

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ «РОССИЙСКИЙ
ГОСУДАРСТВЕННЫЙ ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ им. А. И.
ГЕРЦЕНА»**

Институт информационных технологий и технологического образования

Кафедра информационных технологий и электронного обучения

КУРСОВАЯ РАБОТА

УПРАВЛЕНИЕ РАЗРАБОТКОЙ ПРОГРАММНОГО ПРОЕКТА НА UNITY

Направление подготовки: «Информатика и вычислительная техника»

Руководитель:

Кандидат физ. - мат. наук,
ассистент, доцент

_____ Н. Н. Жуков

«__» _____ 2021 г.

Автор работы:

Студент группы Зоб-ИВТ-2/19

_____ М. Д. Глебов

«__» _____ 2021 г.

Санкт-Петербург

2021

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	3
ГЛАВА 1 «МЕТОДОЛОГИЯ РАЗРАБОТКИ ПРОЕКТА»	4
ГЛАВА 2 «КРУПНЫЕ ПРОЕКТЫ НА UNITY»	6
ГЛАВА 3 «АССЕТЫ»	8
3.1 UNITY ASSET STORE	9
ГЛАВА 4 «СКРИПТИНГ»	9
4.1 КЛАСС MONOBEBHAVIOUR	12
ГЛАВА 5 «СБОРКА ПРОЕКТА»	14
ЗАКЛЮЧЕНИЕ	16
ЛИТЕРАТУРА	18

ВВЕДЕНИЕ

Unity — межплатформенная среда разработки компьютерных игр, разработанная американской компанией Unity Technologies.

Кроссплатформенность является одним из наиболее важных факторов для разработчиков, которые решают разрабатывать свои проекты именно на Unity. Кроссплатформенность позволяет охватывать максимально большую аудиторию, включая мобильные устройства, персональные компьютеры на Windows, Linux и MacOS, а также игровые консоли.

Unity используется не только в области разработки компьютерных игр, но также и в создании графики для кино, строительстве, транспортной логистике и построении математических моделей.

Unity является игровым движком (game engine). Игровой движок - это программное обеспечение, предоставляющее набор необходимых функций для быстрого и эффективного создания компьютерных игр. Игровой движок - это среда для разработки игр, которая поддерживает и сводит вместе ряд ключевых аспектов. Вы можете импортировать художественные материалы и игровые ассеты, будь то 2D или 3D, из другого программного обеспечения, такого как Maya, 3ds Max или Photoshop. Также в игровом движке можно добавлять освещение, аудио, спецэффекты, физику, анимацию, интерактивность и игровую логику. И в завершение всего можно редактировать, отлаживать и оптимизировать свой контент для выбранных вами игровых платформ.

По большей части игровые движки могут предоставлять вам архитектуру с высокопроизводительным рендерингом и с доступом к быстрому графическому API, что даёт вам возможность добиваться великолепной визуальной достоверности в играх. Рендеринг или по-другому "отрисовка" - это процесс получения изображения по модели с

помощью компьютерной программы. Часто в компьютерной графике, художественной или технической, под рендерингом понимается знание плоской картинке, цифрового растрового изображения по разработанной 3D сцене. Синонимом в данном контексте является визуализация. Визуализация - один из наиболее важных разделов компьютерной графики и на практике он тесным образом связан с остальными. Обычно программные пакеты трёхмерного моделирования и анимации включают в себя также и функцию рендеринга. Существуют отдельные программные продукты, выполняющие рендеринг. В зависимости от целей различают: пре-рендеринг как достаточно медленный процесс визуализации, применяющийся в основном при создании видео и рендеринг в режиме реального времени, например в компьютерных играх.

Главной задачей при разработке программного проекта на Unity является организация взаимодействия между разработчиками. К примеру: Дизайнеры уровней должны активно коммуницировать с художниками и программистами игровых механик. Так как Unity в основном применяется в сфере малых компаний, идеальным решением будет максимально вовлечь каждого сотрудника во все аспекты разработки.

ГЛАВА 1 «МЕТОДОЛОГИЯ РАЗРАБОТКИ ПРОЕКТА»

Анализируя подходы к разработке программных продуктов, обеспечивающих максимальную эффективность в условиях небольшой компании, выбор остановился на методологии экстремального программирования (XP). Данная методология берет за основу подход agile, а ее разработчиками являются Кент Бек, Уорд Каннингем и Мартин Фаулер.

Экстремальное программирование представляет собой изменение всех наиболее традиционных подходов в более динамическую форму

работы. Например, если при обычном подходе просмотр написанного кода происходит последовательно, то при экстремальном программировании написание и ревизия происходят параллельно друг другу. Заказчик в данной методологии рассматривается как конечный пользователь.

Важной особенностью данного подхода является парное программирование. Парное программирование предполагает одновременную работу двух человек за одним компьютером, один из которых непосредственно занимается написанием кода, а второй наблюдает и участвует в обсуждении. Внутри коллектива пары постоянно меняются, тем самым обеспечивая более углубленное понимание каждого сотрудника во всех областях разработки и усиливает уровень взаимодействия внутри команды.

Не менее важным является «Игра в планирование». Под этим подразумевается работа с быстро составленным планом разработки, в который, по мере прогресса, обновляется до более четкого. Артефактами игры в планирование является набор бумажных карточек, на которых записаны пожелания заказчика (customer stories), и приблизительный план работы по выпуску следующих одной или нескольких небольших версий продукта. Критическим фактором, благодаря которому такой стиль планирования оказывается эффективным, является то, что в данном случае заказчик отвечает за принятие бизнес-решений, а команда разработчиков отвечает за принятие технических решений.

Подход к тестированию подразумевает уделение особого внимания юнит-тестам и функциональному тестированию. Тесты модулей (юнит-тесты) позволяют разработчикам убедиться в том, что каждый из них по отдельности работает корректно. Они также помогают другим разработчикам понять, зачем нужен тот или иной фрагмент кода, и как он функционирует — в ходе изучения кода тестов логика работы тестируемого кода становится понятной, так как видно, как он должен использоваться.

Функциональные тесты покрывают гораздо больше — то есть, у тестов, которые при своём выполнении затрагивают больший объём кода, шанс обнаружить какое-либо некорректное поведение, очевидно, больше. По этой причине в промышленном программировании написание функциональных тестов нередко имеет больший приоритет, чем написание юнит-тестов.



Рис.1 Этапы разработки игры

ГЛАВА 2 «КРУПНЫЕ ПРОЕКТЫ НА UNITY»

Если говорить об опыте разработки крупных проектов не стоит забывать о компьютерной графике. Движок способен создавать фотореалистичное изображение, соответствующее современным стандартам кинематографа. В этой сфере Unity часто используется студиями визуальных эффектов, такими как Weta Digital, которая была выкуплена Unity Software. Weta Digital учувствовала в создании компьютерной графики для таких проектов как:

- Властелин Колец
- Аватар
- Игра престолов
- И многие др.

На сегодняшний день студия активно сотрудничает с Disney и принимает участие почти в каждом их проекте.

На рынке компьютерных игр Unity больше закрепился в области инди-игр. Так как большинство крупных студий предпочитают использовать собственные разработанные компьютерные движки, избегая достаточно высокую комиссию за использование движка Unity. Главным конкурентом в крупных проектах является Unreal Engine от компании Epic Games, который в последнее время стал сильно вытеснять Unity из рынка крупнобюджетных игр, предоставляя новые технологии в работе со светом и большим количеством детализированных объектов.

На рынке инди разработчиков Unity занимает лидирующие позиции поглощая более 50% рынка, в том числе и на мобильных устройствах.

(рис.1)

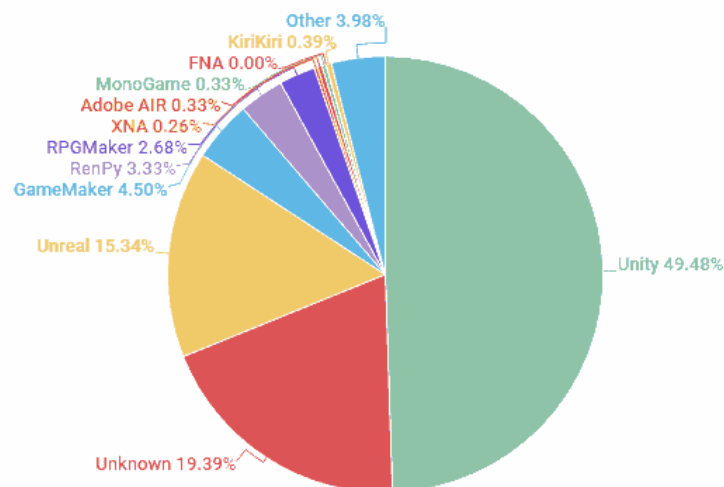


Рис.2 Рынок игровых движков

Несмотря на высокий уровень конкуренции Unity смог добиться результата, расширив рынок с 13% в 2013 году до 49.5% в 2021. За это время на движке было большое количество продуктов, получивших популярность среди пользователей всех платформ, такие как:

- Hearthstone (Лучшая мобильная игра 2014/2015, лидер среди жанра карточных игр)
- Pokémon Go (Установила множество рекордов среди всех мобильных приложений: Самая скачиваемая мобильная игра в первый месяц выхода; Самый большой доход, который принесла мобильная игра в первый месяц выхода; Самая “кассовая” игра в 55 странах мира в первый месяц выхода)
- Temple run (Крайне популярная игра на мобильные платформы, выпущенная в 2011 году.)
- Cuphead и Hollow Knight (Оба проекта создавались на Unity небольшими независимыми командами. Игры продались тиражом 5 и 1 млн копий соответственно.)
- Beat Saber (Один из самых популярных VR проектов. Продано более 2млн копий и 10млн песен в качестве дополнений)

ГЛАВА 3 «АССЕТЫ»

Любой комплексный объект состоит из большого количества более простых объектов. В случае с компьютерными играми, они все состоят из ассетов. Игровой ассет - это ресурс, который может выступать в виде объекта. Он является строительным блоком игровых объектов, к

ассетам относятся все данные, используемые компьютерной игрой, такие как: геометрические модели, текстуры, отдельные звуки, анимационные данные и пр. Собирать эти ассеты можно в сцену и окружающую среду. Игра была бы не полной без какого-либо звука, будь то музыкальный фон или звуковые эффекты. Аудиосистема Unity - гибкая и мощная, она может импортировать большинство стандартных аудиоформатов и имеет сложные функции для воспроизведения звуков в 3D пространстве с опциональными эффектами, такими как применение эха и фильтрации.

3.1 UNITY ASSET STORE

Unity Asset Store – это очень большая торговая площадка с множеством ассетов, разработанными другими пользователями. Пользователь может не только найти подходящие для своего продукта ассеты, но и разместить собственные. В магазине ассетов присутствует множество бесплатных ассетов. Все они разделяются по категориям: Аудио, Визуальные эффекты, Модели, Инструменты и многое другое. Так же при поиске необходимого ассета можно найти его под конкретную версию движка Unity, так как ассеты, разработанные под более новые версии, редко будут иметь возможность использоваться в предыдущих.

ГЛАВА 4 «СКРИПТИНГ»

Даже если у вас нет навыка программирования, вы всё же можете достичь многого в Unity. С магазином Unity Asset Store и новыми художественными и дизайнерскими возможностями, интегрированными в редактор, вы располагаете мощными решениями, помогающими вам

быстрее достичь своей цели при создании проектов. Большинство интерактивного контента, создаваемого в Unity, опирается на программирование с текстовым интерфейсом. В Unity поддерживается язык программирования C#.

Скриптинг говорит объектам GameObjects как себя вести. Игровой процесс создают скрипты и компоненты, прикрепленные к GameObjects, а также их взаимодействие друг с другом. Скриптинг в Unity отличается от чистого программирования.

Движок Unity работает большими циклами, он считывает все данные в игровой сцене. Например, он считывает данные об освещении в сетках, поведении, и обрабатывает все эти данные за вас. Вы отдаёте Unity прямые команды, записанные в ваших скриптах. Unity выполняет их кадр за кадром, насколько быстро он может. Достижение высокой скорости смены кадров означает не только то, что ваша игра будет выглядеть красивее, но и скрипты будут выполняться чаще, что делает управление более отзывчивым.

При создании скрипта вы по создаёте новый тип компонента, который можно прикрепить к игровым объектам, как и любой другой компонент. Подобно тому как другие компоненты имеют свойства, которые можно редактировать в инспекторе, вы также можете разрешить редактирование значений в вашем скрипте из инспектора.

Для редактирования значений в окне инспектора переменная должна обладать модификатором доступа Public. Если вы поставите переменную модификатора доступа Private, то в инспекторе переменная не отобразится и её нельзя будет там изменять. Также Unity позволяет изменять значение переменных скрипта в запущенной игре. Это очень полезно, чтобы увидеть эффекты от изменений сразу же, без остановки и перезапуска. Когда проигрывание заканчивается, значение переменных сбрасывается в то

состояние, которое они имели до нажатия кнопки Play. Это гарантирует, что вы можете играть с настройками объектов, не боясь что-то испортить. В редакторе Unity вы изменяете свойства компонента, используя окно Inspector. Так, например изменение позиций компонента Transform приведёт к изменению позиции игрового объекта. Аналогично вы можете изменить цвет материала компонента Renderer или массу Rigidbody, соответствующим влиянием на отображение или поведение игрового объекта. По большей части, скрипты также изменяют свойства компонентов для управления игровыми объектами.

В случае создания игры в двухмерном пространстве, перемещения персонажа будет происходить в двух плоскостях. Так же будет добавлена возможность ходить, бежать, прыгать, приседать и красться. Для реализации был написан следующий код:

```
namespace UnityStandardAssets._2D
{
    [RequireComponent(typeof (PlatformerCharacter2D))]
    public class Platformer2DUserControl : MonoBehaviour
    {
        private PlatformerCharacter2D m_Character;
        private bool m_Jump;

        private void Awake()
        {
            m_Character = GetComponent<PlatformerCharacter2D>();
        }

        private void Update()
        {
            if (!m_Jump)
            {
                m_Jump = CrossPlatformInputManager.GetButtonDown("Jump");
            }
        }

        private void FixedUpdate()
        {

```

```

        bool crouch = Input.GetKey(KeyCode.LeftControl);
        float h = CrossPlatformInputManager.GetAxis("Horizontal");
        m_Character.Move(h, crouch, m_Jump);
        m_Jump = false;
    }
}
}

```

4.1 КЛАСС MONOBEHAVIOUR

MonoBehaviour - это базовый класс для всех новых сценариев Unity. Класс MonoBehaviour предоставляет вам список всех функций и событий, доступных для стандартных сценариев, которые прикреплены к игровым объектам. Каждый игровой объект имеет положение, вращение и масштаб в пространстве, будь то 3D или 2D. Это представлено компонентом Transform. Помимо предоставления этой информации, компонент преобразования имеет множество полезных функций, которые можно использовать для перемещения, масштабирования, поворота, повторного отображения и манипулирования объектами, а также для преобразования координат из одного пространства в другое. В Rigidbody и Rigidbody (2D) для большинства элементов игрового процесса физический движок предоставляет самый простой набор инструментов для перемещения объектов, обнаружения триггеров и столкновения, а также применения сил.

Класс MonoBehaviour обладает несколькими функциями. Некоторые из них были реализованы в коде выше, но необходимо ознакомиться с каждым из них.

Функция Start вызывается в кадре, когда скрипт активирован непосредственно перед первым вызовом любого из методов Update. Как и функция Awake, Start вызывается ровно один раз за время существования скрипта. Однако Awake вызывается при инициализации объекта сценария,

независимо от того, включен сценарий или нет. Start не может быть вызван в том же кадре, что и Awake, если сценарий не включен во время инициализации.

Функция Awake вызывается для всех объектов сцены перед вызовом функции Start любого объекта. Этот факт полезен в случаях, когда код инициализации объекта А должен полагаться на уже инициализированный объект В; Инициализация В должна выполняться в Awake, а инициализация А должна выполняться в Start.

Функция Update вызывается каждый кадр, если MonoBehaviour включен. Update - это наиболее часто используемая функция для реализации любого игрового скрипта.

Функция FixedUpdate вызывается с частотой физической системы, раз в определенный промежуток времени. По умолчанию параметр функции FixedUpdate равен 0,02 или же 50 кадров в секунду.

LateUpdate вызывается после того, как были вызваны все функции обновления. Это полезно для порядка выполнения скрипта. Например, камера слежения всегда должна быть реализована в LateUpdate, потому что она отслеживает объекты, которые могли перемещаться внутри Update.

OnGUI вызывается для рендеринга и обработки событий графического интерфейса. OnGUI - единственная функция, которая может реализовать систему графического интерфейса «Immediate Mode» (ImGui) для рендеринга и обработки событий графического интерфейса. Реализация OnGUI может вызываться несколько раз за кадр (один вызов на событие). Если для свойства enabled MonoBehaviour установлено значение false, OnGUI () не будет вызываться.

Функция OnEnable вызывается, когда объект становится доступным и активным.

Функция `OnDisable` вызывается, когда поведение становится отключенным. Это также вызывается при уничтожении объекта и может использоваться для любого кода очистки. Когда скрипты перезагружаются после завершения компиляции, будет вызываться `OnDisable`, а затем `OnEnable` после загрузки скрипта.

Перечисленные функции это не все, что можно реализовать при помощи класса `MonoBehaviour`, они являются наиболее используемыми и ими вполне можно ограничиться в рамках небольшого проекта.

ГЛАВА 5 «СБОРКА ПРОЕКТА»

Для создания уровня игры необходимо объединить ассеты в сцену. В первую очередь был разработан фон. Фоном будет являться фиксированной изображение, состоящее из двумерных ассетов неба, облаков и гор. Для этого выберем в качестве родительского объекта небо, а облака и горы будут его дочерними объектами. Таким образом получится ненавязчивое изображение, которое послужит хорошим фоном для игры. Следует добавить, что к облакам в дальнейшем возможно добавить скрипт, который будет постепенно перемещать их по игровому экрану, добавляя большей динамики.

Площадкой для перемещения персонажа будут являться платформы. Они будут состоять из прямоугольников, обладающих коллизией, с наложенными поверх спрайтами растительности. Платформа будет соскальзывать игрового персонажа с края, если он на нем остановится и препятствовать прыжкам снизу-вверх.

В качестве игрового персонажа выбран стандартный ассет робота. К нему так же имеется анимация для бега, ходьбы в присядку и прыжка.

Дочерним объектом игрового персонажа будет являться анимация и написанный скрипт на перемещение.



Рис.3 Игровой персонаж

Объект main camera – это часть игровой сцены, которую видит пользователь. Для того, чтобы корректно следовать за персонажем и отображать фон воспользуемся функциями дочернего объекта. Фон будет являться дочерним объектом main camera, а main camera, в свою очередь, будет являться дочерним объектом для игрового персонажа (Рис.4). Для main camera так же написан скрипт, который делает перемещение за персонажем более плавным, избегая проблемы с постоянным нахождением персонажа в одной и той же точке на экране и устраняя эффект укачивания игрока. Таким образом камера плавно следует за пользователем и фон не выходит за рамки игрового окна.

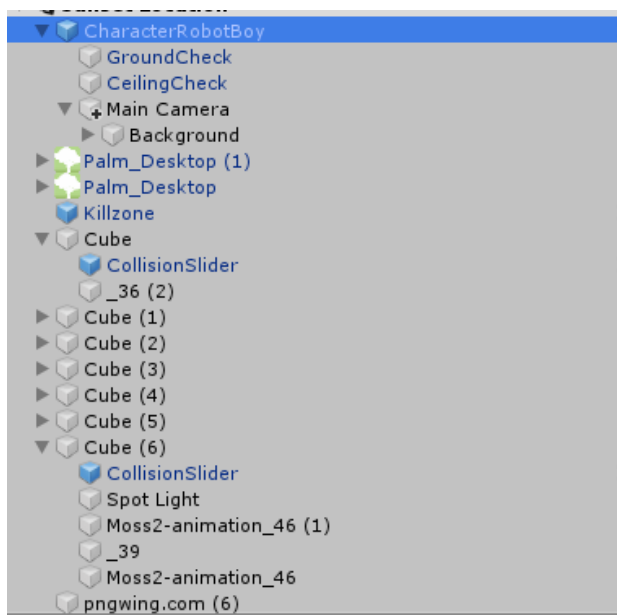


Рис.4 Окно иерархии объектов

Также необходимо сделать невидимый объект `killzone`, который будет перезапускать уровень в случае падения персонажа. Данный скрипт фиксирует состояние нахождения в своей зоне игрового персонажа, и если получает значение `True`, то перезапускает уровень.

Для создания уровня необходимо расположить ассеты платформ, добавить свет, декоративных объектов и указателей. При расположении платформ стоит учитывать максимальную высоту и дальность прыжка игрового персонажа.

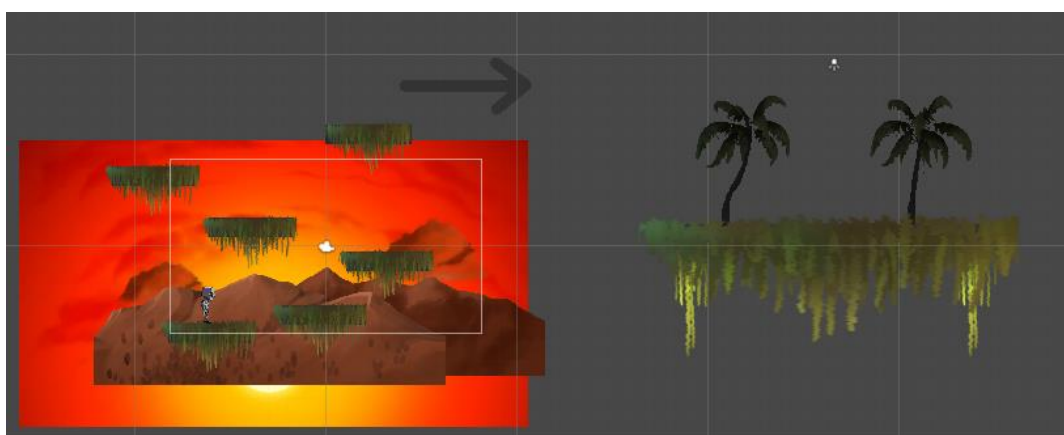


Рис.5 Итоговая игровая сцена

ЗАКЛЮЧЕНИЕ

Unity является наилучшим выбором для начала своего пути в сфере разработки компьютерных игр и графики. Данный движок позволяет углубиться в тонкости разработки, при этом давая возможности обходиться без профессиональных навыков в области программирования, предоставляя множество решений для написания скриптов.

Так же Unity содержит множество профессиональных инструментов для работы над крупнобюджетными проектами, предоставляя

разработчикам свободу в создании своих проектов. Мультиплатформенность делает Unity наиболее дружелюбным для начинающих разработчиков, предоставляя большую площадку для размещения своих проектов.

С точки зрения организации и управления проектами на Unity, движок предоставляет инструменты для сбора внутриигровой информации, которая позволит оптимизировать и улучшить игровой процесс как с точки зрения производительности, так и для бизнеса и заработка.

При работе над крупными проектами, включающими более 1000 работников, современные компании разграничивают отделы непосредственной разработки и отдел маркетинга, что нередко приводит к проблемам в коммуникации между этими отделами, так как зачастую курировать эти процессы доверяют топ-менеджерам, которые, в свою очередь, могут не знать особенностей разработки и быть нацеленными только на получения прибыли, забывая о качестве продукта. Мною хотелось бы отметить важность понимания руководства всех аспектов разработки проектов, использующих игровые движки в качестве инструмента. При принятии маркетинговых решений необходимо консультироваться с разработчиками и поддерживать высокий уровень коммуникации между всеми отделами. Это положительно скажется как на опыте разработки проекта для каждого сотрудника, так и на общем уровне качества производимой компьютерной игры или графики для кино.

ЛИТЕРАТУРА

1. Открытое образование [openedu] // Учебный курс "Основы разработки игр на Unity"
URL: [Открытое образование - Основы разработки игр на Unity \(openedu.ru\)](https://openedu.ru)
2. Свободная энциклопедия [Wikipedia] // Unity (игровой движок), 2021 URL: [Unity \(игровой движок\) — Википедия \(wikipedia.org\)](https://ru.wikipedia.org/wiki/Unity_(игровой_движок))
3. Официальный сайт Unity. URL: [Unity - Unity](https://unity.com)
4. Документация к движку Unity. URL: [Unity - Руководство: Руководство Unity \(unity3d.com\)](https://docs.unity3d.com/ru/)
5. Unity Asset Store. URL: [Unity Asset Store - The Best Assets for Game Making](https://assetstore.unity.com/)
6. Weta Digital. URL: [Filmography | Weta Digital \(wetafx.co.nz\)](https://www.wetafx.co.nz/filmography)
7. Статья «Unity покупает студию, создавшую спецэффекты для «Властелина колец» и «Аватара»» / Ведомости // URL: [Unity покупает студию, создавшую спецэффекты для «Властелина колец» и «Аватара» - Ведомости \(vedomosti.ru\)](https://www.vedomosti.ru/tech/news/2017/10/26/unity-pokupaet-studiju)
8. Статья «Приемы кино в Unity» / DTF // 26 окт 2017 URL: [Приемы кино в Unity: маркетинговые ролики для игр внутри движка \(dtf.ru\)](https://www.dtf.ru/tech/unity-marketingovye-roliki)
9. Статья «Самые успешные игры на Unity» / Skillbox // 16 апр 2020 URL: [Самые успешные игры на Unity за 10 лет / Skillbox Media](https://www.skillbox.ru/articles/samye-uspehnyie-igry-na-unity/)
- 10.Статья «7 лучших игр, написанных на Unity» / ITVDN // 04 сен 2020 URL: [7 лучших игр, написанных на Unity - Блог ITVDN](https://itvdn.ru/blog/7-luchshih-igr-na-unity/)
- 11.Статья «Unity – самый популярный игровой движок?» / habr // 10 сен 2021 URL: [Unity — самый популярный игровой движок? Обзор движков, на которых делают игры для Steam / Хабр \(habr.com\)](https://habr.com/ru/post/494444/)
- 12.Свободная энциклопедия [Wikipedia] //Beat Saber, 2021 URL: [Beat Saber — Википедия \(wikipedia.org\)](https://ru.wikipedia.org/wiki/Beat_Saber)
- 13.Свободная энциклопедия [Wikipedia] // Hearthstone, 2021 URL: [Hearthstone — Википедия \(wikipedia.org\)](https://ru.wikipedia.org/wiki/Hearthstone)

- 14.Свободная энциклопедия [Wikipedia] // Экстремальное программирование,
2021 URL: [Экстремальное программирование — Википедия \(wikipedia.org\)](https://ru.wikipedia.org/wiki/Экстремальное_программирование)
- 15.Agile coach [Atlasian] // Что такое agile? URL: [Что такое agile? | Atlassian](https://www.atlassian.com/agile)