

VR  AR



ООО СП «СОДРУЖЕСТВО»

Учебно-методические и презентационные материалы
для проведения интерактивных практических занятий

«Разработка игры в виртуальной реальности»



Оглавление

Введение.....	3
Мастер-класс по разработке игры в формате виртуальной реальности с использованием механики стрельбы из лука.....	4
Занятие №1. Среда Unity и Project View.....	5
Занятие №2. Анимация в Unity и создание игрового проекта.....	21
Анимация твердого тела.....	23
Анимация на основе скелета.....	23
Анимация спрайтами.....	23
Анимация, основанная на физике (использование физической системы Unity).....	23
Анимация частицами.....	23
Анимация персонажей с помощью системы Mecanim.....	24
Создание игрового проекта для Windows.....	25
Занятие №3. 3D-проектирования Blender в среде Unity.....	28

Введение

В цифровую эпоху у современных обучающихся множество возможностей для развития и обучения. С помощью новейших технологий сегодня можно полетать в космосе, посмотреть на настоящего динозавра и провести химические опыты как настоящий учёный.

Тенденцией последних лет является активное внедрение и использование программного обеспечения во всех сферах образования, где множество различных технологий применяется для обеспечения интерактивного обучения.

К таким интерактивным инструментам относятся и технологии виртуальной и дополненной реальности.

Технологии дополненной реальности (Augmented Reality, AR) способны проектировать цифровую информацию (изображения, видео, текст, графику) поверх экранов устройств и объединять виртуальные объекты с реальной средой.

Виртуальная реальность (Virtual Reality, VR) с помощью 360° картинки переносит человека в искусственный мир, где окружающая среда полностью изменена. Познакомиться с дополненной реальностью можно с помощью одного лишь смартфона, однако для погружения в виртуальное пространство понадобится специальный шлем или очки.

Иммерсивные технологии характеризуются следующими преимуществами:

1. **Наглядность.** В виртуальном пространстве без препятствий можно детализировано рассмотреть любой процесс или объект, что значительно интереснее, чем смотреть на картинки в учебнике. Например, через приложение Anatomyo можно изучить строение тела в мельчайших подробностях, а Operation Apex покажет все богатства подводного мира.

2. Сосредоточенность. В виртуальной среде человек не будет отвлекаться на внешние раздражители, что позволит полностью сфокусироваться на материале.
3. Максимальное вовлечение. Иммерсивные технологии предоставляют возможность полностью контролировать и изменять сценарий событий. Ученик может стать свидетелем исторических событий, собственноручно провести опыт по физике или химии или решить задачу в игровой и доступной для понимания форме.
4. Безопасность. С помощью VR- и AR-технологий можно провести сложную операцию, управлять спорткаром или даже космическим шаттлом, провести опыт с опасными химическими веществами и при этом не причинить вреда ни себе, ни окружающим
5. Результативность. В ходе проведённого исследования двум группам людей предлагалось запомнить расположение определённых изображений. Во время эксперимента одна из групп использовала шлемы виртуальной реальности, вторая — обычные компьютеры. При этом группа, которая изучала изображения с помощью VR-шлемов, показала результат на 10% выше, чем участники второй группы.

Данные иммерсивные методы обучения могут стать основным инструментом в образовании и совершить революцию в обучении как школьников, так и студентов.

Мастер-класс по разработке игры в формате виртуальной реальности с использованием механики стрельбы из лука.

Цель: первоначальное знакомство с межплатформенными средами разработки игр и приложений виртуальной реальности.

Задачи:

- овладение базовыми понятиями виртуальной и дополненной реальности;
- приобретение теоретических знаний для понимания конструктивных особенностей и принципов работы VR/AR-устройств;

- приобретение теоретических знаний для понимания конструктивных особенностей и принципов работы VR/AR-устройств;
- приобретение навыков работы в межплатформенной среде разработки игровых приложений Unity;
- формирование основных приёмов работы в программах для разработки AR/VR-приложений, 3D-моделирования;
- приобретение навыков работы с готовыми 3D-моделями, создания несложных 3D-моделей;
- приобретение умений создавать собственные AR/VR-приложения с помощью специальных программ и приложений.

Продолжительность: 80 минут.

Описание. В рамках интенсива обучающиеся познакомятся с движком для разработки игр Unity, научатся собирать сцену с игровой механикой стрельбы из лука по противникам. Для сборки проекта обучающимся будут даны готовые 3D-модели игрового окружения, разработанные в программе Blender (юрты, модели противников, живность, деревья и др.). После сборки собственного виртуального мира каждый сможет запустить игру в формате виртуальной реальности в VR-шлеме и поиграть в него. (Уфимский университет)

Занятие № 1 Среда Unity и Project View.

Тема: Unity – межплатформенная среда разработки игровых приложений.

Цель: знакомство с межплатформенной средой разработки игровых приложений Unity.

Задачи:

- приобретение навыков установки программы;
- изучение интерфейса и основных инструментов среды Unity.

Оборудование:

- компьютер (ноутбук);
- проектор;
- интерактивная доска;

- очки дополненной реальности;
- смартфон;
- веб-камера;
- МФУ.

Список источников:

1. **Ферроне Харрисон.** Изучаем C# через разработку игр на Unity. 5-е издание – СПб.: Питер, 2022. – 400 с.: ил.
2. **Джереми Гибсон Бонд.** Unity и C#. Геймдев от идеи до реализации. 2-е издание – СПб.: Питер, 2022. – 928 с.
3. **Джозеф Хокинг.** Unity в действии. Мультиплатформенная разработка на C# – СПб.: Питер, 2019. – 352 с.
4. **Программирование искусственного интеллекта в играх.** Руководство / Паласиос Хорхе – М.: ДМК Пресс, 2017. – 274 с.
5. **Доусон М. Д71.** Изучаем C++ через программирование игр – СПб.: Питер, 2016. – 352 с.: ил. Хорхе, Паласиос Unity 5.x.

Виртуальная реальность (VR) – это технология, которая позволяет создавать иллюзию полного погружения в цифровое пространство. Сегодня VR-технологии становятся все более доступными и популярными, особенно в области видеоигр. VR-игры предлагают игрокам совершенно новый уровень взаимодействия с виртуальным миром и позволяют полностью погрузиться в игровой процесс.

VR-игры не только доставляют удовольствие игрокам, но и предлагают множество преимуществ для разработчиков:

- VR-технологии позволяют создавать более реалистичные игровые миры;
- виртуальная реальность дает возможность создавать детализированные 3D-модели и более точную физику, что позволяет создавать более интересные и реалистичные игры;
- VR-технологии находятся на стадии активного развития, поэтому их рынок еще не насыщен, следовательно, существует большой потенциал для создания новых и инновационных VR-игр, которые могут стать хитами и принести большую прибыль разработчикам;

- VR-игры предлагают разработчикам новые возможности для экспериментирования с игровым дизайном;
- VR-технологии позволяют создавать игры, которые ранее были невозможны в традиционном формате, например, VR-игры могут включать в себя элементы социальной игры, многопользовательской игры или даже физической активности.

Существует уже множество инструментов и платформ для разработки VR-игр, которые значительно упрощают этот процесс.

Unity – это самая популярная в мире платформа разработки игр, на базе которой создано более 50% всех мобильных игр, 60% всего контента для дополненной и виртуальной реальности, а Unity-разработчик – это седьмая по росту популярности профессия согласно недавнему отчёту LinkedIn U.S. Emerging Jobs.

Unity – это инструмент для разработки двух – и трехмерных игровых приложений.

В двухмерных (2D) играх игровое пространство состоит из готовых плоских изображений. Задача компьютера сводится к комбинированию этих изображений в соответствии с алгоритмами игры, чем объясняются низкие требования двухмерных игр к мощности компьютера. Свобода передвижения персонажа в двухмерных играх ограничена: он может передвигаться только по предусмотренному разработчиками маршруту.

Компьютерная трехмерная игра (3D, англ. three-dimensional) – игра, визуальное пространство полностью построено из трехмерных объектов. Персонаж находится в трехмерном пространстве и в некоторых играх имеет полную свободу передвижения.

Управление в трехмерных играх, как правило, осуществляется одновременно с использованием клавиатуры и мышки. С помощью кнопок можно совершать движение персонажа в нужном направлении (обычно это клавиши W, S, A, D, реже – клавиши-стрелки). Движение мыши заставляет персонажа возвращаться. Нажатие левой кнопки мыши соответствует какому-либо действию – в зависимости от жанра игры это может быть, например, ударом мечом, взаимодействие с предметом или другим персонажем.



Рисунок 1 — Инструмент для разработки двух — и трехмерных игровых приложений Unity

Нажатие правой кнопкой мыши, как правило, в шутерах и ролевых играх соответствует блокировке атаки или альтернативной атаке, а в квестах — открытие меню. Клавиша **Space** обычно соответствует прыжку, а одновременное нажатие комбинации клавиш **Shift** и **W** заставляет персонаж бежать. Назначение других клавиш клавиатуры индивидуально для каждой игры.

Созданные с помощью Unity программы работают под операционными системами Windows, OS X, Windows Phone, Android, Apple iOS, Linux, а также на игровых приставках Wii, PlayStation 3, PlayStation 4, Xbox 360, Xbox One и MotionParallax3D дисплеях (устройства для воспроизведения виртуальных голограмм), к примеру, Nettlebox.

Unity также имеет возможность создавать приложения для запуска в браузерах с помощью специального модуля Unity (Unity Web Player) посредством реализации технологии WebGL.

Приложения, созданные с помощью Unity, поддерживают графические технологии DirectX и OpenGL.

В силу удобного интерфейса, простоты работы с движком, а также наличия бесплатной версии игрового движка, которая отличается от платной отсутствием не обязательных для не ТОПовых проектов функций, Unity3d является одним из наиболее популярных готовых движков для разработки компьютерных игр.

Встроенный в игровой движок Unity редактор обладает интуитивно понятным интерфейсом, который легко адаптировать под себя. Состоит интерфейс из нескольких окон и рабочих панелей и позволяет производить отладку игрового проекта непосредственно в редакторе. Отдельно следует упомянуть о встроенном редакторе ландшафта, который позволяет моделировать игровые локации непосредственно в движке, создавая сложную географию и накладывая текстуры прямо в редакторе. Движок поддерживает три скриптовых языка: C#, модифицированный JavaScript и Boo Script (диалект Python). С версии 4.0 появился встроенный редактор анимаций Mecanim, который позволяет использовать анимации со схожих персонажей.

Таким образом, с помощью Unity можно создавать кроссплатформенные компьютерные игры.

Под кроссплатформенностью подразумевается способность программного обеспечения работать более чем на одной аппаратной платформе и (или) операционной системе.

Рассмотрим понятие интерактивности и интерактивного игрового контента.

Интерактивность (от англ. interaction – «взаимодействие») – понятие, которое раскрывает характер и степень взаимодействия между объектами или субъектами.

В интерактивных системах приложение состоит из одних и тех же объектов и при разработке, и при выполнении. Мало того, отсутствие разбиения на инструментальную и исполнительную среду позволяет использовать одни и те же средства и при разработке, и при выполнении, поэтому можно изменять работающее приложение и немедленно видеть результат этого изменения.

Контент — любой вид информации (текст, аудио, видео, изображение, текстуры, материалы), составляющей содержание программного продукта.

Интерактивный игровой контент — это любой контент, который подразумевает активное участие пользователей и побуждающий их к совершению определённых действий.

Для того чтобы создать новое приложение, запускаем Unity и в стартовом окне нажимаем кнопку New. Ранее открывавшиеся проекты отображаются в центральной части окна (рисунок 2).

С целью создания нового проекта в появившемся окне:

- в поле Project name вводим название проекта, например, ARread;
- в поле Location указываем путь размещения проекта, например C:\my_Unity\, в этой папке будет создана папка с именем проекта, в ней будут размещаться все файлы проекта: например, C:\ my_Unity \ARread;
- поле Template выбираем тип создаваемого проекта — 2D или 3D (необходимо выбрать 3D).

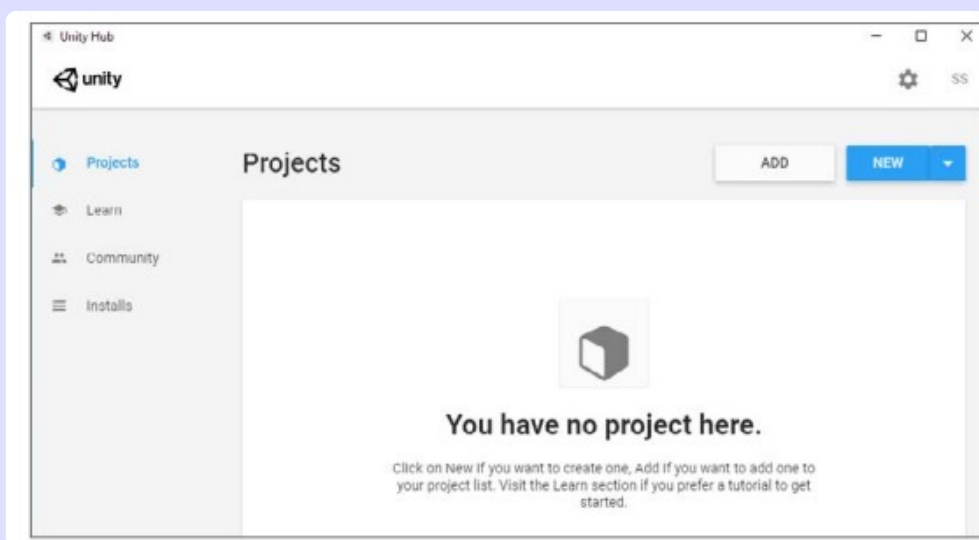


Рисунок 2 — Окно среды разработки Unity

Нажимаем на кнопку **Create**, после чего Unity начнёт создавать в указанной папке необходимые файлы. По окончании этого процесса откроется рабочее окно Unity.

Слева располагается окно Hierarchy, которое показывает структуру сцены и какие объекты на ней располагаются. На сцене 2 объекта: MainCamera — камера рендера и Directional Light — источник света.

Редактор Unity имеет простой **Drag&Drop** интерфейс (рисунок 3), состоящий из различных окон, благодаря чему можно производить отладку игры прямо в редакторе. Движок поддерживает два скриптовых языка: C#, JavaScript (модификация).

Все версии среды разработки Unity содержат интегрированный редактор проектов, встроенные ландшафты, шейдерную систему, сочетающую простоту использования, гибкость и производительность, поддерживают импорт графических и неграфических ресурсов (моделей, в том числе анимированных, текстур, скриптов и т.д.). Программирование графики в Unity осуществляется средствами **JavaScript**, **Boo** (диалект **Phyton**) и **C#** на основе **.NET**.

Расчёты физики производит физический движок **PhysX** от NVIDIA.

Проект в Unity делится на сцены (уровни) – отдельные файлы, содержащие свои игровые миры со своим набором объектов, сценариев и настроек. Сцены могут содержать в себе как, собственно, объекты (модели), так и пустые игровые объекты – объекты, не имеющие модели. Объекты, в свою очередь, содержат наборы компонентов, с которыми и взаимодействуют скрипты.

У любого объекта на сцене обязательно присутствует компонент **Transform** – он сохраняет в себе координаты местоположения, поворота и размеров объекта по всем трём осям. У объектов с видимой геометрией также по умолчанию присутствует компонент **Mesh Renderer**, который делает модель объекта видимой.

Для случая создания игрового приложения пользователем в качестве физического лица можно использовать бесплатную версию Unity. При этом есть некоторые ограничения: недоступен исходный код и нельзя осуществлять командную работу над проектом. Даже с этими ограничениями Unity остаётся надёжным инструментом для создания игр с нуля.

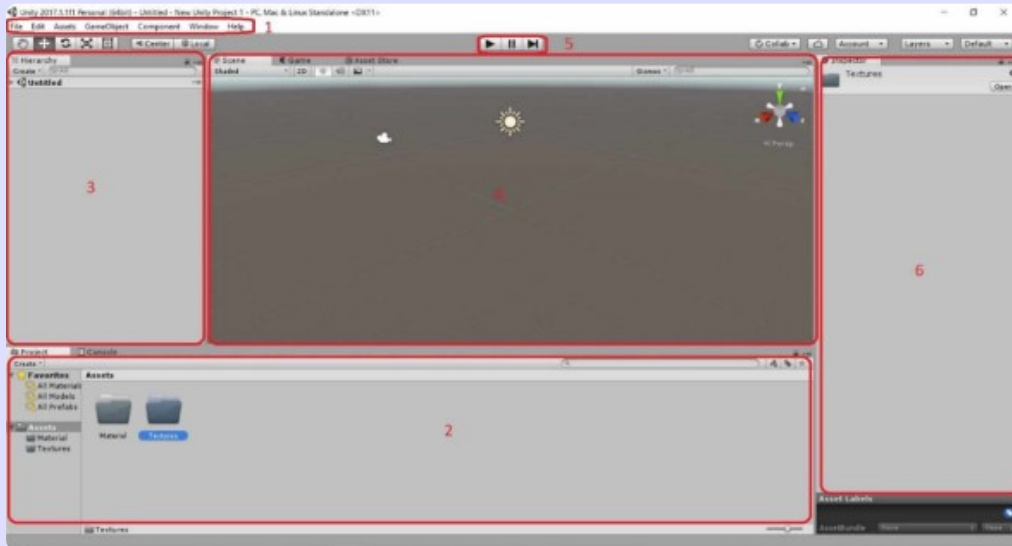


Рисунок 3 — Рабочее окно Unity

Каждый Unity – проект содержит папку **Assets**. Содержимое этой папки представлено в **Project View**. Это ресурсы игры: скрипты, 3D-модели, текстуры, аудиофайлы, префабы (**Prefabs**). Можно открыть папку, содержащую ресурс через стандартный Проводник **Windows**, нажав правой кнопкой мыши по нему и выбрав **Reveal in Explorer**.

Следует заметить, что при перемещении ресурсов внешними инструментами (например, Проводником) будут потеряны все метаданные, связанные с ним.

Метаданные сохраняют информацию о ресурсе и его связях с другими ресурсами. Всегда перемещайте ресурсы только в **Project View**.

Существует два способа добавления ресурса в проект:

1. Перетащите файл из Проводника в **Project View**.
2. На панели **Project View** выберите пункт **Assets** → **Import New Assets**.

Ресурс будет автоматически импортирован в проект и станет доступен для использования.

Сцены также хранятся в папке **Assets** и отображаются в **Project View**. Их можно считать отдельными уровнями.

Создать новую сцену можно с помощью меню **File → New Scene (Ctrl + N)**. Сохранение сцены можно выполнить с помощью меню **File → Save Scene (Ctrl+S)**.

Unity использует концепцию наследования (Parenting). Любой объект может быть дочерним по отношению к другому. Для этого достаточно перетащить его на «отца» в **Hierarchy**. Дочерний объект будет двигаться и вращаться вместе с родителем.

Главное меню Unity 3D стандартно располагается в верхнем левом углу окна (рисунок 4).

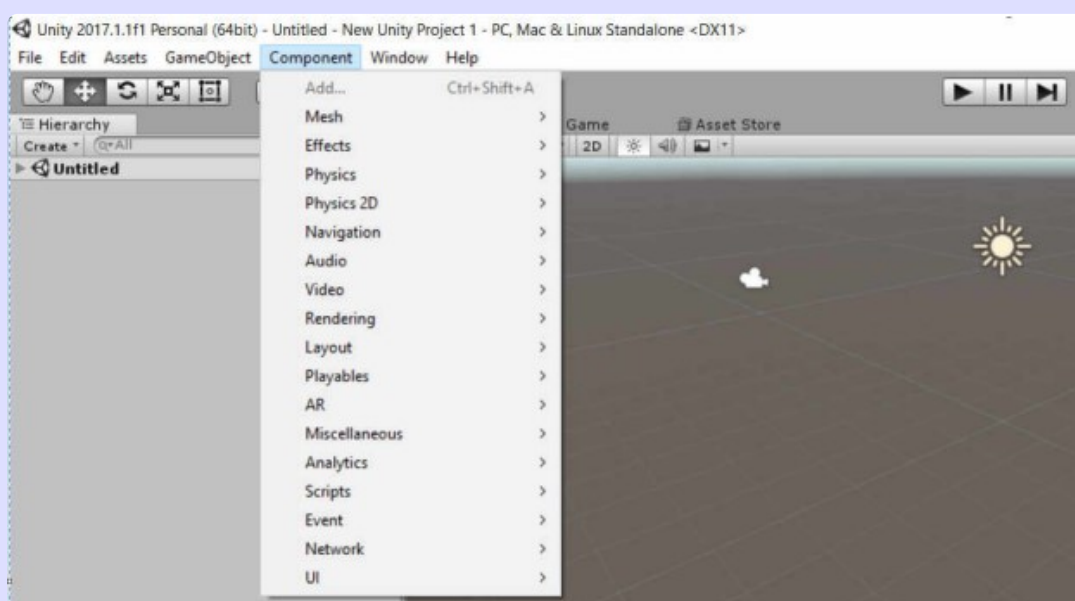


Рисунок 4 — Главное меню Unity

В главном меню содержатся вообще все команды программы, а в основных областях окна они только дублируются.

Панель инструментов (Toolbar) состоит из пяти базовых групп инструментов (рисунок 5). Каждая группа отвечает за свой участок редактора.

Transform Tools – инструменты трансформации объектов, используются в Scene View.

Transform Gizmo Toggles – инструменты для работы с Гизмоконтейнером, используются в Scene View.

Play/Pause / Step Buttons - инструменты просмотра игры, используются в Game View.

Layers Drop-down – определяет, какие объекты будут отображаться в Scene View.

Scene View используется для позиционирования объектов (окружение, персонажи, камеры, системы частиц и др.).

Unity использует движок **Ageia PhysX** от NVIDIA для симуляции физики.

Ageia PhysX является первым процессором, специально разработанным с целью совершить революцию в мире компьютерных игр благодаря невероятным физическим взаимодействиям в игре, позволяющей получить непревзойдённую динамику и реалистичность.

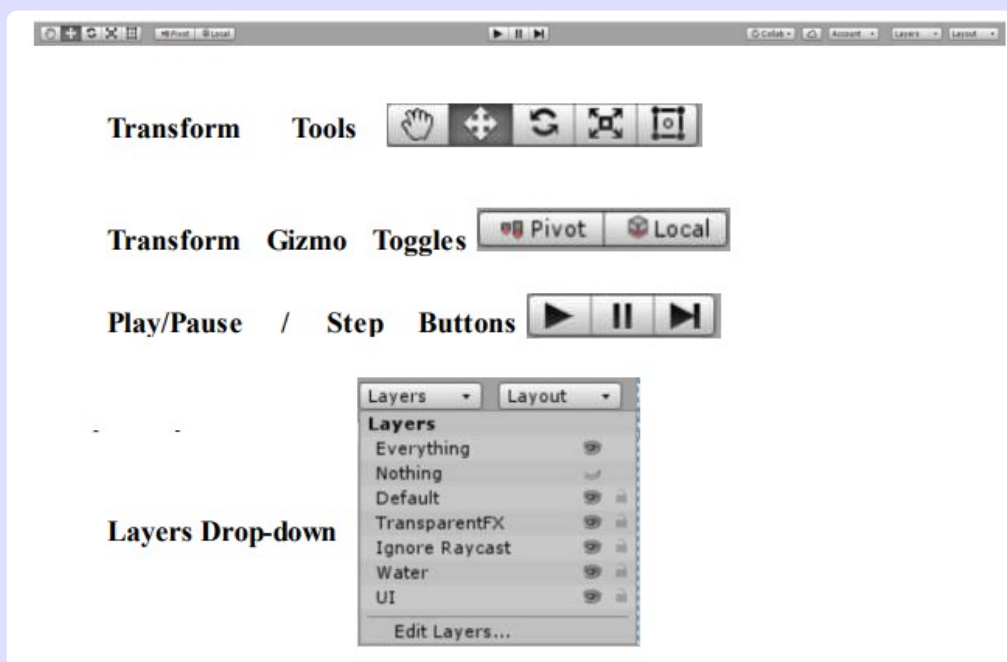


Рисунок 5 — Панель инструментов Toolbar

Для создания новой сцены следует выполнить: **File → New Scene**. Затем сохраним её: **File → Save As...**, указав имя primer. Добавим на сцену простейший 3D-примитив — куб. Для этого необходимо нажать на пункт меню **GameObject** и далее на закладки: **3D Object → Cube** (рисунок 6).

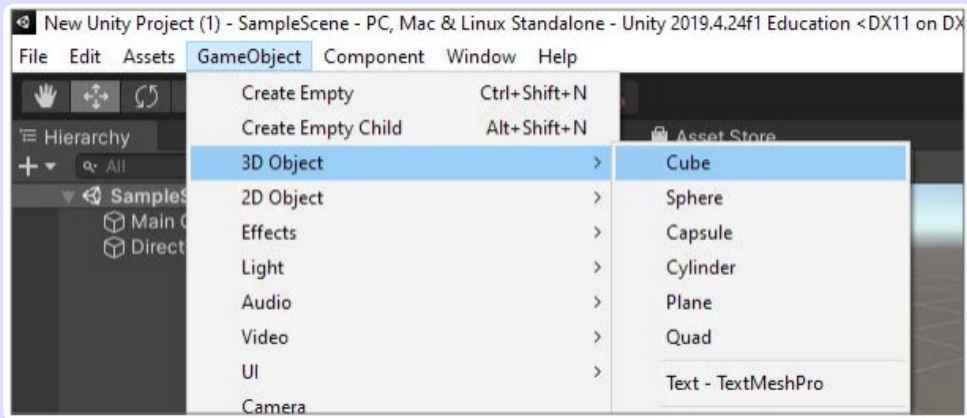


Рисунок 6 — Добавление на сцену 3D-примитива — куба

После добавления куба на сцену можно его увидеть в списке иерархии сцены и на самой сцене в окне **Scene** (рисунок 7).

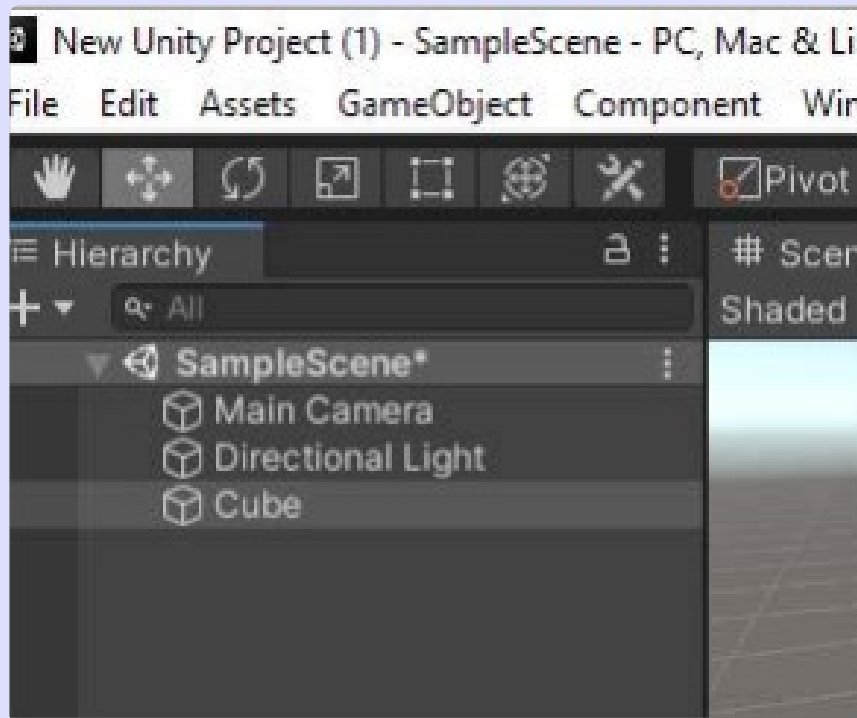


Рисунок 7 — 3D-примитив (куб) в окне Scene

Среда Unity может работать с 3D-моделями любой формы, создаваемыми в приложениях для моделирования. Однако существует ряд примитивных моделей, которые можно создать прямо в Unity: куб (Cube), сфера (Sphere), капсула (Capsule), цилиндр (Cylinder), плоскость (Plane) и квад (Quad).

Если нажать на вкладку **Game**, то можно увидеть, как сцена будет рендериться (рисунок 8).

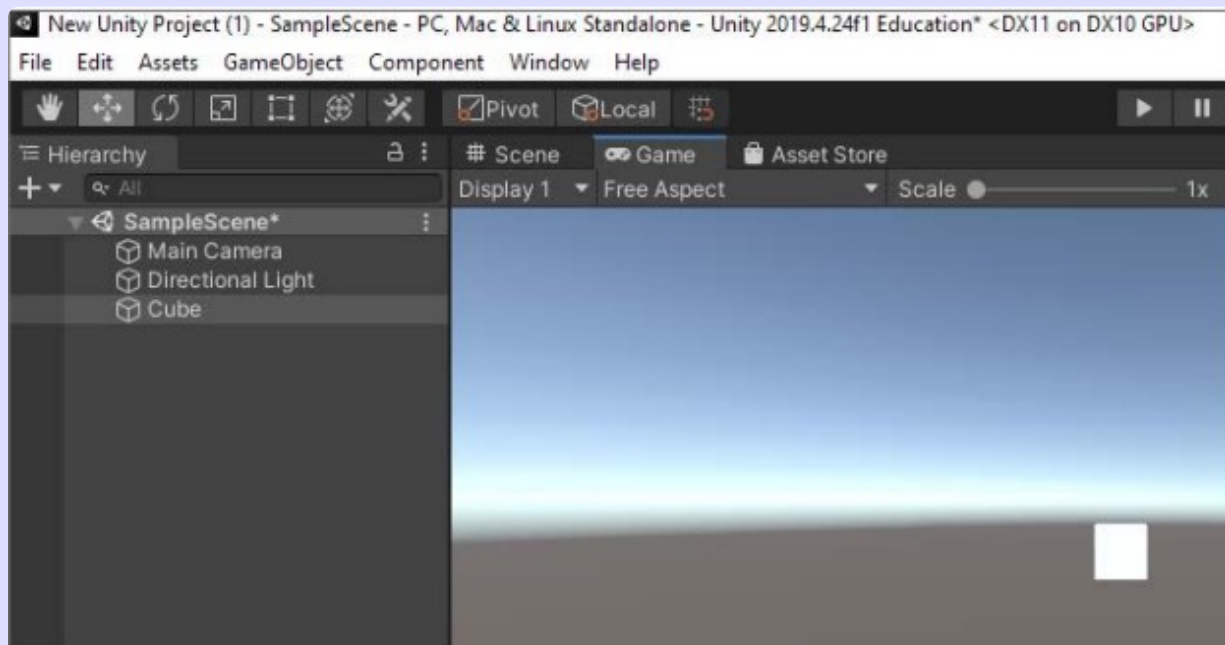


Рисунок 8 — Рендеринг сцены

Для того чтобы создать папку в окне **Project**, нужно правой кнопкой мыши нажать на: **Assets** → **Create** → **Folder**. Переименуйте папку в **Image**. Если в окне **Hierarchy** выбрать **Cube**, то справа в окне **Inspector** увидим список связанных с объектом компонентов.

Сегодня процессор **Ageia PhysX** является ключевым аппаратным элементом в обеспечении оптимизированной игровой физики. Его взаимодействующие **PhysX** ядра, обладающие невероятными возможностями для параллельных вычислений, оптимизированы под обработку динамической и крупномасштабной физики, чтобы обеспечить ускорение физического движения и взаимодействия на уровне, что сильно превышает масштабы и качество, предоставляемые обычными процессорами.

Поведение игровых объектов контролируется с помощью компонентов (**Components**), которые подключаются к ним. Несмотря на то, что встроенные компоненты Unity могут быть очень разносторонними, пользователю часто необходимо выйти за пределы их возможностей, чтобы реализовать ваши собственные особенности геймплея.

Unity позволяет создавать свои компоненты, используя скрипты. Они позволяют активировать игровые события, изменять параметры компонентов и отвечать на действия пользователя любым способом.

Unity поддерживает следующие языки программирования:

1. **C#** – язык подобный Java или C++;
2. **UnityScript** – язык, разработанный специально для использования в Unity по образцу JavaScript.

В редакторе Unity можно изменять свойства компонента, используя окно Inspector. Так, например, изменение позиции компонента Transform приведёт к изменению позиции игрового объекта. Аналогично вы можете изменить цвет материала компонента Renderer или массу твёрдого тела Rigidbody с ответным влиянием на отображение или поведение игрового объекта. В основном скрипты также изменяют свойства компонентов для управления игровыми объектами. Разница, однако, в том, что скрипт может изменять значение свойства постепенно со временем. За счёт смены, создания и уничтожения объектов в заданное время может быть реализован любой игровой процесс.

Чтобы появилась возможность воспроизведения звуков, нужно импортировать аудиофайлы в Unity-проект (рисунок 9). Процедура начинается с подбора файлов нужного формата, которые затем переносятся в Unity и настраиваются под цели проекта.

Форматы аудиофайлов, поддерживаемые Unity:

- **WAVE, WAV. Waveform Audio File Format (WAVE, WAV, от англ. Waveform – «в форме волны»)** – формат файла-контейнера для хранения записи оцифрованного аудиопотока, подвид RIFF. Этот контейнер, как правило, используется для хранения несжатого звука в импульсно-кодовой модуляции, однако контейнер не налагает никаких ограничений на использование алгоритма кодирования.
- **AIFF. Audio Interchange File Format (AIFF)** – формат аудиофайлов, который используется для хранения звуковых данных. AIFF был разработан компанией **Apple Computer** на основе формата IFF компании **Electronic Arts** и чаще всего используется в компьютерах **Apple Macintosh**. Звуковые данные в стандартном файле формата AIFF представляют собой несжатую импульсно-кодированную модуляцию. Также существует и сжатая версия формата AIFF, которую называют **AIFC** (иногда **AIFF-C**), в которой для сжатия могут быть использованы разные кодеки.

AIFF, наряду с **CDA** и **WAV**, является одним из форматов, используемых в профессиональных аудио — и видеоприложениях, так как в отличие от более популярного формата **MP3**, звук в **AIFF** кодируется без потерь в качестве. Как и любые сжатые файлы, файлы **AIFF** занимают гораздо больше дискового пространства, чем их сжатые аналоги.

MP3. **MP3** — кодек третьего уровня, разработанный командой **MPEG**, формат файла для хранения аудиоинформации. Формат был лицензированный, но 23 апреля 2017 года срок действия всех патентов истёк и лицензионное собрание прекратилось. Формат **MP3** использует спектральные отсеечения, согласно психоакустической модели. Звуковой сигнал разбивается на уровне по продолжительности отрезки, каждый из которых после обработки упаковывается в свой фрейм (кадр). Разложение в спектр требует непрерывности входного сигнала, в связи с этим для расчётов используется также предварительный и последующий фрейм.

В звуковом сигнале являются гармоники с меньшей амплитудой и лежащие гармоники вблизи более интенсивных — такие гармоники отсекаются, так как среднестатистическое человеческое ухо не всегда сможет определить наличие или отсутствие таких гармоник. Такая особенность слуха называется эффектом маскировки.

Также возможна замена двух и более соседних пиков одним усреднённым (что, как правило, и приводит к искажению звука). Критерий отсеечения определяется требованием к исходному потоку.

Поскольку весь спектр актуален, высокочастотные гармоники не отсекаются, а только избирательно удаляются, чтобы уменьшить поток информации за счёт разрежения спектра. После спектральной «зачистки» применяются математические методы сжатия и упаковка во фреймы. Каждый фрейм может иметь несколько контейнеров, что позволяет хранить информацию о нескольких потоках (левый и правый канал или центральный канал и разница каналов). Степень сжатия можно варьировать, в том числе в пределах одного фрейма. Интервал возможных значений битрейта составляет 8–320 кбит/с.

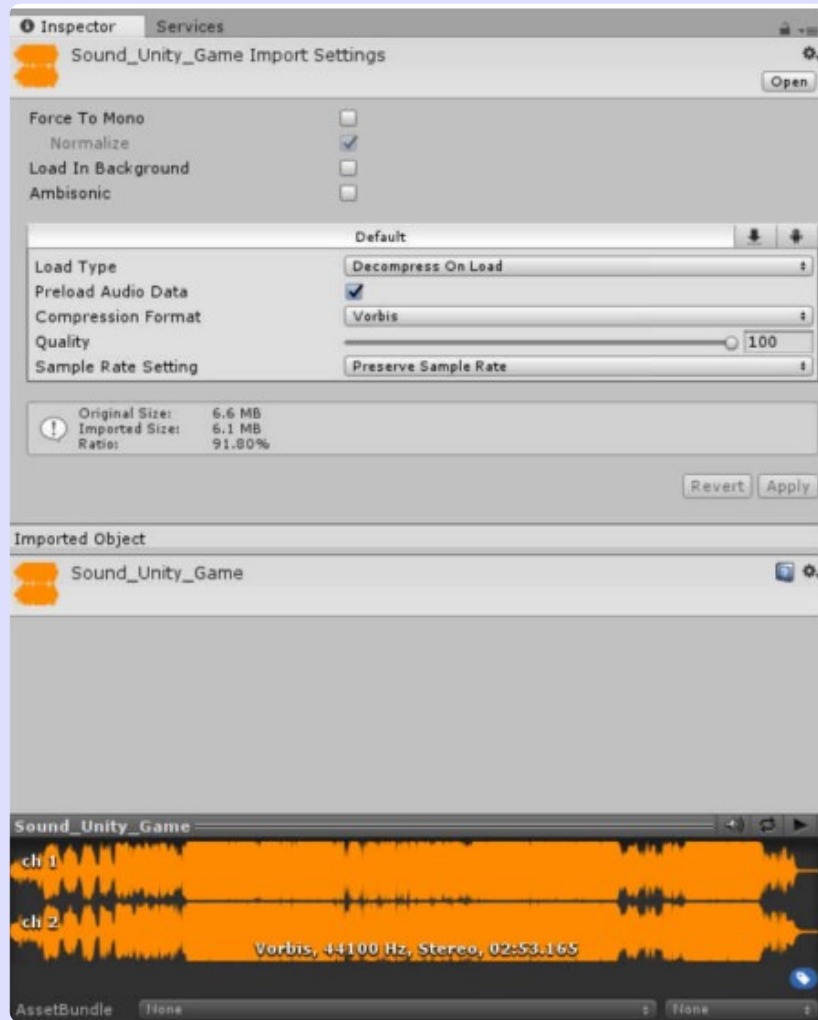


Рисунок 9 — Работа с аудиофайлами

Существует три версии формата **MP3** для различных потребностей: **MPEG-1**, **MPEG-2** и **MPEG-2.5**. Различаются они возможными диапазонами битрейта и частоты дискретизации.

OGG. Ogg – открытый стандарт формата мультимедиаконтейнера, который есть основным файловым и потоковым форматом для мультимедиакодексов фонда **Xiph.Org**, а также название проекта, занимающегося разработкой этого формата и кодеков для него.

Как и все технологии, разрабатываемые под эгидой **Xiph.Org**, формат **Ogg** является открытым и свободным стандартом, не имеющим патентных или лицензионных ограничений. Однако **Ogg** является всего лишь контейнером. Звук или видео сжимаются кодеками, а результат обработки хранится в подобных контейнерах. Контейнеры **Ogg** могут хранить потоки, закодированные несколькими кодеками. Например, файл с видео и звуком может содержать данные, закодированные аудио — и видеокодеками.

В контейнере **Ogg** можно хранить звук и видео в разных форматах (таких как **MPEG-4**, **Dirac**, **MP3** и другие), но обычно **Ogg** используется со следующими аудиокодеками:

1. с потерями:

Opus (ранее Harmony) – с низкой задержкой кодирования (от 2,5 мс до 60 мс, настраивается) и более высокой компрессией аудио, также битрейт от 6 до 510 кбит/с;

Speex – для сжатия речевого сигнала на низких битрейтах (~ 8–32 (кбит/с)/канал);

Vorbis – для сжатия звука на средних и высоких битрейтах (~ 16–500 (кбит/с)/канал).

2. без потерь:

FLAC — для обработки звуковых архивов и других аудио, требующих высокого качества воспроизведения.

MOD. MOD — формат файлов, разработанный для создания, хранения и воспроизведение музыкальных композиций на ПК. **Amiga**Свое название получил от того, что стал первым форматом, сохраняющим свои фрагменты (например, сэмплы) в других файлах (принцип модульности). Файлы этого формата имеют, как правило, расширение **.mod**. Каждый файл формата **MOD** включает в себя оцифрованные записи реального звучание инструментов — **сэмплы**. Композитор, пишущий в формате **MOD**, использует программу, которая называется трекером, в которой указывает, какой именно инструмент, в какое время, какой нотой и какой из октав должен прозвучать.

Последовательность нот записывается в список — **трек**, а несколько параллельно воспроизводимые треки образуют блок, который называется **паттерном**. Паттерны, создаваемые композитором, получают номера, после чего композитор может в произвольной форме указывать какой паттерн и когда должен прозвучать. Совокупность паттернов и образует модуль — файл в формате **MOD**.

XM. Формат **XM** — это расширенный модуль **MOD**, тип аудиофайла **FastTracker 2**, который был введён разработчиком — демогруппой **Triton**. XM-файл является мультисемплингом с помощью доступных инструментов с объёмным сигналом в панорамной оболочке, а также сжатой структурой. Формат расширил список доступных команд эффектов и каналов, добавил 16-битную поддержку и предложил альтернативную таблицу частот для портаменто (способ выполнения, при котором следующая нота не сразу берётся точно (в звуко-высотном отношении), а используется плавный переход к нужной высоте от предыдущей ноты). XM является главным форматом для большинства трекерной музыки.

Коллекцию аудиофайлов необходимо импортировать в Unity.

Для воспроизведения звука в Unity необходимо добавить три компонента: **AudioClip**, **AudioSource** и **AudioListener**.

Контрольные вопросы:

1. Что собой представляет среда Unity?
2. Какие основные окна интерфейса среды Unity можете назвать?
3. Для чего нужен рендер сцены в Unity?
4. Как создать сцену?
5. Как создать новую папку в окне Project?
6. Какими свойствами обладают объекты в Unity?

Занятие № 2 Анимация в Unity и создание игрового проекта.

Тема: Особенности реализации анимации в Unity.

Цель: знакомство с возможностями реализации анимации межплатформенной средой разработки игровых приложений Unity.

Задачи:

- изучение особенностей реализации анимации и моделирования физических свойств трехмерных объектов в среде Unity;
- изучение физических свойств трехмерного тела (модели) в среде Unity;

- освоение приёмов организации взаимодействия объектов в среде Unity.

Оборудование:

- компьютер (ноутбук);
- проектор;
- интерактивная доска;
- очки дополненной реальности;
- смартфон;
- веб-камера;
- МФУ.

Список источников:

1. **Ферроне Харрисон.** Изучаем C# через разработку игр на Unity. 5-е издание – СПб.: Питер, 2022. – 400 с.: ил.
2. **Джереми Гибсон Бонд.** Unity и C#. Геймдев от идеи до реализации. 2-е издание – СПб.: Питер, 2022. – 928 с.
3. **Джозеф Хокинг.** Unity в действии. Мультиплатформенная разработка на C# – СПб.: Питер, 2019. – 352 с.
4. **Программирование искусственного интеллекта в играх.** Руководство / Паласиос Хорхе – М.: ДМК Пресс, 2017. – 274 с.
5. **Доусон М. Д71.** Изучаем C++ через программирование игр – СПб.: Питер, 2016. – 352 с.: ил. Хорхе, Паласиос Unity 5.x.

Unity 3D поддерживает следующие типы анимации:

- анимация твердого тела;
- анимация на основе скелета;
- анимации спрайтами (отдельные спрайты, атлас спрайтов);
- анимация, основанная на физике (использование физической системы Unity);
- видеоанимация (воспроизведение видеофайлов в виде анимированных текстур);
- анимация частицами;
- анимация персонажей с помощью системы Mecanim.

Анимация твёрдого тела

Анимация твёрдого тела используется для создания готовых последовательностей анимации, которые перемещают или изменяют свойства объектов с учётом того, что объекты являются единым целым.

При анимации твёрдого тела изменения в ключевых кадрах распространяются только на целые объекты и их свойства высокого уровня. Они не касаются свойств и внутренних компонентов и не изменяют сущности или собственной формы объектов.

Анимация на основе скелета

Этот тип анимации изменяет в ключевых кадрах не положение, вращение или масштаб объекта, а движение и деформацию его внутренних частей. Как правило, анимация на основе скелета создаётся как полная последовательность анимации в программе 3D-моделирования и импортируется в Unity как часть файла, который может быть обработан и доступен посредством системы анимации Mecanim.

Анимация спрайтами

Анимация состоит из последовательности изображений или кадров и проигрывает их в нужном порядке и с заданной скоростью для получения непрерывной анимации.

Анимация, основанная на физике (использование физической системы Unity)

Данный тип анимации позволяет управлять поведением объектов с помощью физической системы Unity. Например, падение объекта на землю под действием силы тяготения. Видеоанимация (воспроизведение видеофайлов в виде анимированных текстур) – возможность воспроизводить видеофайлы в виде анимированных текстур как на настольных платформах, так и на мобильных устройствах, таких как iOS и Android.

Анимация частицами

Использование системы частиц сюрикен (Shuriken particle system) для создания дождя, снега, фейерверков, искр и в другой нематериальной анимации из множеством подвижных частиц.

Анимация персонажей с помощью системы Mecanim

Возможность создавать анимированные персонажи с помощью системы анимации Mecanim.

Unity предлагает широкий спектр инструментов для создания анимации. Эти инструменты называются встроенными функциями анимации, они включают в себя следующее:

1. **Редактор анимации Unity (Unity animation editor)** для анимации жёстких тел, например, для создания дверей, летающей камеры, платформы лифта и другого.
2. **Система частиц сюрикен (Shuriken particle system)** для создания дождя, снега, фейерверков, искр и в другой нематериальной анимации с множеством подвижных частиц.

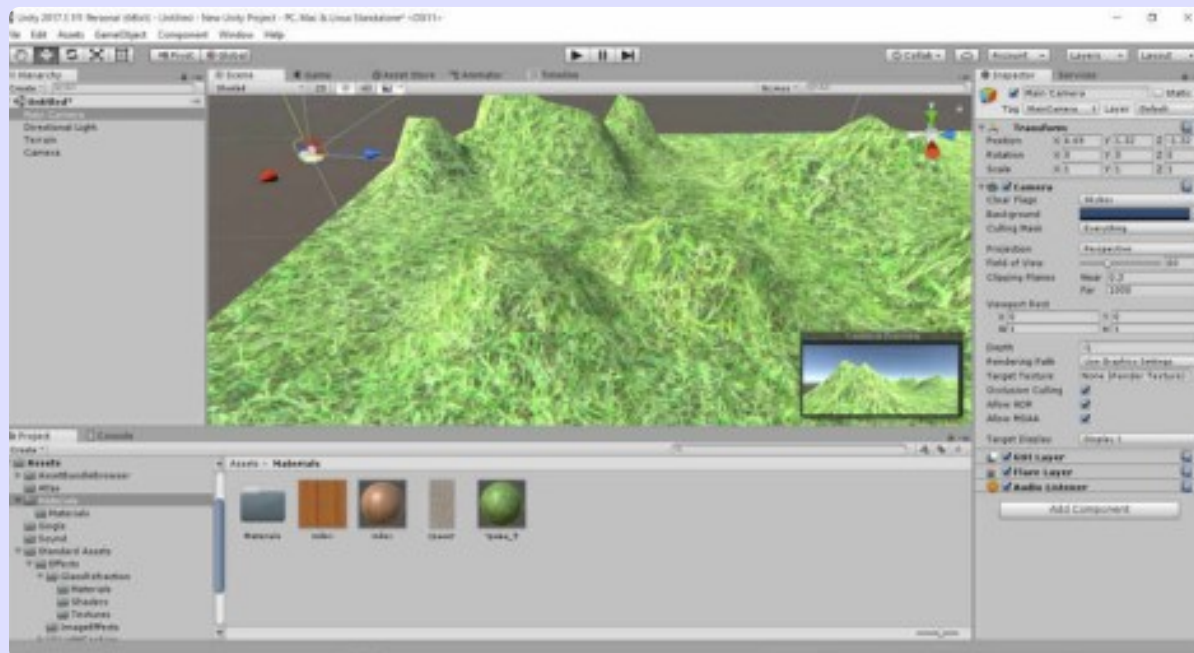


Рисунок 11 — Создание ландшафта и камеры

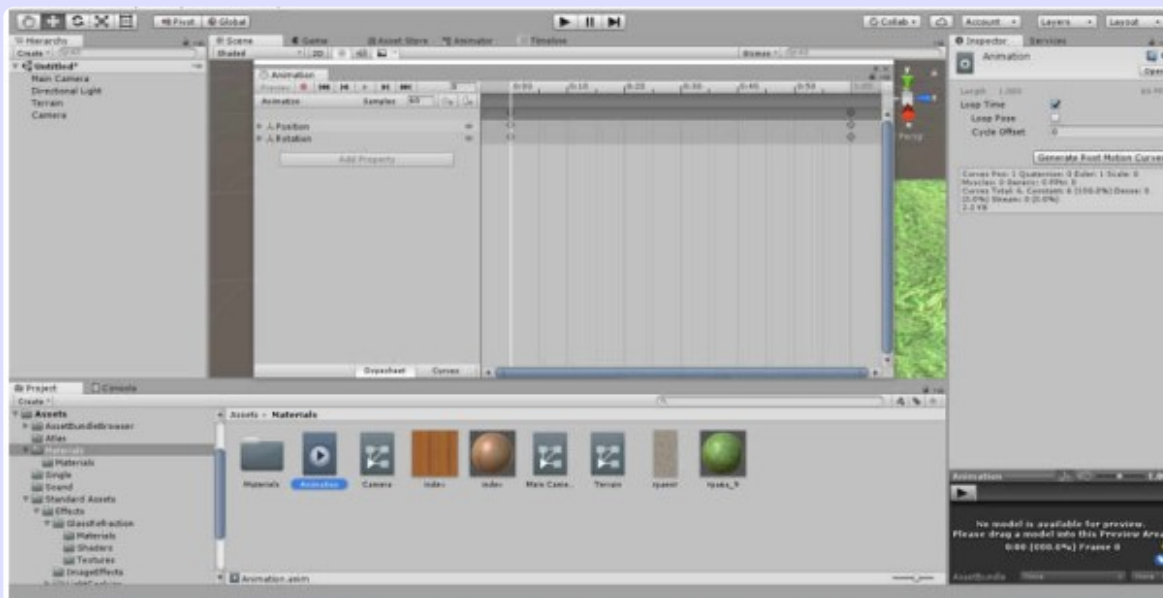


Рисунок 12 — Создание клипа анимации

Создание игрового проекта для Windows

Под созданием кроссплатформенных приложений подразумевается генерация прикладных пакетов, запускаемых на разных платформах. На каждой платформе (Windows, Android и др.) своя форма пакета, но как только вы сгенерировали исполняемый файл, появляется возможность распространять игру и играть в неё без привязки к Unity.

Один проект Unity можно развернуть на разных платформах – его не нужно каждый раз генерировать заново.

Просмотреть все варианты платформ можно в окне **Build Settings**. Открыть окно **Build Settings** можно с помощью меню **File Build Settings** (рисунок 13).

Окно **Build Settings** содержит минимальные настройки, необходимые для сборки проекта. Первое, что необходимо сделать, – добавить в проект игровую сцену. Сделать это можно с помощью кнопки **Add Open Scenes**, или просто переместить файл сцены из окна проекта в поле **Scenes In Build**.

Необходимо также обращать внимание на цифру ноль справа от названия файла сцены. Это её **порядковый номер**. Если сцен в проекте несколько – необходимо помнить, что сцена, которая должна будет загружаться первой, должна получить идентификатор ноль. Иначе Unity автоматически загрузит вместо вводного ролика сцену, что случайно получило главный идентификатор.

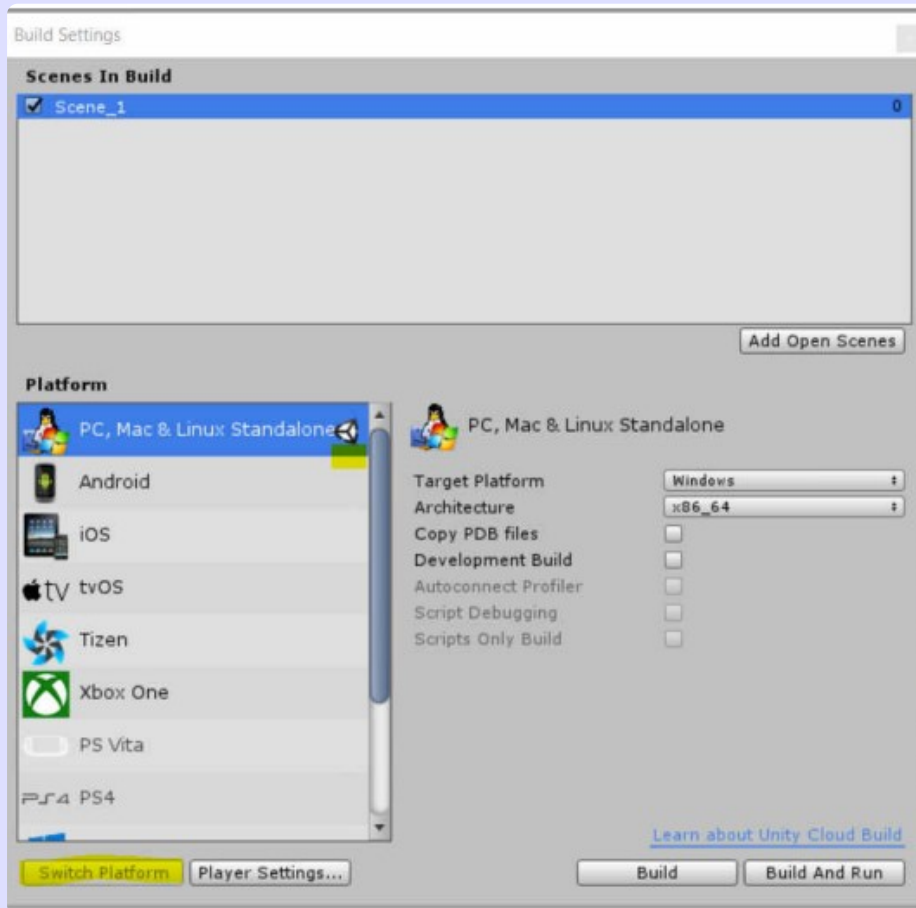


Рисунок 13 — Окно Build Settings

Список платформ, поддерживаемых Unity, представлен в левой части окна.

Активные платформы отмечены значком Unity. Достаточно выделить платформу из этого списка и нажать кнопку **Switch Platform**.

Далее справа на вкладке **Target Platform** необходимо выбрать платформу «**Windows**», на вкладке «**Architecture**» выбирать архитектуру, которая будет поддерживаться проектом. Для генерации игрового проекта в нижней части окна находятся кнопки **Build** и **Build and Run**. Кнопка **Build and Run** отличается от кнопки **Build** тем, что автоматически запускает сгенерированное игровое приложение.

Открыть список настроек игрового приложения на панели **Inspector** можно с помощью кнопки **Player Settings** (рисунок 14). Эти настройки контролируют разные аспекты готового приложения.

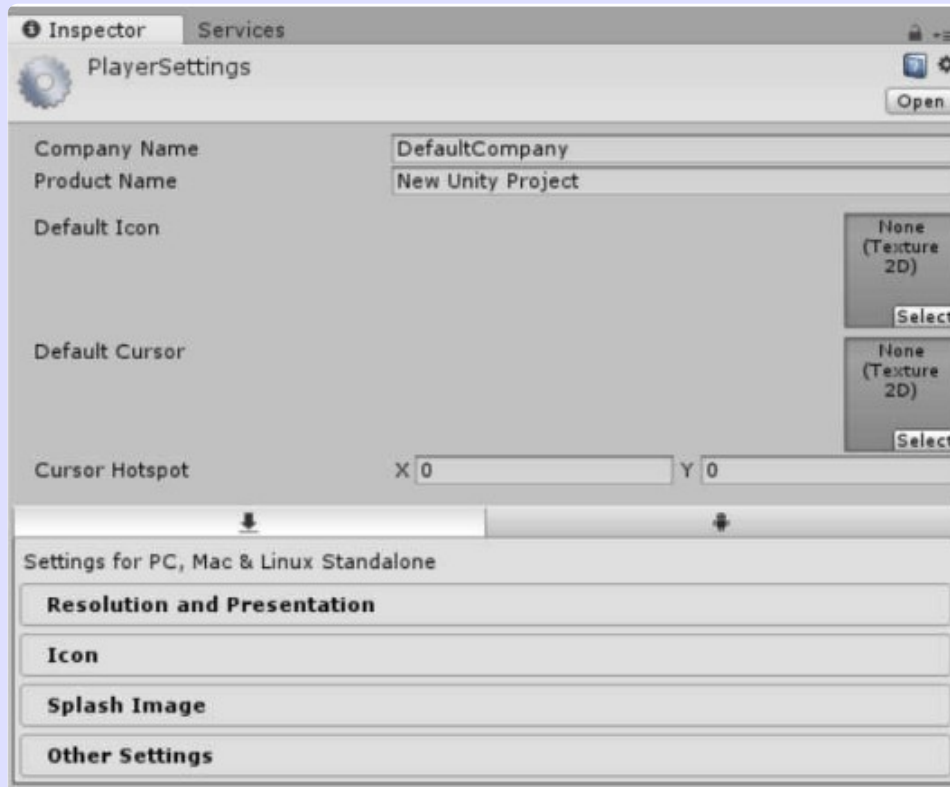


Рисунок 14 — Окно Player Settings

Поля **Company Name** и **Product Name** позволят указать название игры и разработчика. С помощью **Default Icon** можно установить игровую иконку приложения. Для чего необходимо выбрать изображение (при необходимости импортировать его) на вкладке **Project** и перетащить его в раздел **Texture 2D** на панели **Inspector**. Кроме того, с помощью **Default Cursor** можно также задать изображение курсора.

После нажатия на кнопку **Build** откроется окно выбора файла, в котором следует указать адрес для генерации пакета игрового приложения. Сразу после указания местоположения начнётся процесс постройки, после чего Unity создаст исполняемый файл для активной в данный момент платформы.

В нашем случае это платформа Windows.

В дополнение также влияют настройки, доступ к которым осуществляется с помощью меню **Edit**. В частности, именно здесь настраивается визуальное качество готового приложения.

Unity может генерировать файлы формата **APK (Android Application Package)**. Для этого необходимо добавить в Unity путь к среде разработки **Android SDK**, которое уже содержит требуемый компилятор. **Android** – операционная система (ОС) для смартфонов, планшетных компьютеров, электронных книг, цифровых проигрывателей, нетбуков, смартбуков, очков Google, телевизоров, систем автоматического управления автомобилем и другими устройствами. ОС базируется на **Linux-ядре** и собственной реализации виртуальной машины **Java** от Google.

Контрольные вопросы:

1. Что собой представляет анимация?
2. Какие виды анимации можно реализовать в среде Unity?
3. Как осуществляется настройка анимаций 3D-модели в Unity и использование их в AR.

Занятие № 3 3D-проектирования Blender в среде Unity.

Тема: пакет 3D-проектирования Blender и его использование в среде Unity.

Цель: знакомство с возможностями интегрирования объектов Blender в межплатформенную среду разработки игровых приложений Unity.

Задачи:

- изучение особенностей реализации моделирования физических свойств трехмерных объектов в Blender;
- изучение физических свойств трехмерного тела (модели) в среде Blender;
- освоение приёмов организации взаимодействия объектов в среде Blender.

Оборудование:

- компьютер (ноутбук);
- проектор;
- интерактивная доска;
- очки дополненной реальности;

- смартфон;
- веб-камера;
- МФУ.

Список источников:

1. **МАРИЯ СЕРОВА** Учебник-самоучитель по трехмерной графике в BLENDER 3D. Моделирование, дизайн — М.: Солон-пресс 2021. — 272 с.
2. **Прахов, А.А.** Самоучитель Blender 2.6 — СПб.: БХВ-Петербург, 2013. — 384 с.
3. **Ферроне Харрисон.** Изучаем C# через разработку игр на Unity. 5-е издание — СПб.: Питер, 2022. — 400 с.: ил.
4. **Джереми Гибсон Бонд.** Unity и C#. Геймдев от идеи до реализации. 2-е издание — СПб.: Питер, 2022. — 928 с.
5. **Джозеф Хокинг.** Unity в действии. Мультиплатформенная разработка на C# — СПб.: Питер, 2019. — 352 с.

Одной из интересных разработок, которая может быть реализована обучающимися на начальном этапе изучения возможностей данного программного продукта, является сбор сцены с игровой механикой стрельбы из лука по противникам (рисунок 15).

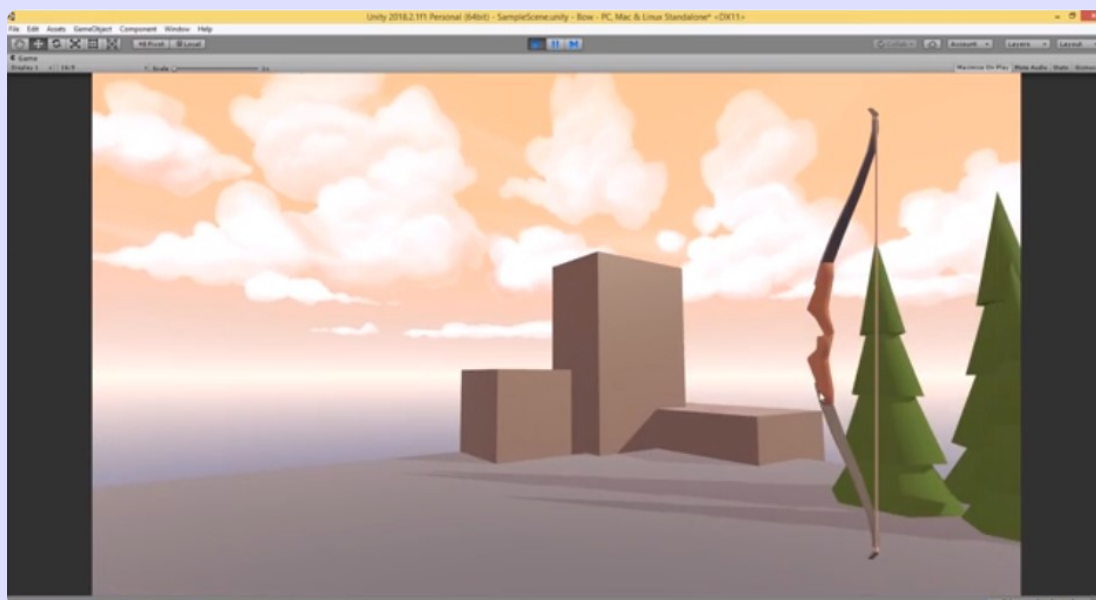


Рисунок 15 — Сбор сцены с игровой механикой стрельбы из лука по противникам

Совместимость форматов Blender и Unity

Следует отметить, что в ряде проектов могут быть использованы готовые 3D-модели игрового окружения. Подобные модели могут быть разработаны в разных приложениях, в частности, в программе Blender (юрты, модели противников, живность, деревья и др.).

Мир игровой разработки устроен так, что чем больше у разработчика инструментов, тем больше возможностей.

Blender — это один из самых мощных доступных инструментов, который по праву занимает ведущие позиции в графике и анимации. Это приложение повышает гибкость игровой разработки и в конечном итоге помогает сделать игру качественнее.

Blender – важный компонент в составе инструментария начинающего разработчика игр в Unity.

Blender (рисунок 16) больше десяти лет остаётся любимым инструментом инди-студий и небольших команд. И это неудивительно, ведь Blender — это бесплатное открытое ПО, возможности которого впечатляют. Более того, Blender регулярно обновляется, поддерживая актуальность технологий, и позволяет создавать высококачественные визуализации.

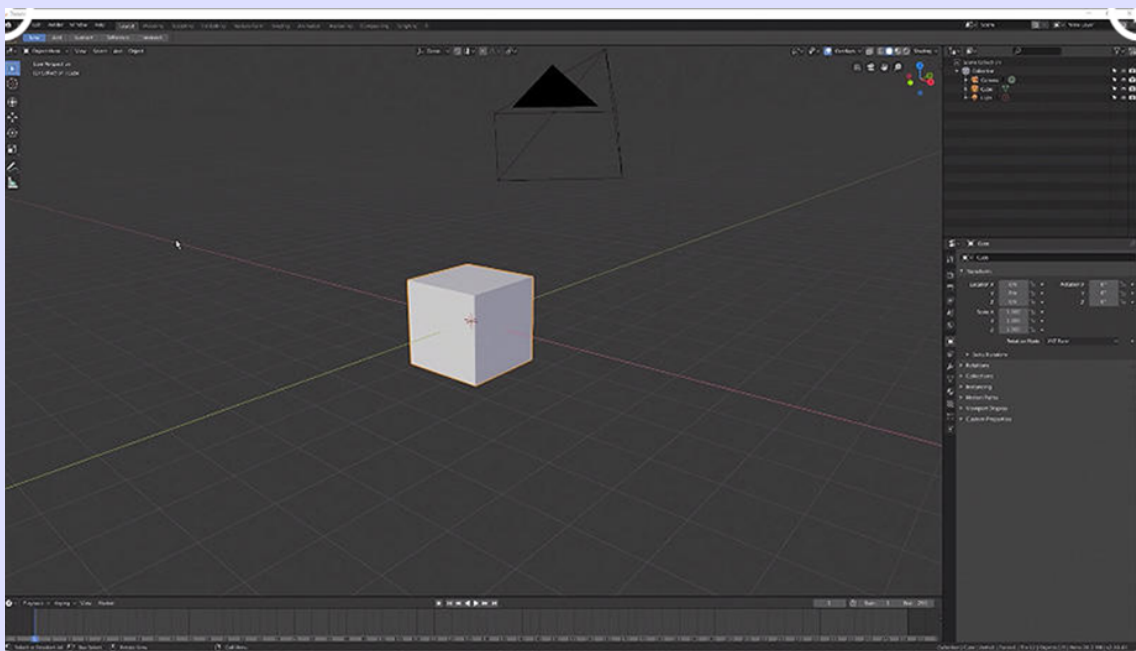


Рисунок 16 — Рабочее окно Blender

Возможности **Blender** не ограничиваются только графикой или анимацией, выходя далеко за пределы этих задач, объединяя множество других функций и преимуществ:

- полный цикл разработки 3D-моделей (игровые ассеты);
- разработка выдающихся визуальных эффектов;
- сборка полноценных сцен и окружений;
- рендеринг всей сцены (получение изображения из трехмерной модели);
- поддержка дополнений, расширяющих существующие возможности.

Запуск программы **Blender** осуществляется через главное меню или по ярлыку на рабочем столе. Все пункты меню, команды и названия окон после запуска программы отображаются на английском языке. Для выбора русскоязычного интерфейса следует выполнить следующие действия: **File** → **User preferences** → вкладка **System** → в окне **International fonts** выбрать язык **Русский**® отметить пункты **Interface**, **Tooltips**, **New Data** → нажать кнопку «**Сохранить настройки**».

Центральное место в рабочем окне редактора занимает 3D-окно, в котором размещены следующие элементы: 3D-окно с обозначением осей, 3D-курсор, отображаемый объект (по умолчанию — куб), лампа, камера. Все вместе (в данном случае, куб, лампа, камера) формируют сцену — представление события.

Сцена, созданная в Blender, — это ещё не изображение. Для того чтобы получить из сцены графический файл, необходимо «отрендерить» файл. В Blender, чтобы посмотреть конечное изображение объекта, следует нажать клавишу **F12**. Отображаемая сторона, удалённость и др. на получившемся изображении зависят от того, где размещена и как повернута камера.

Необходимость конвертировать файлы стороннего проекта для использования их в других приложениях — это долгий процесс. Кроме того, он может быть сопряжён с ошибками, даже в случае программ, гарантирующих совместимость. Нативный импорт файлов в проект из Blender в Unity очень прост (рисунок 17).

Для импорта ресурсов из Blender в Unity необходимо выбрать пункт меню **Assets > Import New Asset** на панели меню Unity, а затем найти и открыть файл.blend.

Blender предоставляет широкие возможности для моделирования, одним из преимуществ этого программного продукта является то, что в нем собраны возможности для моделирования различными методами: это и инструменты для скульптурирования, и возможности работы со сплайнами, кривыми Безье, и полигональное моделирование. Каждый из методов имеет свои преимущества и недостатки. В результате моделирования мы получаем модель, которую можем в этом же редакторе оттекстурировать, а также задать анимацию. Полученную модель можно использовать как для создания видео или статичной картинки в самом Blender, так и экспортировать модель в другие программные продукты, где использовать для конечных целей. Например, в Unity для использования модели при создании компьютерной игры.

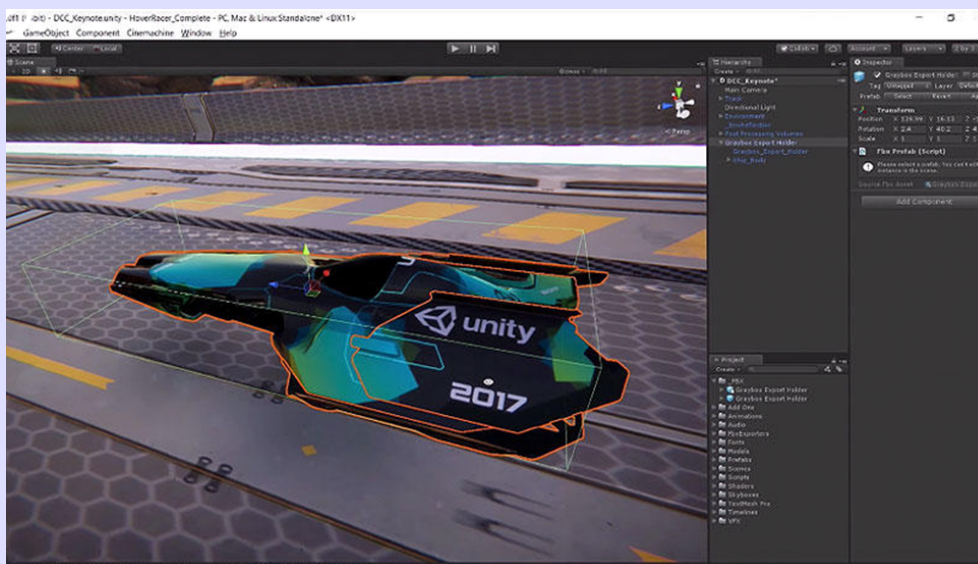


Рисунок 17 — Бесшовный импорт из Blender в Unity

Перед экспортом объекта необходимо провести некоторые подготовительные работы:

1. проверить размеры созданного 3D-объекта. Необходимо учитывать, что в Blender единицы измерения по умолчанию одни, а в Unity — другие. Несомненно, размеры объекта можно подгонять и в игровом движке, однако стоит учитывать, что при уменьшении или увеличении размеров объекта могут вылезти недостатки, которые при изначальных размерах не были заметны;

2. проверить направления нормалей. Как правило, в Blender в процессе создания видны полигоны с неправильным направлением нормалей, однако их можно не заметить, если не просмотреть объект со всех сторон. Данную ошибку можно исправить в игровом движке, однако подобные методы будут ресурсозатратными, поэтому лучше проводить пересчёт нормалей в 3D-редакторе;
3. проверить направления локальных осей 3D-редактора и игрового движка. Одной из главных проблем при импорте моделей является путаница осей, в которых экспортируется готовый объект. В Blender осью, направленной вверх, является ось z, а осью, определяющей направление «лица» модели (forward, перед), является ось -y. Это можно увидеть на рисунке 1, а в Unity вертикальной осью является ось y, а «передом» модели — ось z. Таким образом, для правильного экспорта в Unity 3D-модели в Blender модель должна смотреть лицом в направлении оси -y (минус y), а верхушкой — в направлении положительной оси z;
4. трансформации 3D-модели должны завершаться сбросом углов поворота модели на значения по умолчанию. Для выполнения сброса углов поворота и совмещения локальных осей 3D-модели (или оси сразу нескольких 3D-моделей) с глобальными осями координат необходимо выбрать 3D-модель в режиме объекта и нажать сочетание клавиш **Ctrl+A**, вызвав в открывшемся меню **Apply**. В этом меню необходимо выбрать пункт **Rotation**. В меню **Apply**, помимо сброса поворота модели и всего остального, присутствует ещё и пункт сброса масштаба модели. Это связано с тем, что после масштабирования 3D-модели на сцене в режиме объекта значение параметров масштабирования также необходимо сбросить на значения по умолчанию, выбрав в упомянутом выше меню **Scale**;
5. проверить местоположение точки **Origin**. Для объектов, не имеющих скелета, скорее всего, точка **Origin** была пересчитана перед созданием им анимации, однако если речь о персонажах, то в их анимации точка **Origin** не принимала участия, а для правильной работы персонажа в игровом движке необходимо точку переместить на уровень ступней персонажа, таким образом, персонажи будут ходить по плоскости, а не проваливаться под неё;
6. почистить историю, чтобы при экспорте не увеличивался размер файла ненужной информацией;

7. проверить правильность имен импортируемых объектов. Правильное наименование экспортированных объектов из 3D-редактора необходимо для ускорения процесса настройки объектов в игровом движке. Названия объектов следует писать на английском языке без дополнительных символов в начале (например, тире), так как списки объектов в игровых движках располагаются, как правило, в алфавитном порядке;
8. следует соблюдать чистоту сцены. Пустые или лишние скрытые объекты могут импортироваться в файл **FBX** вместе с объектом. В результате чего размер файла увеличится и возникнет неудобство в работе с ним в игровом движке;
9. следует сохранять текстурные карты в правильном формате. Предпочтительным форматом текстурных карт является **PNG**. Данный формат прост и удобен в редактировании, корректно сохраняет картинку и имеет альфа-канал. Формат **PSD** также поддерживается игровыми движками (автоматически сливает слои), удобен, когда текстура находится в процессе изготовления, но для использования в игровом движке следует заменить этот тяжелый многослойный формат, так как он сильно увеличивает размер проекта игры. Более подробно о чтении форматов текстур игровыми движками можно прочитать в документации к ним;
10. проверить созданные зацикленные анимации — первый и последний кадр должны совпадать. Малейшее отклонение по оси, направленной вверх, (как правило, оси Y) может привести к постепенному проваливанию или поднятию персонажа относительно плоскости в процессе игры;
11. проверить объекты на наличие полигонов, просматриваемых с двух сторон;
12. проверить растровую карту на соответствие текстур UV развертке в местах близкого расположения частей с разными текстурами;
13. проверить в анимации корректность перемещения полигонов.

В процессе работы над созданием игрового виртуального мира могут быть использованы разные объекты – элементы ландшафта (деревья, возвышенности и др.), модели противников, разнообразная флора и т. п., особенности которых, а также учет физики визуализируемого процесса (стрельба из лука) и тонкости его реализации обеспечат в дальнейшем занимательную игру (рисунок 18, рисунок 19, рисунок 20, рисунок 21).

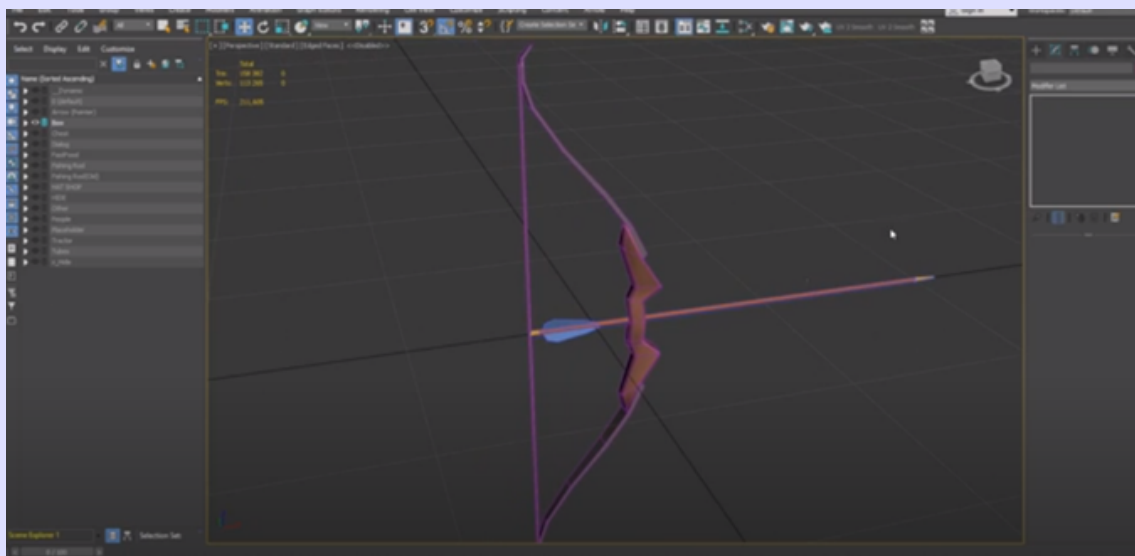


Рисунок 18 – Создание виртуального мира (элементы для реализации стрельбы из лука)

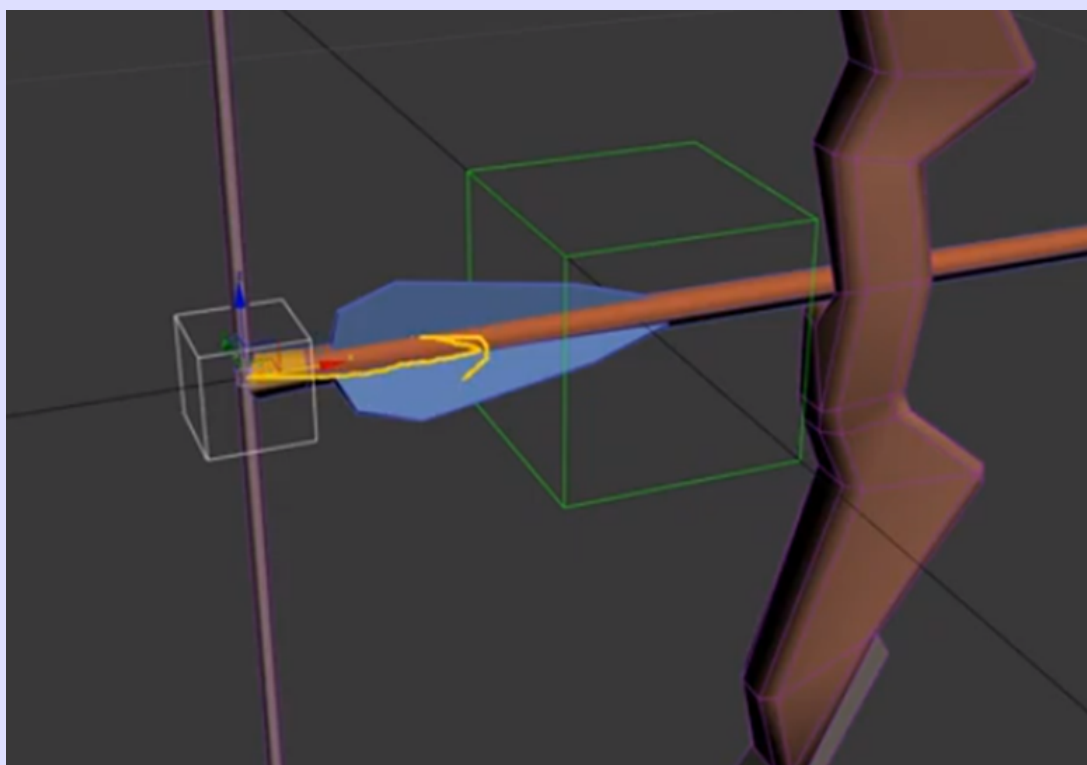


Рисунок 19 – Создание виртуального мира (элементы для реализации механики стрельбы)

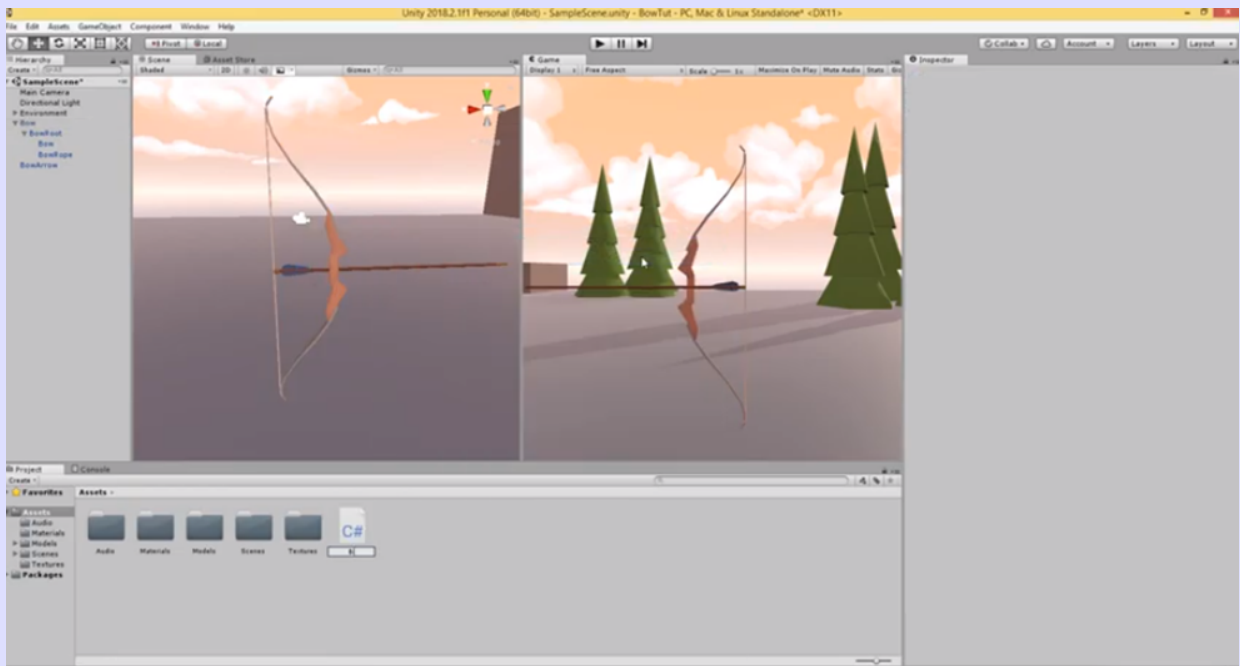


Рисунок 20 – Создание виртуального мира (элементы для реализации стрельбы из лука)

В результате работы в Unity с применением Blender в формате интенсива осуществляется первичное знакомство обучающихся со средами разработки игр и приложений виртуальной реальности. В итоге после сборки собственного виртуального мира каждый участник мастер-класса получает возможность запустить игру в формате виртуальной реальности в VR-шлеме и поиграть с использованием инновационных технологий.



Рисунок 21 – Игра в стрельбу из лука в формате виртуальной реальности

Новейшие технологии играют важную роль в обучении детей с физическими, социальными или когнитивными нарушениями. Ведь с помощью иммерсивных технологий можно создать инклюзивную учебную среду с учетом потребностей и возможностей каждого, что призвано стать одним из важных шагов в демократизации знаний.

Таким образом, технологии виртуальной и дополненной реальности обеспечивают обучающимся возможность глубже изучать предметы, анализировать последствия мировых событий, участвовать в экспедициях на Луну и многое другое, а главное — в развлекательной форме. AR и VR позволяют приобрести опыт, к которому у обучающихся нет доступа.

Контрольные вопросы:

1. Что собой представляет 3D-моделирование?
2. Какие программы 3D-графики вы знаете?
3. Какие элементы содержит рабочее окно Blender?
4. Как добавить новый объект в 3D-окно?

