Колледж Автономной некоммерческой образовательной организации высшего образования

"Научно-технологический университет "Сириус"

**Реферат по введению в специальность**

***«Разработка видеоигр»***

Работу подготовил:

студент группы 1.9.7.3

Терещенков Дмитрий Сергеевич

Проверил:

преподаватель введения в специальность:

Тенигин Альбер Андреевич

Сочи, 2022

**Оглавление**

[Введение. 3](#_Toc122308445)

[1. Разработка видеоигр в наши дни. 4](#_Toc122308446)

[2. Профессии в геймдеве. 5](#_Toc122308447)

[3. Как начать? 7](#_Toc122308448)

[4. Так как все же создать игру? 8](#_Toc122308449)

[Путь инди разработчика 8](#_Toc122308450)

[5. Путь работника студии. 9](#_Toc122308451)

[6. Движки. 10](#_Toc122308452)

[8. Начало разработки. 12](#_Toc122308453)

[9. Альфа тест и альфа версия игры. 13](#_Toc122308454)

[10. Маркетинг. 14](#_Toc122308455)

[11. Бета версия и Бета тест. 15](#_Toc122308456)

[12. Релиз. 16](#_Toc122308457)

[13. Поддержка и общение с комьюнити. 17](#_Toc122308458)

[Заключение. 18](#_Toc122308459)

[14. Список использованных источников 19](#_Toc122308460)

# Введение.

Привет. Сегодня хочу рассказать про разработку видеоигр.

В первую очередь хочу отметить, что видеоигры это одна из самых популярных и успешных отраслей в индустрии развлечений, приносящая лицам, которые в ней работают целые состояния, но так было не всегда.

Изначально, как ни странно, видео игры были лишь забавой первородных программистов, которые работали еще на мейнфреймах, а это говорит о том, что это было еще в «палеозойские времена». Тогда это было лишь что-то из разряда – «Эй! Ребята сморите что я сделал». И это не похоже на то, что мы имеем сейчас. Поэтому предлагаю окунуться в более близкие, но все еще далекие 1980-е годы, это те времена, когда индустрия видео игр уже успела стать весьма популярной, но не успела разрастись, это времена перехода от аркадных автоматов к консолям типа Nes и Sega megadrive, но это все еще малость похоже на то, что представляет из себя индустрия видеоигр сейчас. Тогда в работе участвовали в основном только программисты, ведь только они могли взаимодействовать с железом или же другим словом - кодить, но кодили они не как сейчас, в удобной обстановке, в красивом фреймворке и с выводом игрового окружения во вьюпорте, а на ассемблере (очень низкоуровневый язык программирования), что занимало гораздо больше времени, чем теперь. Грубо говоря, у программистов того времени были вечные траблы с железом и отсутствие какой-либо среды разработки, ведь разработка под устройства того времени требовала более кропотливого подхода, потому что устройства тех лет были оборудованы слабым железом и приходилось идти на всякие ухищрения. Помимо этого, раньше не существовало такого понятия как патч первого дня, из-за чего все игры приходилось всячески проверять и далеко не одну сотню раз на наличие разнообразных багов и гличей, ведь раньше выпустить игру с кривым исполнением для студии было равно, что навсегда уйти в закат, потому что такой студии уже никто не поручит делать еще одну игру.

# Разработка видеоигр в наши дни.

Возвращаясь в наше время, не могу не отметить разность того, как делали игры тогда и сейчас вновь, но, обо всем по порядку.

В первую очередь, в геймдеве появилось гораздо больше специальностей. Разработчик – это теперь не только программист, теперь разработчиком можно назвать художника, сценариста, дизайнера – в общем любого человека кто причастен к разработке. А также, у разработчиков появились фреймворки, из-за чего общая скорость выполнения работы значительно увеличилась. Но поговорим подробнее про профессии в геймдеве.

# Профессии в геймдеве.

Геймдизайнер, словно архитектор игрового мира, придумывает и детально описывает концепцию, формулируя правила игры: сеттинг (место действия, эпоха), для какой целевой аудитории она предназначена, какие в ней будут механики, какой игровой опыт он хочет передать. Он генерирует идеи, ставит задачи и сопровождает их даже после окончания разработки. Он тесно взаимодействует с командой, поэтому хорошие коммуникативные навыки – несомненное преимущество для геймдизайнера. Помимо этого, для геймдизайнера важно иметь хотя бы базовое знание во всех областях геймдева: разработка, арт, тестирование, бизнес-стратегия и т.д.

Разработчик

Как бы ни был талантлив геймдизайнер в команде, его идеи должен кто-то реализовать, поэтому разработка игр не сможет обойтись без хороших программистов. Разработчик в геймдеве занимается реализацией того, что придумал геймдизайнер. Его задачи — проектировать архитектуру игры, имплементировать игровые фичи, писать чистый код и оптимизировать игру. В зависимости от специфики игры, разработчик также может заниматься сетевой инфраструктурой, созданием искусственного интеллекта или же создавать рабочие инструменты для своих коллег, чтобы им было проще выполнять те или иные рутинные задачи.

Художник

Художники и арт-директоры прорабатывают визуальную составляющую игры. В этом направлении множество ответвлений:

**3D-моделирование.** Специалисты в этом направлении занимаются созданием 3D ассетов для игры — все, что игрок в дальнейшем будет видеть в готовом продукте в виде трехмерных объектов. Для того чтобы научиться 3D моделированию, нужно иметь чувство перспективы, пространства и объема. А общее знание анатомии пригодится всем художникам по персонажам.

**Концепт-арт.** Эти специалисты рисуют или моделируют сразу несколько вариантов одного и того же объекта, персонажа или локации, чтобы общими усилиями арт-отдела выбрать лучший из них для передачи в дальнейшую доработку. Чтобы стать концепт-артистом, нужно уметь рисовать в 2D или 3D и обладать теми же навыками, которыми должен обладать художник.

**Аниматоры.** Эти люди занимаются созданием костной, процедурной и инверсивной анимации для персонажей и объектов, с которыми игрок сможет взаимодействовать в готовом проекте. Чтобы стать аниматором, нужно хорошо разбираться в анатомии, ведь именно от реалистичности движения зависит, поверит ли игрок в происходящее в игровом мире.

**UX/UI-дизайнеры.** Интерфейс — важная часть составляющей любого продукта. Через него игрок учится взаимодействовать с игрой, через интерфейс игрок совершает множество повторяющихся операций. Грамотно построенный интерфейс делает это взаимодействие интуитивно понятным и простым.

# Как начать?

Начнем с того, что становясь разработчиком видеоигр у вас есть два направления, вы можете быть indie разработчиком (независимый разработчик), либо работать в студии. Оба этих варианта имеют как свои плюсы, так и минусы. Начнем с инди разработки, становясь инди разработчиком вы можете работать как соло, так и сформировать команду из своих знакомых единомышленников, либо оплатить чью-то работу (нанять художника, например). Основное отличие инди разработчика от того, который работает в студии это то, что у инди свой собственный график, и у него нет начальства над головой, которое нередко может быть не лучшим. Но это также значит, что инди разработчик должен делать все сам, начиная с защиты права собственности, заканчивая продвижением своего проекта. Но почему большое количество разработчиков выбирают именно инди разработку? Все связанно с тем, что программист зачастую человек асоциальный, и привык к домашнему уюту и личному пространству, а это сложно организовать в офисе, хотя и возможно. Говоря про работу в студии, сразу из плюсов бросается то, что вы стопроцентно будете получать стабильную заработную плату, в то время как у инди разработчика заработок зачастую зависит лишь от удачи. а из минусов это то, что у разработчика из студии нет возможности показать свой истинный творческий потенциал, он зачастую выполняет те проекты, которые интересны студии, в которой он трудится.

# Так как все же создать игру?

# Путь инди разработчика

В начале рассмотрим вариант с инди разработчиком: понятное дело все начинается с желания, если у вас не будет желания зачем этим вообще заниматься? Далее идет идея, вы должны сразу продумать о чем будет ваша игра, будь то очередной клон subway serfers или же оригинальный аналоговый хоррор. Затем, вам необходимо прописать концепт, это общая информация о мире, в котором происходят события, о персонажах и их характерах, об общем лоре и конечно же жанре. Составив концепт сразу, вам будет легче ориентироваться в последствии. Также вам может понадобится знание языка программирования, но все же это необязательно (об это позже). Ну и конечно же терпение, вы должны понимать, что придя в индустрию недавно, новый “майнкрафт” вы не создадите и, вероятно, первое время будете вовсе непопулярны, не говоря уже о заработке. Но заработать вам все же возможно, если вы будете монетизировать свои проекты через внутри игровые покупки, либо распространяя свои проекты на сайтах типа патреон или же itch.io (не реклама) , либо загрузив свой проект в Steam.

# Путь работника студии.

Также как и у инди разработчика, в студии все начинается с идеи, она может прийти от кого угодно – геймдизайнера, художника, программиста или даже от уборщицы. Главное что бы эту идею одобрил издатель, и это следующий шаг – одобрение издателя. Издатель - это то предприятие, которое выпускает ваш продукт от своего имени, при этом помогает с продвижением и многим прочим, но главный плюс того чтобы выпускать игры через издателя это доступ к ДевКитам (под девкитами подразумевается возможность выхода игры не только на Пк через Steam, но и на другие платформы, по типу Nintendo Switch, Play Station и Xbox) ДевКиты - это оборудование, предназначенное для разработчиков. Они чаще всего внешне отличаются от оригинальной консоли, но внутри представляют собой ту же самую консоль, только созданную для разработки. Т.е. если у вас есть обычный Nintendo Switch, то на нём вы свои игры из Юнити не запустите, вам нужен ДевКит свича. То же самое с PS, то же самое с xbox. У хорошего издателя эти ДевКиты есть, и если вы захотите свою игру портировать - они вам понадобятся. Помимо этого стоит отметить, что хороший издатель всегда поможет с локализацией, маркетингом, инвестициями. После разрешения издателя вы должны прописать документ, в котором будет описана вся основная информация о проекте – типа жанра, сеттинга, аудитории на которую нацелен продукт и примерный бюджет. После работы с бумагами в игру вступают концепт художник и режиссер, они должны дать более четкое понимание о будущем продукте, подробнее прописать сюжет и представить четкие образы, чтобы в будущем команда смогла четко ориентироваться. После этих двух промежуточных этапов приходит тяжелое время выбора движка. Но для начала что это такое.

# Движки.

Движок – это программа, которая облегчает процесс разработки, движок в общем–то отвечает за весь “движ” – верное отображение моделей в пространстве, физику и поведение объектов, логику поведения ИИ и многое другое. При выборе движка важно смотреть на его функционал и на те требования, которые нужны для создания вашего проекта. Движки бывают разные, начиная с обще-направленных, на которых можно сделать почти все что угодно, но возможно это не всегда будет удобно, так и те, которые имеют одну направляющую, но бывает и так, что студии делают свои собственные движки. И вот примеры достаточно популярных движков.

Unreal Engine — игровой движок, разрабатываемый и поддерживаемый компанией Epic Games. Первой игрой на этом движке был шутер от первого лица Unreal, выпущенный в 1998 году. Хотя движок первоначально был предназначен для разработки шутеров от первого лица, его последующие версии успешно применялись в играх самых различных жанров, в том числе стелс-играх, файтингах и массовых многопользовательских ролевых онлайн-играх. В прошлом движок распространялся на условиях оплаты ежемесячной подписки; с 2015 года Unreal Engine бесплатен, но разработчики использующие его приложения обязаны перечислять 5 % роялти от общемирового дохода с некоторыми условиями

Unity (unity в переводе с англ. — «единство», произносится как «ю́нити») — межплатформенная среда разработки компьютерных игр, разработанная американской компанией Unity Technologies. Unity позволяет создавать приложения, работающие на более чем 25 различных платформах, включающих персональные компьютеры, игровые консоли, мобильные устройства, интернет-приложения и другие. Выпуск Unity состоялся в 2005 году и с того времени идёт постоянное развитие.

Основными преимуществами Unity являются наличие визуальной среды разработки, межплатформенной поддержки и модульной системы компонентов. К недостаткам относят появление сложностей при работе с многокомпонентными схемами и затруднения при подключении внешних библиотек.

На Unity написаны тысячи игр, приложений, визуализации математических моделей, которые охватывают множество платформ и жанров. При этом Unity используется как крупными разработчиками, так и независимыми студиями.

# Начало разработки.

После выбора Движка, вся команда приступает к разработке игры. На первых стадиях разработки, делают низкополигональные модели (а в случае с 2д играми делают пробные спрайты), а также делают “Gray block[[1]](#footnote-1)” версии тестовых уровней, это нужно чтобы создать первородный скелет игры, дабы посмотреть в действии некоторые механики, и продумать будущее наполнение уровней. Зачастую так и протекают первые месяцы разработки большинства игр, первые версии любой игры выглядят примерно вот так-

Изб.1. Ранний билд “Control”

На скрине мы видим, что у большинства объектов нет даже полноценных текстур, а у юнитов используются совершенно случайные модельки, которые не попали в финальную версию игры. А теперь релизная версия того же проекта.

Изб.2. Финальная версия все того же “Control”

Здесь мы видим совершенный прогресс в графике и в геймплее. Вообще разработчики стараются не показывать ранние билды своих игр, но иногда и до обычных геймеров доходят сливы, бывает забавно сравнить разные версии игр и узнать, что добавили, а что убрали.

# Альфа тест и альфа версия игры.

Спустя какое-то время у разработчиков уже есть альфа версия игры – но для начала, что это такое? Альфа версия – это определенная стадия в разработке видеоигры, когда в ней уже присутствует геймплей и большинство механик, но при этом игра все еще далека от релиза, ведь в ней все еще много багов, гличей и в целом она все еще недоработана. В альфа тесте зачастую участвуют только штатные сотрудники и в основном это тестировщики. К альфа версиям не подпускают журналистов и блогеров, хотя и есть пара исключений, но если они и происходят, то лица, которые участвуют в альфа тесте, не должны делиться с кем-либо информацией о проекте.

# Маркетинг.

А когда у игры уже есть готовая и наполненная часть, приходить время для маркетинга, тут в игру вступает маркетолог, именно он должен придумать и исполнить способ распространения вашей игры, это может быть что угодно, начиная с видеорекламы на любом хостинге типа You tube, заканчивая той же рекламой на билбордах. Мастодонты игровой индустрии порой платят больше за маркетинг чем за саму разработку, только чтобы о их проекте узнало как можно больше людей.

Изб.3. Реклама игры “GTA 5 от ROCKSTAR” в 2013 году прямо на здании отеля Figueora.

# Бета версия и Бета тест.

Бета весьма сильно отличается от альфа версии игры. Бета это тоже версия игры не для всех (зачастую бывает именно так, но и здесь есть исключения). Бета версия игры подразумевает, что игра еще находится в доработках, но она уже гораздо больше похожа на финальную версию. Некоторые игры находятся, можно сказать, в “вечной” бете - это те игры, в которые периодически добавляют больше контента, а припиской “бета” разработчики практически снимают с себя некоторую ответственность. Но в большинстве случаев в бету играют на таких больших выставках как E3, на подобных выставках зачастую журналисты, а иногда и любой желающий, могут поиграть в еще «сыроватую» версию игры и составить свое личное мнение, а после поделится им с кем-нибудь. А еще есть вариант разослать ключи на игру большому количеству блогеров. Это тоже делается с целью как можно больше охватить медиа пространство, плюс это самый легкий способ вызвать у людей желание купить игру, ведь смотря как в ту или иную игру играет блогер, у зрителя, с большой вероятностью, возникнет желание тоже поиграть. Но кому угодно ключи засылать не станут. Обычно ох отсылают тем блогерам, у которых уже есть большая фан база. Такие выставки как E3 тоже можно отнести к рекламе, ведь они популярны, и именно из них вы можете заинтересовать еще больше потенциальных покупателей вашего продукта.

# Релиз.

А теперь, о самом кульминационном этапе для всех причастных к разработке – выпуск проекта в свет. Здесь потеют все! И вот почему, проводя параллель с играми прошлого , они были весьма примитивны, и в них было сложнее допустить ошибку и вызвать очередной баг, который придется исправить, а ведь исправляли, но тут роль играет интернет, а вернее его распространение со временем. Раньше интернет не был распространен повсеместно, и именно поэтому игры делали так долго. В основном время тратили на тесты, ведь у разработчиков не было такой прекрасной возможности как патч первого дня, о котором речь пойдет далее. Патч первого дня - относительно новое выражение, подразумевающее, что при выходе игры, она может оказаться весьма сломанной, ведь пускай разработчики и тестировали большую часть аспектов, но этого все равно недостаточно, потому что игроки это люди, а люди имеют огромнейший фактор случайности, плюс игроков очень много и шансы найти новые ошибки возрастают в феноменальной прогрессии. Именно здесь и вступает в игру патч первого дня, он позволяет разработчика получить фидбек от игроков, и исправить все ошибки и баги в первый же день (но опять же, есть проекты, которым даже патч первого месяца не помог, настолько они были сломаны.)

# Поддержка и общение с комьюнити.

После релиза, вам не следует «забивать» на свой проект, ведь если вы собираетесь выпустить продолжение вашей игры или же вы являетесь создателем онлайн игры, вы должны прислушиваться к своему комьюнити. К примеру, предположим, фанат вашей игры не всегда хотел бы увидеть в вашем творении инопланетное оружие, при том, что ваша игра является реалистичным симулятором фермерской жизни. Вам, вероятно, следует подумать - не бредовая ли это идея? Но вот если уже большая часть фан сообщества вашей игры требует чтобы вы добавили инопланетное оружие в ваш симулятор про бедного фермера, вероятно, вам следует это сделать. Ведь тогда вы сможете показать своему комьюнити, что вы готовы выслушивать и добавлять их идеи.

На этом финальном этапе вы должны до последнего поддерживать жизнь в своем продукте, пока считаете, что это сможет принести вам прибыль, а не убыток.

# Заключение.

В заключении хочу отметить, что индустрия видеоигр популярна и будет популярна еще долгое время, ведь она имеет большой потенциал. Ведь с приходом будущего, весьма вероятно изобретение новых аналоговых устройств, которые позволят експириенсу от видеоигр перейти совершенно на новый уровень. Следовательно, игровая индустрия будет только развиваться и пополняться новыми идеями, возможностями и, следовательно, рабочими местами. И аналогия с прошлым только подтверждает этот прогресс развития.

# Список использованных источников

1. [Ссылка](https://habr.com/ru/company/otus/blog/586280/) на статью про специальности в геймдеве.
2. [Ссылка](https://cq.ru/news/industry/posle-sliva-gta-6-razrabotchiki-pokazali-rannie-versii-svoikh-igr-v-podderzhku-rockstar) на статью с первым изображением (не советую переходить, на этом сайте много странной рекламы.).
3. [Ссылка](https://store.epicgames.com/ru/p/control) на страничку игры в epic games (изображение было взято оттуда.).
4. [Ссылка](https://inlnk.ru/JjgQV9) на статью с третьим изображением.
5. Книга из которой я подчеркнул для себя большое количество информации - Unity и C#. Геймдев от идеи до реализации. (Джереми Гибсон Бонд.)

1. Gray Block – уровень из примитивов (кубы, цилиндры, сферы и т.д.) зачастую в движке уже встроен инструмент для создания gray block версии уровня. [↑](#footnote-ref-1)