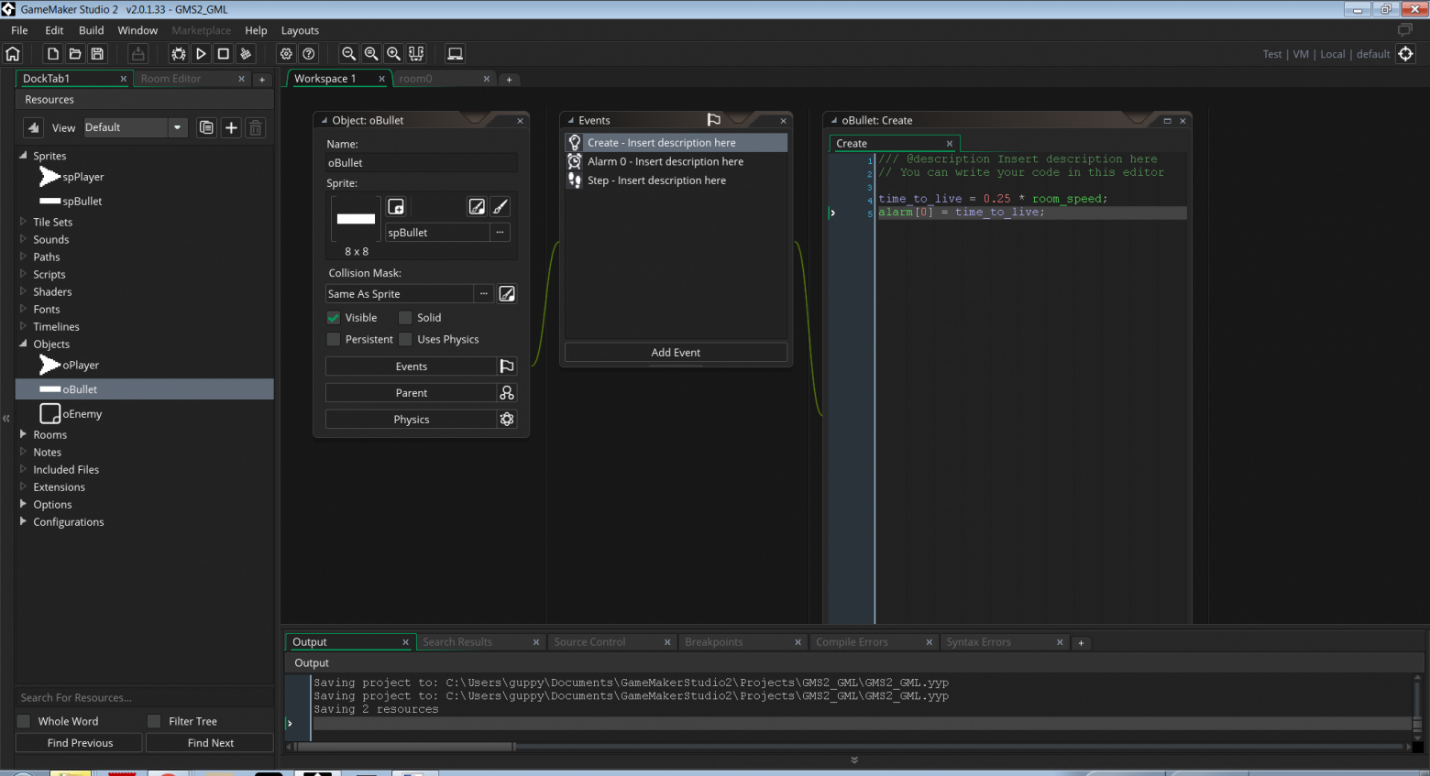
* Занимает много места сама программа и проекты на ней;
* Высокий порог входа, который частично снижен визуальным программированием;
* Интерфейс достаточно громоздкий с десятками элементов. Хотя если привыкнуть - логичнее чем в Unity;
* Более закрытое и не такое многочисленное сообщество;
* Акцент - на AAA-проекты;
* Размер движка и его требовательность;
* Замечены проблемы с производительностью [3,4].

## **3. GameMaker Studio**

Прародителем этого движка был gamemaker, который появился ещё в 1999 году. Изначально миссией компании YoYo Games было предоставление простого конструктора для игр и долгое время продукт был бесплатен.

Позже появились спонсоры Playtech, которые купили YoYo Games. Результатом в 2017 году стала вторая версия известного движка [4]. Летом 2021 года YoYo Games сделали беплатный доступ без ограничений по времени. Изначально они выдавали trial-версию только на месяц, для знакомства. Правда экспортировать на бесплатной версии можно разве что на GXC. В базовой подписке CREATOR (≈ 80 руб./мес.) появляется экспорт для Windows, macOS и Linux. В подписке INDIE (≈ 260 руб./мес.) появляется экспорт для HTML5 и мобильных устройств.

Интерфейс Game Maker объединяет в себе редакторы спрайтов, объектов, комнат, скриптов, а также тайм-лайнов (последовательностей действий с привязкой по времени), путей (маршрутов) движения и констант.



*Рисунок 3. Окно программы «GameMaker Studio»*

Проблема GameMaker, и других движков типа «наведи и щелкни» в том, что разработчики более ограничены. С другой стороны, GameMaker популярен, потому что позволяет создать игру без предварительных навыков программирования [5].

Gamemaker прост в освоении и при этом весьма эффективный. На движке создано немало инди-хитов, некоторые из них перешагнули миллионные отметки скачиваний в онлайн-сервисе Steam. Такие известные игры как: Hotline Miami, Nuclear Throne, Undertale, Hyper Light Drifter, The Swords of Ditto, Forager, недавний Katana Zero и многие другие.

Это в основном инди-2d игры под Steam, мобильные устройства или игровые приставки. Создаются в основном при помощи визуального программирования drag-n-drop. Есть возможность программировать на собственном совсем простом языке gml, напоминающим javascript [4].

**Плюсы GameMaker Studio:**

* Низкий порог вхождения, не обязательно знание программирования;
* Большое и активное комьюнити;
* Не слишком современный и красивый, но при этом очень удобный интерфейс;
* Высокая скорость разработки;
* Большое количество успешных игр на платформе;
* Есть своя аналитика.

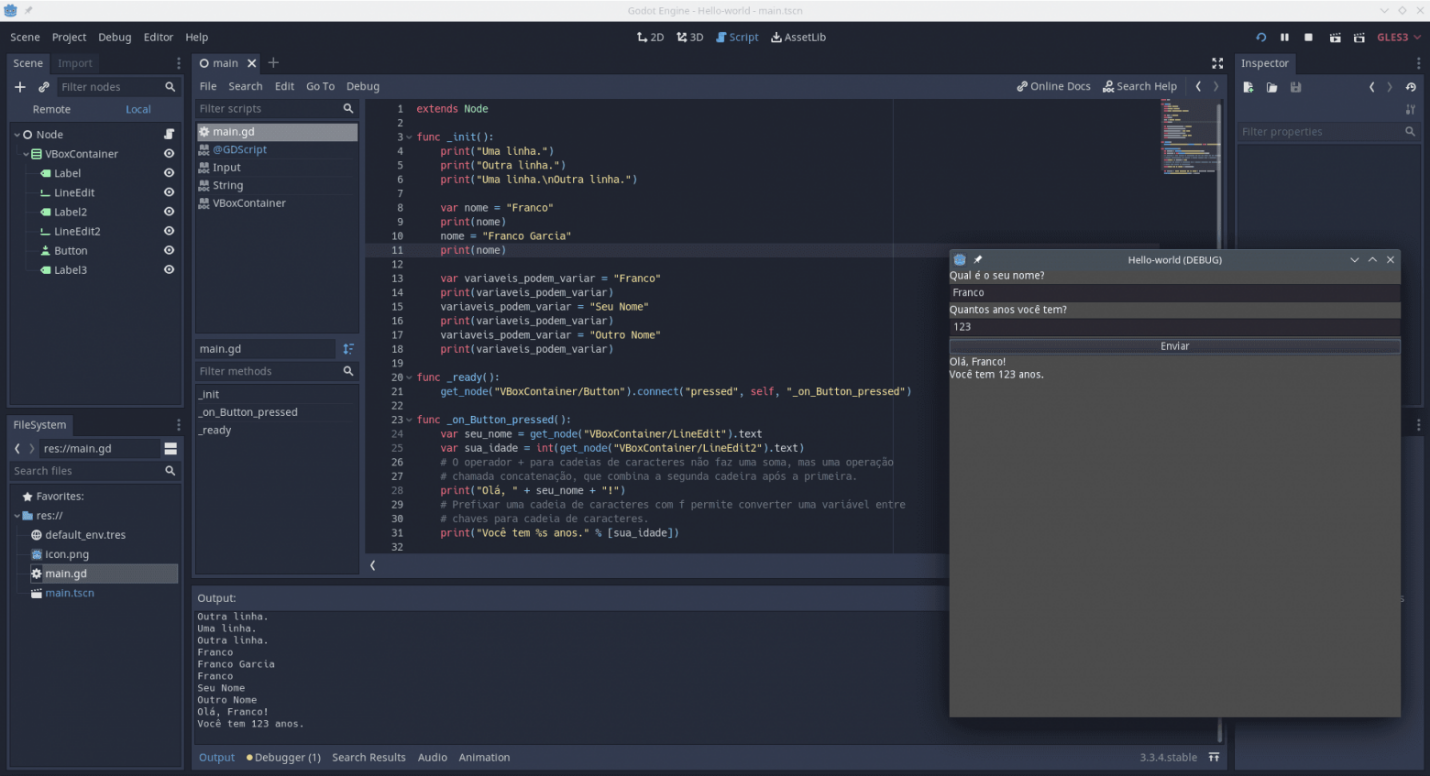
**Минусы GameMaker Studio:**

* Возможность создавать только 2d игры, при этом они уступают 2d играм на других движках. Есть ограниченная поддержка 3d, но она в основном не используется;
* Документация и форум плохо структуризированы;
* Программистами не воспринимается серьёзно и часто критикуем из-за архитектуры проекта и сложности поддержки таких проектов;
* Интерфейс становится неудобен на более-менее больших проектах;
* Магазин ассетов по сравнению с Unity совсем маленький;
* Под каждую платформу своя лицензия, что очень неудобно [3,4].

## **4. Godot**

Открытый кроссплатформенный бесплатный 2D и 3D игровой движок под лицензией MIT, который разрабатывается сообществом Godot Engine Community. Появился в 2014 году, стабильная версия 2.0 вышла в 2016. В 2018 разработчики добавили поддержку 3d и движок получил название версии 3.0.

Среда разработки запускается в Linux, BSD, macOS и Windows. Godot поддерживает GDScript (свой собственный язык), C++ и C#. Плюс есть еще средство для создания привязок к другим языкам — GDNative. Сообщество поддерживает Rust, Nim, JavaScript, Haskell, Clojure, Swift и D [5].



*Рисунок 4. Окно программы «Godot»*

Один из самых современных и, наверное, самый нашумевший движок. Часто можно увидеть статьи или ролики с названием типа "27 причин почему я выбрал Godot вмеcто Unity". Название отсылает к абсурдистской пьесе Сэмюэля Бэккета "В ожидании Годо".

Мультиплатформенный, имеет большие возможности в 2d и несколько ограниченные в 3d. Ожидается поддержка Vulkan API, что позволит использовать фотореалистичные текстуры и сильно улучшит качества 3d движка. Имеет много контрибьютеров, самое большое количество среди опенсорсных движков. Но также имеет и недостатки, местами сыроват и имеет проблемы с производительностью если на экране находится большое количество объектов. Что обещают пофиксить в грядущей версии 3.2 [4].

**Плюсы Godot:**

* Хорошая документация, большая часть переведена на русский;
* Бесплатен;
* Достаточно большое и активное сообщество. Уступает, наверное, только Unity и Unreal;
* Опенсорсный, достаточно быстро фиксятся баги;
* Работает на всех платформах;
* Быстрый, игры мало весят. По некотором данным выигрывает в 2d у Unity до 30%;
* Хорошо организована структура файлов внутри движка;
* Поддержка нескольких языков: GDScript, C#, C++. C# - пока не полная.

**Минусы Godot:**

* Сыроват. Некоторые вещи в больших движках из коробки, здесь что-то надо искать, доустанавливать или даже самим доделывать;
* Проседания по производительности с большим количеством объектов;

По сравнению с большими движками мало туториалов, баз знаний и возможности решить проблему. А из-за смены версий, те что есть часто устаревают;

* Есть неплохие игры, но хитов пока нет [3,4].

# III. Заключение

В итоге, сравнивать одну технологию с другой – очень субъективно. Поэтому выделить лучший движок для создания игры невозможно. Каждый популярный движок - это инструмент с большим комьюнити и серьёзным функционалом. Выбирать нужно по конкретным критериям проекта.

Рассмотрим аудиторию, которой могут быть интересны игры, на следующие сегменты:

**1)** Просто геймеры, которые не умеют программировать, но хотят создавать игры. Это могут быть художники, дизайнеры или совсем молодые игроки.

**2)** Те, кто с кодом знаком, в той или иной степени. Понимают конструкции языка. Но кодирование не является их основной деятельностью. Архитектура, паттерны проектирования, производительность - это всё не про них. Геймдизайнеры, менеджеры, джуны-разработчики.

**3)** Разработчики хорошего уровня, например, работающие не в геймдеве или которые хотят сменить технологию.

**4)** Малый бизнес. Кто-то кто не будет заниматься разработкой напрямую, а собирает свою инди-команду.

**5)** Те, кто ищет работу в офисе в геймдеве [4].

Для самой первой категории движки Unity, Unreal Engine 4 слишком сложны и масштабны, Godot подойдёт не всем - есть визуальный скриптинг, есть достаточно дружелюбный интерфейс и сам движок полностью русифицирован, но новичку здесь запуститься будет всё-таки непросто, а вот GameMaker Studio подойдёт хорошо.

Те, кто имеют некоторый опыт и навыки программирования могут попробовать работать в любом движке, много туториалов помогут в освоении. Но Unreal Engine 4 рекомендуется для более опытных программистов, хотя есть примеры инди разработчиков, в одиночку делающих игру на Blueprints.

Для третьей категории, хорошо знающих С++, хорошо подойдут Unity (хотя в этой категории обитают как раз наибольшее количество его хейтеров), Unreal Engine 4 и Godot (в основном самые ярые фанаты, но есть и которые скептически настроены).

Для инди команд в малом бизнесе существует много рисков при создании игры. А Unity в данном случае наименее рискованный вариант (так как есть много разработчиков, которых можно сменить, и много гайдов и ассетов для старта). Для Unreal Engine 4 людей на проект подобрать также можно, хотя и сложнее. А вот в GameMaker Studio и Godot пока рисков явно побольше, чем с Unity и Unreal Engine 4, лучше от них отказаться.

Для офисной работы в геймдеве у нас практически кроме Unity и Unreal других движков не существует. И все процессы заточены под них. Они популярны не только у программистов и гейм-дизайнеров, но и у 3d-художников и левел-дизайнеров.

В заключении, движков существует много, но я рассмотрел самые популярные и обязательно опенсорсные. Я не стал затрагивать закрытые игровые движки, такие как RAGE. И не вижу смысла их рассматривать в рамках данного доклада, так как обычному пользователю/программисту, не работающему в нужной компании, они недоступны.

# Список литературы

1. Люк Эхерн. Самоучитель по созданию компьютерных игр. - ДМК-Пресс, 2008. – 304 с.
2. Джереми Бонд. Unity и C#. Геймдев от идеи до реализации. - СПб.: Питер, 2019. – 928 с.
3. Путеводитель по геймдеву. Не Unity едины. Большой обзор игровых движков для начинающих и профи: [Электронный ресурс] // DTF. URL: https://dtf.ru/indie/966434-putevoditel-po-geymdevu-ne-unity-ediny-bolshoy-obzor-igrovyh-dvizhkov-dlya-nachinayushchih-i-profi. (Дата обращения: 09.10.2022)
4. Сравнение игровых движков: [Электронный ресурс] // Sc2tv Reborn и что тут вообще происходит (FAQ). URL: https://sc2tv.ru/blogs/falcon/2019/07/23/sravnenie-igrovykh-dvizhkov. (Дата обращения: 15.10.2022)
5. Самые популярные бесплатные движки для разработки игр: [Электронный ресурс] // Хабр. URL: https://habr.com/ru/company/timeweb/blog/659891. (Дата обращения: 16.10.2022)
6. Как разобраться в игровых движках: [Электронный ресурс] // DTF. URL: https://dtf.ru/s/superjam/236542-kak-razobratsya-v-igrovyh-dvizhkah. (Дата обращения: 20.10.2022)