**ЦЕЛЬ РАБОТЫ:**

Познакомится с окружением игрового движка Unity.

**ЗАДАЧИ:**

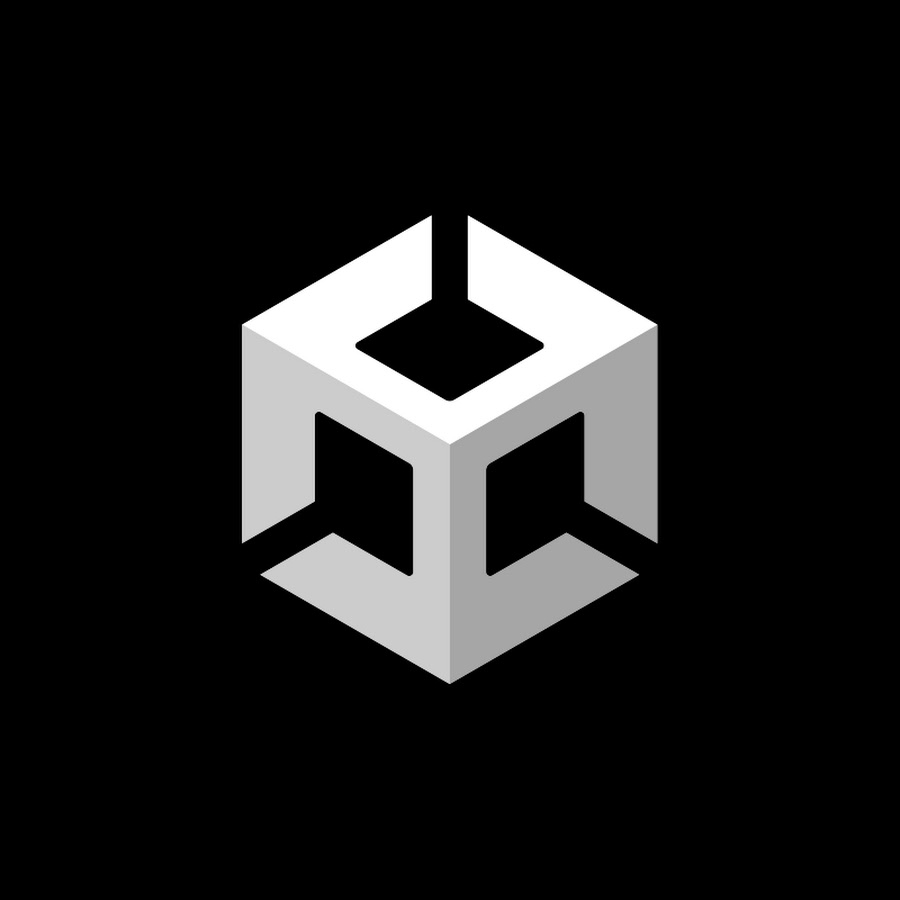
1. Понять основные особенности и преимущества работы в игровом движке.
2. Познакомится с базовой навигацией в движке, разобраться в особенностях работы с векторами, координатами и горячими клавишами.
3. Разобраться в таких понятиях как Scene, GameObject, Canvas.
4. Разобраться, какие Ide лучше всего подходят для работы с Unity.
5. Понять базовую теорию работы с Git.
6. Понять минимальные технические характеристики для ПК в работе с игровым движком.

**ЗАДАНИЕ:**

1. Установить игровой движок UNITY на свой ПК.
2. Установить Visual Studio(Ide) на свой ПК.
3. Создать пустой проект в Unity.
4. Создать подающие домино из примитивных объектов.

**ТЕОРИТИЧЕСКИЙ МАТЕРИАЛ**

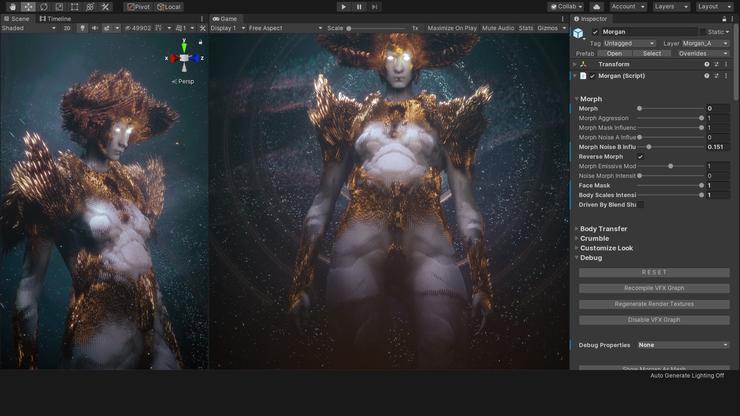
**Немного о Unity.**



Unity - это игровой движок, позволяющий создавать игры под большинство популярных платформ. С помощью данного движка разрабатываются игры, запускающийся на персональных компьютерах (работающих под Windows, MacOS, Linux), на смартфонах и планшетах (iOS, Android, Windows Phone), на игровых консолях (PS, Xbox, Wii) и даже есть поддержка HTML5.

Unity сразу идёт в комплекте со средой разработки. Это одновременно и редактор кода, и работа с графикой, и логика поведения предметов в игровой сцене, и всё остальное.

Особенность этого редактора в том, что всё это можно делать и настраивать прямо во время запуска или тестирования сюжета. Например, если вам не понравилось, как машина в игре реагирует на лужи, то, можно не выходя из сцены поправить нужный параметр и сразу увидеть, что изменилось.



Игровой движок отлично справляется с задачами вроде трёхмерного моделирования объектов из реального мира. После того, как вы соберёте модель и зададите нужные свойства, можно будет посмотреть, как она работает в разных условиях, в том числе и в движении.

То, как предметы в игре взаимодействуют между собой и с окружающей средой, называется физикой. Если писать правила такого взаимодействия с нуля для каждого объекта, на это может уйти очень много времени. Например, как описать физику резинового мяча по сравнению с физикой камня или металла? А физику автомобильного колеса? Зачастую в игровых движках (не только Unity) существуют готовые решения.

В Unity уже есть готовая физика и шаблоны для создания своих правил — можно сразу использовать их в игре. Движок уже знает, что, если на горку поставить коробку, она сползёт, перевернётся и остановится, а мяч в этой же ситуации должен просто покатиться дальше.



Одно из сильных мест Unity — скрипты на JS и C#. Вам не нужно изучать встроенный язык скриптования с его логикой и ограничениями — вы просто берёте любой из этих двух популярных языков программирования и пишете, что вам нужно.

Unity — это игровой движок, написанный на неуправляемом C++. Вы пишете код на C#, JavaScript (UnityScript) или Boo. Ваш код (не код движка Unity) выполняется в Mono или Microsoft .NET Framework и транслируется JIT-компилятором (кроме iOS, которая не поддерживает JIT-код, и для нее код компилируется Mono в неуправляемый код с использованием компиляции Ahead-of-Time [AOT]).

На скрипты можно повесить что угодно: появление врагов из-за угла, содержимое сундука с сокровищами или ответы второстепенного героя. Если вы умеете программировать, то сможете сделать настолько сложную и интересную игру, насколько захотите сами.

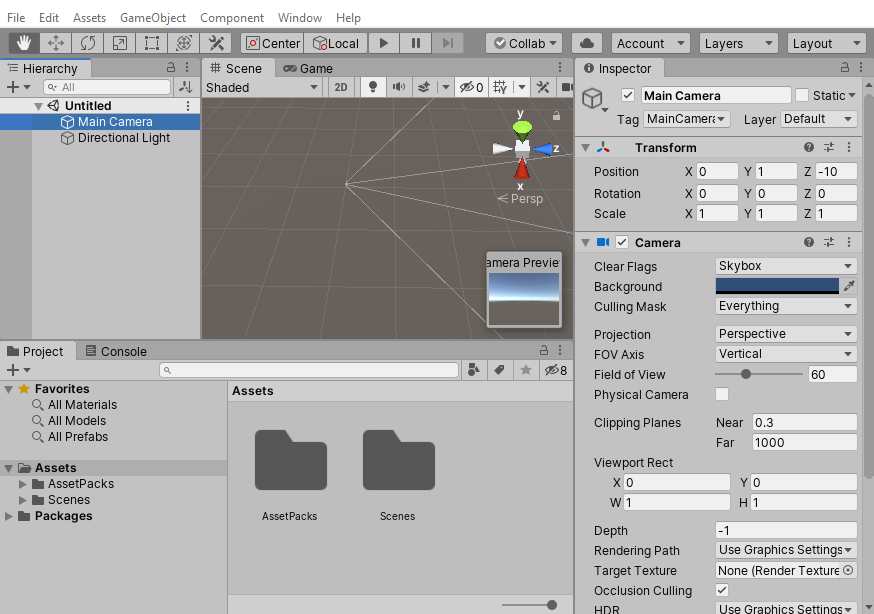
Все языки, с которыми работает Unity, являются объектно-ориентированными языками программирования. Как и любой язык, объектно-ориентированные языки программирования имеют синтаксис или части речи, а основные части называются переменными, функциями и классами.

Unity позволяет оперировать не только кодом, но и визуальными компонентами с последующим их экспортом на любую распространенную мобильную платформу, а также делать много всего другого, причем бесплатно. (Существует и профессиональная версия, но она не бесплатна. Впрочем, даже бесплатная версия позволяет делать впечатляющие вещи.) Unity поддерживает все основные 3D-приложения, множество форматов звука и даже понимает формат .psd (Photoshop), поэтому вы можете просто закинуть файл .psd в проект на Unity. Unity разрешает импортировать и собирать ресурсы, писать код, взаимодействующий с вашими объектами, создавать или импортировать анимации для использования с продвинутой системой анимации и многое другое.

**Среда разработки Unity.**

Базовым блоком при любой игры на Unity является «сцена», представляет, зачастую один уровень игры. Однако, бывает так, что одна сцена включает несколько уровней, или один объёмный уровень дробится на ряд сцен. Сцена, как матрёшка, состоит из объектов, наполненных компонентами. Именно компоненты делают интерфейс живым и привлекательным, обеспечивают такие функции как, например, визуализация, анимация, реалистичность. С помощью C# разработчик может сам написать их, сформировать нужную ему логику будущей игры. Перейдём непосредственно к движку.

После запуска редактора и создания нового файла, перед вами появится окно, состоящее из 4-х основных областей:

[](https://tproger.ru/s3/uploads/2021/09/1-3.png)

Слева вверху расположено окно Hierarchy, в котором отображается иерархия игровых объектов сцены: камеру (Main Camera), отображающую виртуальный мир глазами играющего, и источник света (Directional Light), определяющий направление освещённости сцены, без которого она выглядела бы как чёрное поле.

По центру расположено окно Scene, где, собственно, можно визуально редактировать уровень — вращать и передвигать с помощью мыши, получая наглядный результат. Справа от него находится вкладка Game (на скриншоте она неактивна). При переключении на неё кнопкой со значком воспроизведения сцена отображается через Main Camera.

В правой части экрана можно увидеть фрэйм Inspector — представляет собой набор полей с параметрами отмеченного объекта с возможностью их редактирования. На картинке видны 2 компонента: Transform, отвечающий за положение камеры, и, непосредственно, Camera — воплощает её функционал. Компонент Transform свойственен всем игровым объектам в Unity.

Внизу окна находится вкладка Project, содержащая цифровые объекты проекта (ассеты). По сути, это файл с данными, которые можно применить для создания сцен и интерфейса: текстуры, двумерные и трёхмерные элементы, звуковое сопровождение, анимация и конфигурации. Разработчики, не обладающие выдающимися навыками графического дизайна, могут загружать ассеты из Unity Asset Store в том числе и бесплатно. Также в Unity поддерживается опция загрузки файлов в стандартных форматах, например, png, jpg, fbx.

Справа от Project можете наблюдать неактивную вкладку Console. Здесь отображаются баги (ошибки), которые стоит периодически отслеживать, и сюда можно вводить части кода с целью отладки.

**Основные компоненты Unity.**

Компоненты (Components) определяют поведение объектов в игре. Они - функциональная часть каждого игрового объекта (GameObject). Каждый объект в вашей игре - это GameObject, от персонажей и коллекционных предметов до источников света, камер и спецэффектов. Однако GameObject ничего не может делать сам по себе; вам нужно присвоить ему свойства, прежде чем он сможет стать персонажем, средой или специальным эффектом.

Игровой объект является контейнером для различных компонентов. По умолчанию у всех игровых объектов есть компонент Transform. Потому что Transform диктует, где расположен игровой объект, и как он поворачивается и масштабируется. Без компонента Transform, игровой объект не будет иметь место в мире.

Canvas - это область, внутри которой размещаются все элементы пользовательского интерфейса. То есть, когда мы создаем новый игровой объект пользовательского интерфейса, то Сanvas для него создается автоматически. Компонент Canvas представляет абстрактное пространство, в котором размещается и отображается пользовательский интерфейс. Все элементы пользовательского интерфейса должны быть дочерними по отношению к GameObject, к которому прикреплен компонент Canvas.

**Горячие клавиши Unity.**

Горячие клавиши (от англ. hot keys) — сочетание клавиш, которое служит для ускорения выполнения тех или иных действий в среде операционной системы вообще и программ в частности.

|  |
| --- |
| **Tools** (**инструменты**) |
| *“Горячая” клавиша* | *Действие* |
| Q | Pan (перемещение камеры сцены) |
| W | Move (перемещение) |
| E | Rotate (вращение) |
| R | Scale (масштабирование) |
| T | Rect Tool |
| Z | Pivot Mode toggle (переключение Pivot-режима) |
| X | Pivot Rotation Toggle (переключение поворота пивота) |
| V | Vertex Snap (привязка вершин) |
| CTRL/CMD+ЛКМ | Snap (привязка) |

|  |
| --- |
| **GameObject** |
| CTRL/CMD+SHIFT+N | Создать новый игровой объект |
| CTRL/CMD+ALT+F | Переместить объект к точке вида |
| CTRL/CMD+SHIFT+F | Выровнить по виду |
| SHIFT+F или двойное F | Привязывает камеру окна Scene View к выбранному игровому объекту |

Более подробно - <https://sites.google.com/site/rusewyl/home/socetania-klavis-unity>

**Популярные IDE для Unity.**

Microsoft Visual Studio - полнофункциональная интегрированная среда разработки (IDE) с поддержкой популярных языков программирования, среди которых С, C++, VB.NET, C#, F#, JavaScript, Python.

Функциональность Visual Studio охватывает все этапы разработки программного обеспечения, предоставляя современные инструменты для написания кода, проектирования графических интерфейсов, сборки, отладки и тестирования приложений. Возможности Visual Studio могут быть дополнены путем подключения необходимых расширений.

Редактор кода Visual Studio поддерживает подсветку синтаксиса, вставку фрагментов кода, отображение структуры и связанных функций. Существенно ускорить работу помогает технология IntelliSense - автозавершение кода по мере ввода.

В Unity мире Visual Studio используется только с расширением ReSharper. ReSharper - дополнение (плагин), разработанное компанией JetBrains для повышения продуктивности работы в Microsoft Visual Studio. Он ускоряет разработку приложений/игр, позволяет быстрее писать код за счет продвинутых горячих клавиш, предоставляет более глубокий рефакторинг кода, упрощает тестирование.

JetBrains Rider - кроссплатформенная интегрированная среда разработки программного обеспечения для платформы .NET, разрабатываемая компанией JetBrains. Поддерживает языки программирования C#, VB.NET и F#.

Rider включает большинство функций, популярного расширения Visual Studio для разработчиков .NET – ReSharper. В составе Rider есть внушительный набор для рефакторинга, проверки кода и контекстных действий для всех поддерживаемых им языков и технологий. У Visual Studio также есть наборы рефакторингов и проверки ошибок кода, но гораздо более ограниченный, чем те что предоставлены в Rider и ReSharper.

Решения и проекты, с которыми работает JetBrains Rider, полностью совместимы с Visual Studio, и не используют проприетарные форматы.

**Технические характеристики ПК для работы в Unity.**

Возьмем самые минимальные характеристики для комфортной работы и создания простеньких игр.

**Операционная система:** Windows 7, и выше.

Процессор: i5-2500к (4 ядра, 4 потока).

Накопитель: любой HDD, но желательно SSD.

Видеокарта: GT 730 c поддержкой DirectX 12.

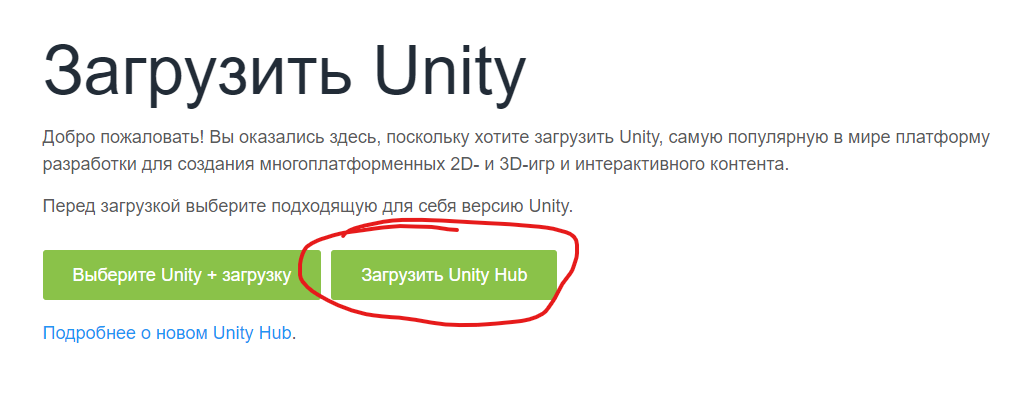
Оперативная память: 8 гигабайт (прям минимум).

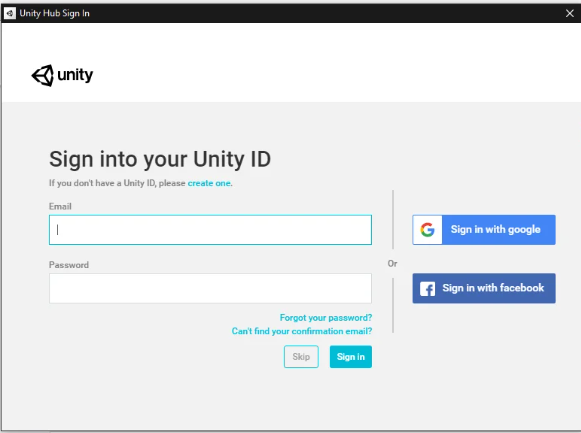
**Работа с Git в Unity.**

Git — это набор консольных утилит, которые отслеживают и фиксируют изменения в файлах (чаще всего речь идет об исходном коде программ, но вы можете использовать его для любых файлов на ваш вкус). С его помощью вы можете откатиться на более старую версию вашего проекта, сравнивать, анализировать, сливать изменения и многое другое. Этот процесс называется контролем версий. А ещё это must-have инструмент для взаимодействия нескольких разработчиков на одном проекте.

Работа с Git в Unity не чем не отличается от работы с web проектами или любыми другими. У Unity есть собственная поддержка Git для работы в командах.

**ЛАБОРАТОРНАЯ РАБОТА**

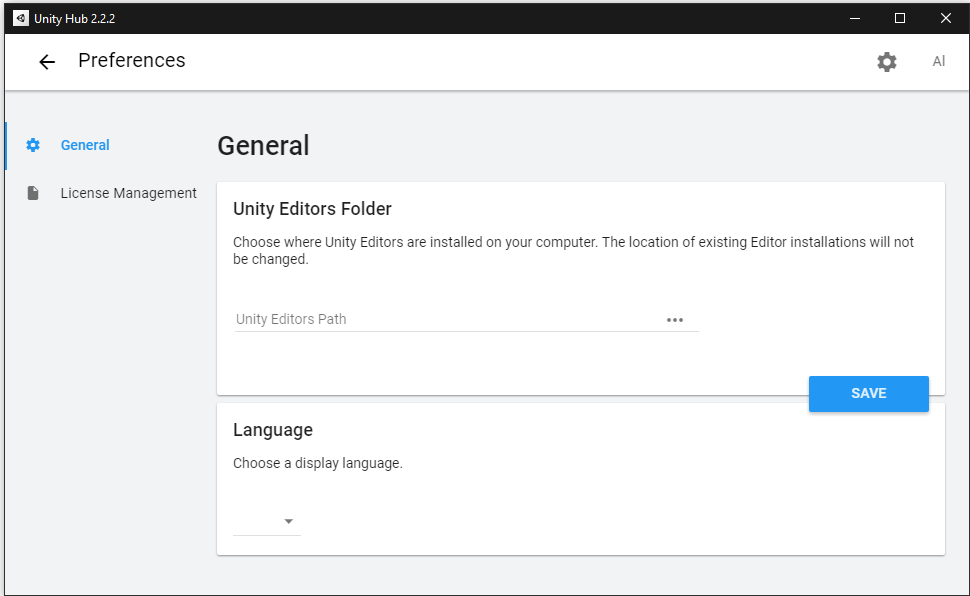
1. Необходимо скачать Unity Hub по данной [ссылке](https://unity3d.com/ru/get-unity/download). После скачивания установите в любое удобное для вас место.
2. Для установки и использования самого Unity вы должны иметь зарегистрированный аккаунт на [сайте юнитеков](https://id.unity.com/en/conversations/4f406259-be6c-4e1f-b103-0dd3526135b201af). Если таковой есть, то при первом запуске Unity Hub необходимо войти в свой аккаунт, выбрать тип лицензии и перейти к установке редактора Unity. Если же у вас еще нет аккаунта, вы можете создать его. Можно как сделать новый аккаунт на самом сайте Unity и войти, используя данные созданной учетной записи, так и воспользоваться входом с помощью одной из популярных социальных сетей.



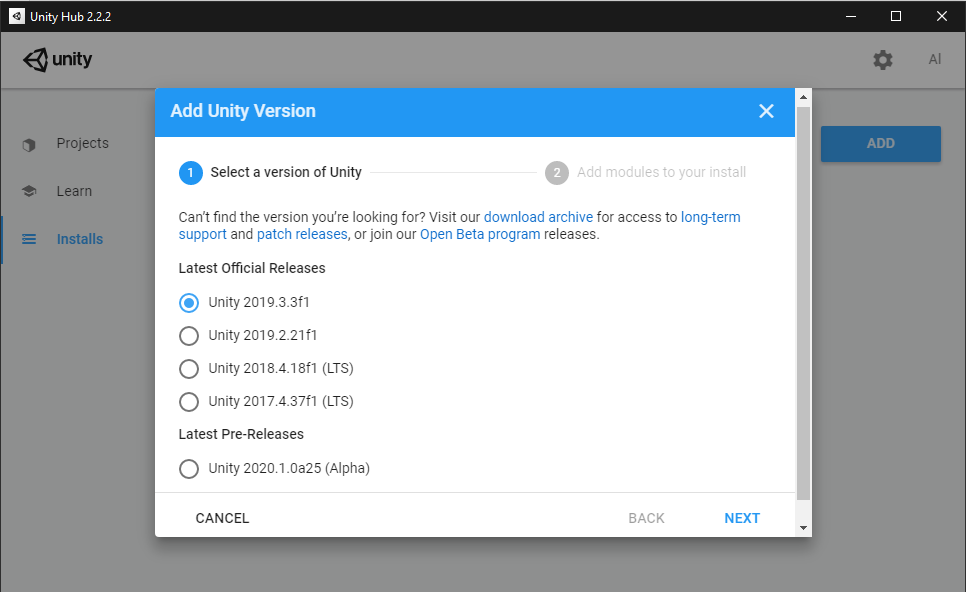
После проделывания всех вышеописанных манипуляций, можно установить сам редактор Unity. Для этого в Unity Hub в левом меню выбираем вкладку Installs (необходимо помнить, что по умолчанию папкой для установки всех версий редактора под Windows будет *C:\Program Files\Unity\Hub\Editor*, а под Mac */Applications/Unity/Hub/Editor*. Поэтому если у вас мало места на основном диске, и вы хотите поменять папку для установки редактора, необходимо сделать следующие шаги:

2.1. В верхнем правом углу окна Hub’a нажмите на значок шестерни

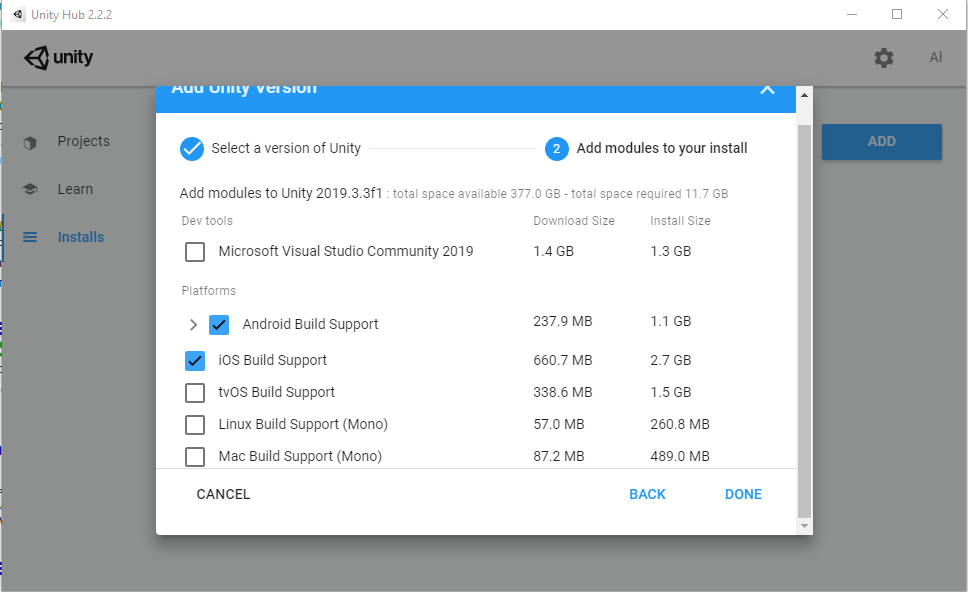
2.2. В окне выбора папки расположения версий редактора выберите другую папку и нажмите Save.



1. После того как вы определились с выбором места для хранения версий редактора, вернитесь к вкладке Installs и нажмите кнопку Add, после чего откроется окно выбора версий Unity редактора для установки (я рекомендую устанавливать официальный релиз последней версии, он при открытии данного окна выбран по умолчанию).



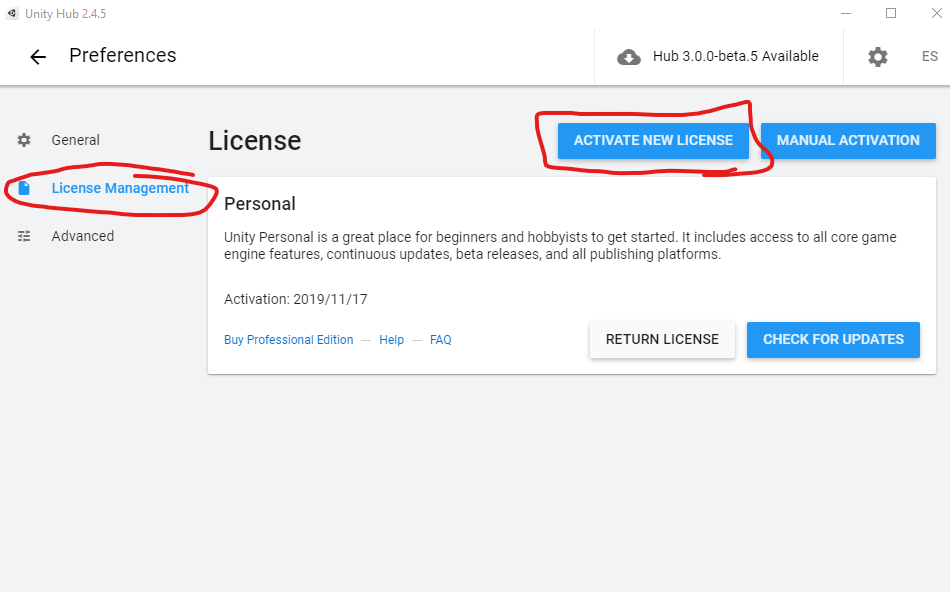
Затем нажимаем Next, в следующем окне нам предлагается выбрать компоненты редактора (так как он модульный), а также установить Microsoft visual studio 2019 если у нас его нет на компьютере. Ставим галочки напротив Android Build Support и IOS Build Support. После всех проделанных манипуляций жмем кнопку Done и ожидаем скачивания и установки редактора. (Если вы до этого уже устанавливали Visual Studio и не устанавливали в него расширение Unity, то следует это выполнить).

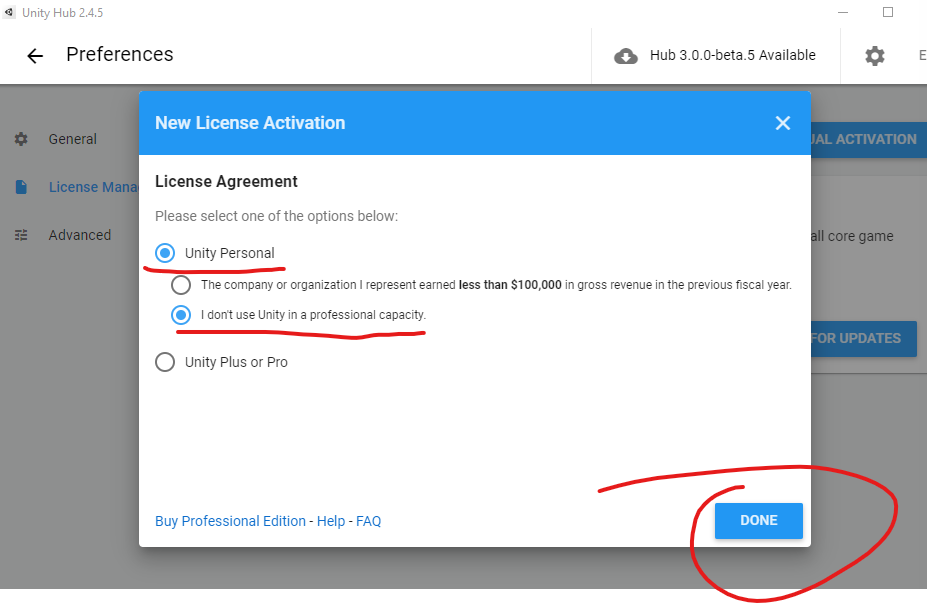


1. После установки убедитесь, что версия Unity установлена, для этого в Unity Hub выберите вкладку Installs.
2. Также стоит активировать бесплатную лицензию, для этого в Unity Hub нажмите на шестерёнку.

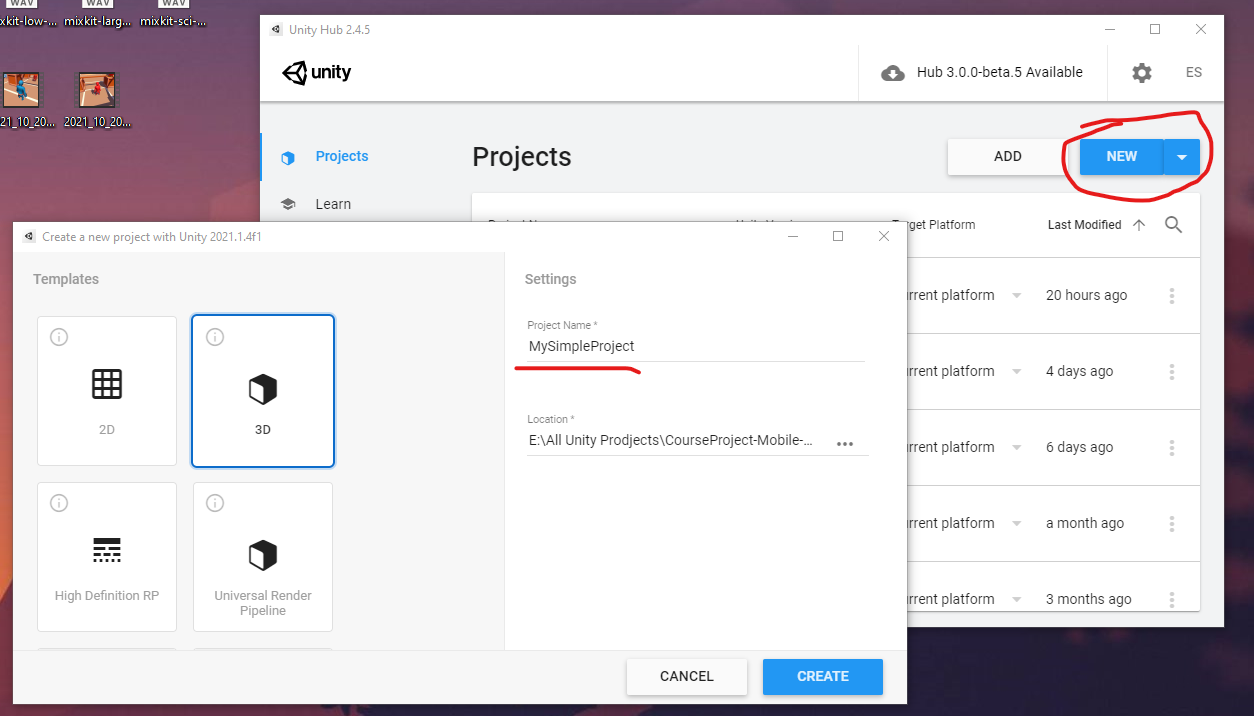


Затем перейдите на вторую вкладку License Management и нажмите на кнопку activate new license, в сплывающем окне выберите Unity Personal и I don’t use Unity in a professional capacity, затем нажмите на кнопку done.

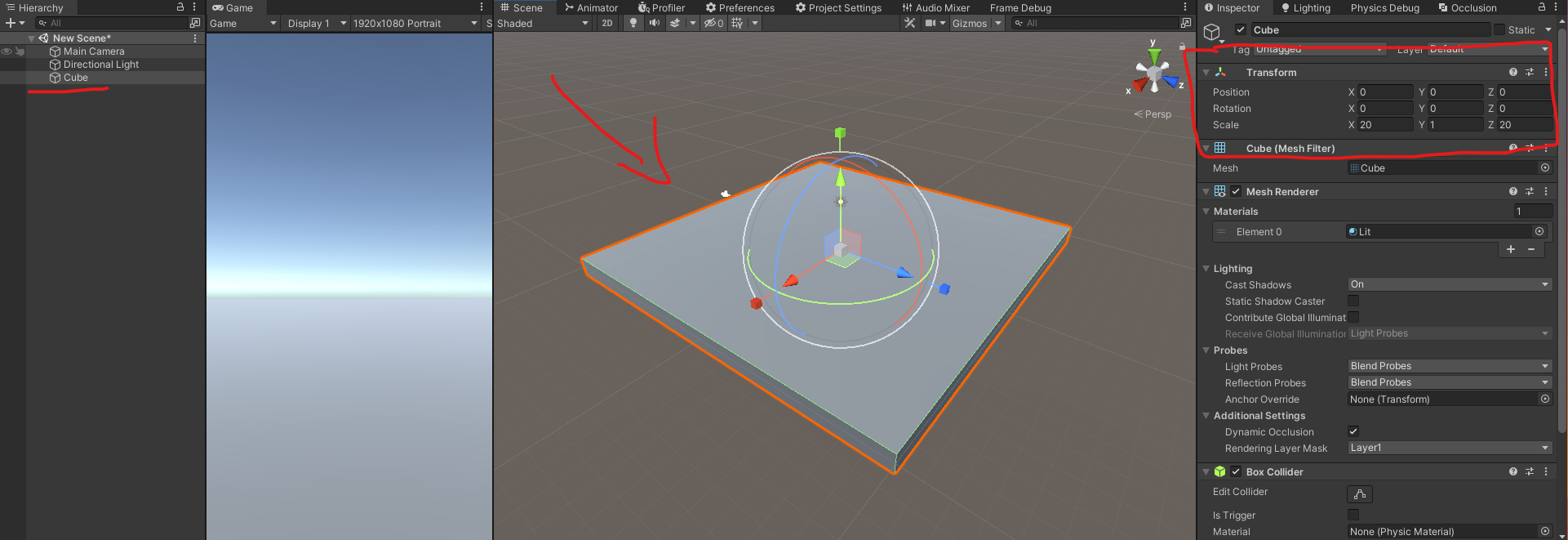




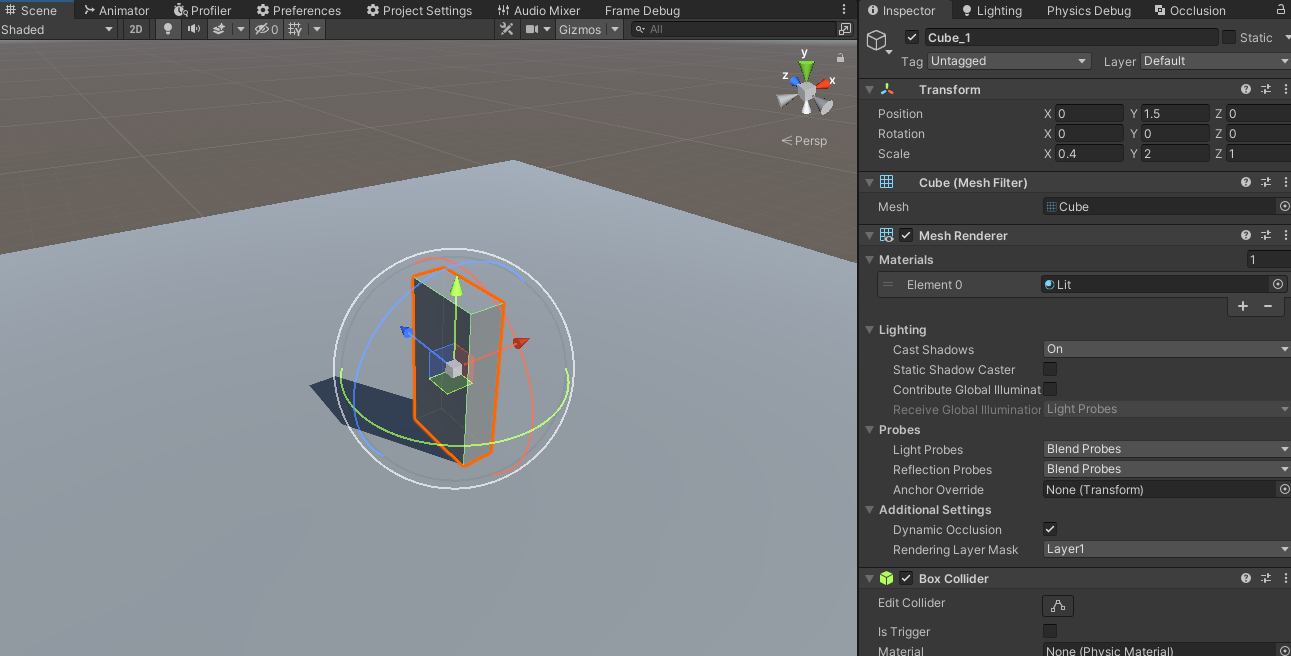
1. Создаем новый проект Unity нажав на стартовой странице Unity Hub кнопку New, затем в открывшимся окне вводим имя проекта MySimpleProject, и нажимаем Create.



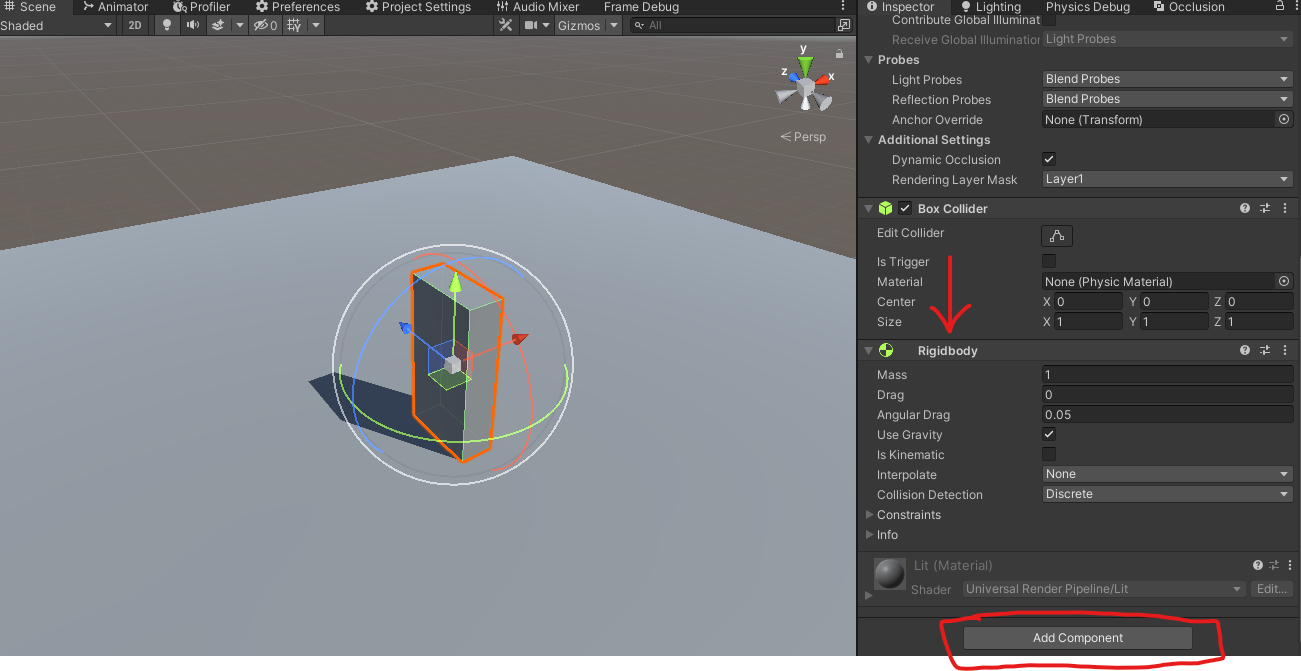
1. В открывшейся сцене добовляем «пол» в виде большого бокса. Должно получиться что-то вроде этого:



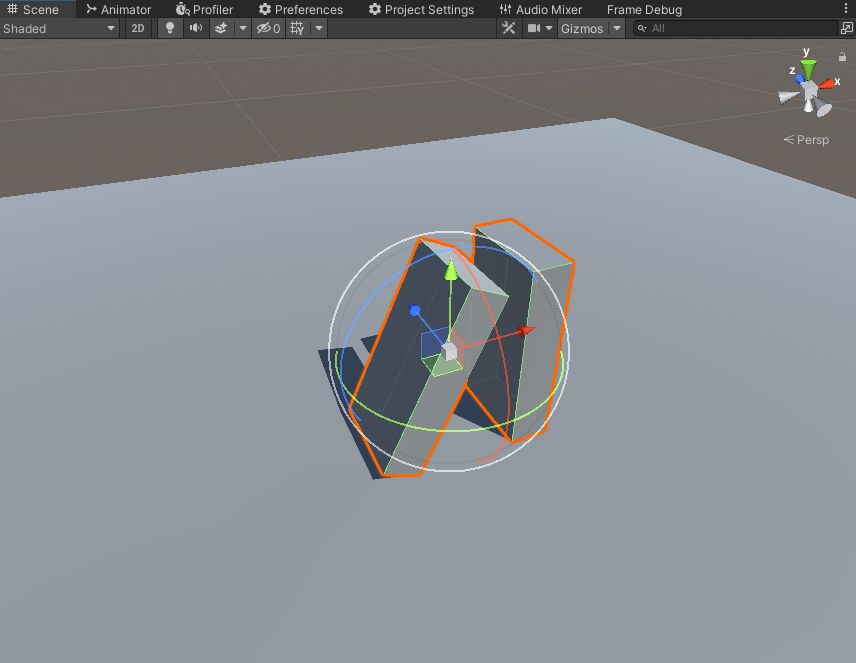
1. Далее создаем «Доминошку», также добавляем на сцену бокс и создаем из него нужную нам форму изменяя параметры Scale в компоненте Transform у нашего бокса. Должно получиться что-то вроде этого:



1. Теперь добавляем на нашу доминошку компонент Rigidbody, который в свою очередь отвечает за физику.



1. Далее копируем нашу доминошку кликнув на обьект в вкладе Hierarchy, выбрав доминошку и нажать сочетание клавишь ctrl + D. Новую доминошку следует наклонить под 45 градусов в сторону первой.



1. Теперь можно запустить нашу импровизированную игру и увидеть, как одна доминошка сталкивается с другой, и они обе падают. Вашей задачей будет создание некоего пути, состоящего из 20 доминошек. Не забывайте импровизировать, вы также можете сделать целый замок из домино, главное, чтобы он красиво разрушался) Пример работы:

