Муниципальное бюджетное общеобразовательное учреждение "Средняя школа №2 г. Суздаля"

**Индивидуальный итоговый проект п теме**

**“Создание игр на Unity”**

Работу выполнила:

Касаткина Виктория Владимировна

Ученица 9 «Б» класса

Руководитель проекта:

Логинова Наталья Александровна

г. Суздаль

2022

**Оглавление**

1. Введение..............................................................................................2

1.1. Актуальность...........................................................................2

1.2. Цель..........................................................................................2

1.3. Задачи.......................................................................................2

1.4. Выбор темы.............................................................................2

1. Теоретическое обоснование..............................................................3

2.1. Что нужно для создания игры...............................................3

2.2. Установка Unity......................................................................3

2.3. Создание проекта....................................................................7

2.4. Описание среды Unity..........................................................10

2.5. Возможности Unity...............................................................13

1. Практическая часть..........................................................................15

3.1. Описание игры......................................................................15

3.2. Создание персонажа.............................................................15

3.3. Создание камеры на игрока.................................................16

3.4. Создание врага......................................................................16

3.5. Основные блоки кода...........................................................17

3.6. Сбор игры под Android.........................................................18

1. Заключение.......................................................................................19

4.1. Перспектива...........................................................................19

4.2. Источники..............................................................................20

4.3. Приложение………………………………………………...21

**Введение**

**Актуальность**

В современном мире игры неотъемлемая часть жизни.

В настоящее время трудно представить жизнь без современных технологий и, в частности, компьютерных игр. Игры – это место, где человек может расслабится, посмеяться, новые знакомства, отвлечься от проблем в жизни. Люди не только получают удовольствия, многие геймеры зарабатывают посредством компьютерные игр.

Известная в мире компьютерных игр компания Rockstar Games, Inc за один только день заработала 800 миллионов долларов, а за всё время - 7 миллиардов долларов. За 24 года компания выпустила 58 игр и 25 игр стали успешными. В 2013 году вышла игра Grand Theft Auto V или сокращённо GTA V. Игровая индустрия – прибыльный бизнес.

**Цель**

Цель - показать, что геймеры могут не только играть, но и создавать игры с 2D-графикой.

**Задачи**

1.Изучить интерфейс и функционал Unity.

2.Создать 2D 3-х уровневой платформер в среде Unity.

3.Скомпилировать код для использования в мобильной системе на базе Android.

**Выбор темы**

Выбор темы обусловлен интересов к игровой индустрии и возможностям быть успешным в IT сфере

**Теоретическое обоснование**

**Что нужно для создания игры**

Для создания игры нужно:

- Игровой движок

- Знания языка программирования

- Концепция игры

Присутствует больше 10 игровых движков для создания игр.

Самые популярные: GameMaker Studio2, Unity 3D, CryEngine,

Cocos Creator, Corona (Solar2D), Godot, libGDX, Ren’Py, Construct 3, Unreal Engine 5. Мы с вами будем использовать Unity

Язык программирования. В каждом игровом движке используется разный язык программирования. В нашем случае этот это C#

Мы определились с игровым движком и языком программирования. Теперь определимся с игрой.

Присутствует множество разных жанров игр – квесты, экшн, ролевые, стратегии, симуляторы и т.д.

Есть также вид игры. 2D или 3D.

После того как мы определились с жанром и видом игры, нам нужна графика.

Существует двухмерная и трёхмерная графика.

2D-графика бывает двух видов – векторная и растровая.

**Растровая** - состоит из пикселей, описывающих цвет каждой точки изображения на экране.

**Векторная** – описывает способ рисования картинки из примитивов (круг, прямоугольник, линия, кривая).

В трёхмерной графике есть понятие освещения.

Источники света могут быть:

- глобальными

- направленными

- радиальными

- узконаправленными

**Установка Unity**

Для начала давайте познакомимся с Unity

Unity – Межплатформенная среда разработки компьютерных игр, разработанная американской компанией Unity Technologies. Unity позволяет создавать приложения, работающие на более чем 25 различных платформах, включающих персональные компьютеры, игровые консоли, мобильные устройства, интернет-приложения и другие. Выпуск Unity состоялся в 2005 году и с того времени идёт постоянное развитие.

Чтобы начать работать в Unity надо скачать программу.

1. Заходим на официальный сайт Unity[[1]](#footnote-1)

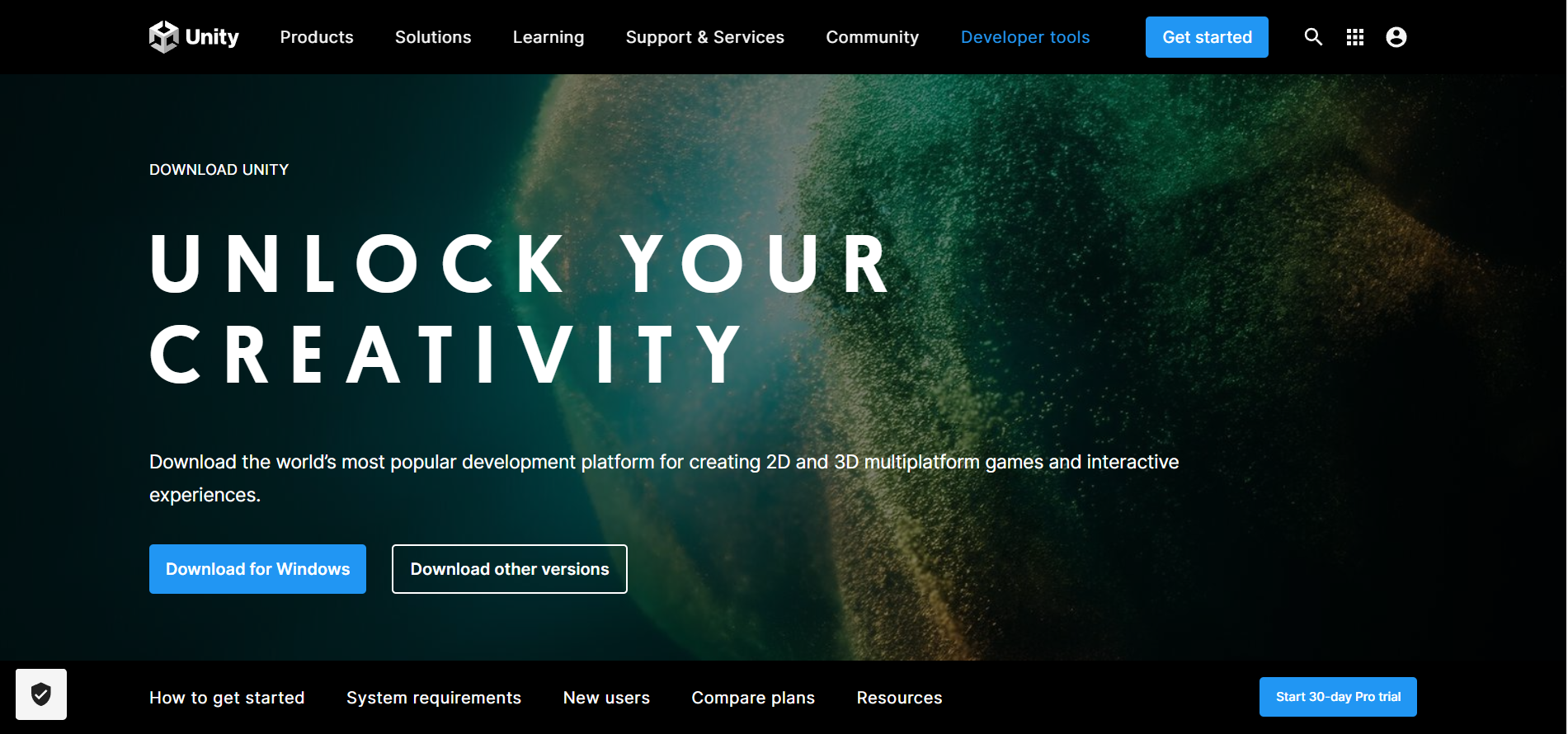


Рисунок Скрин из браузера открытого сайта Unity

1. Нажимаем на кнопку “Download for Windows”

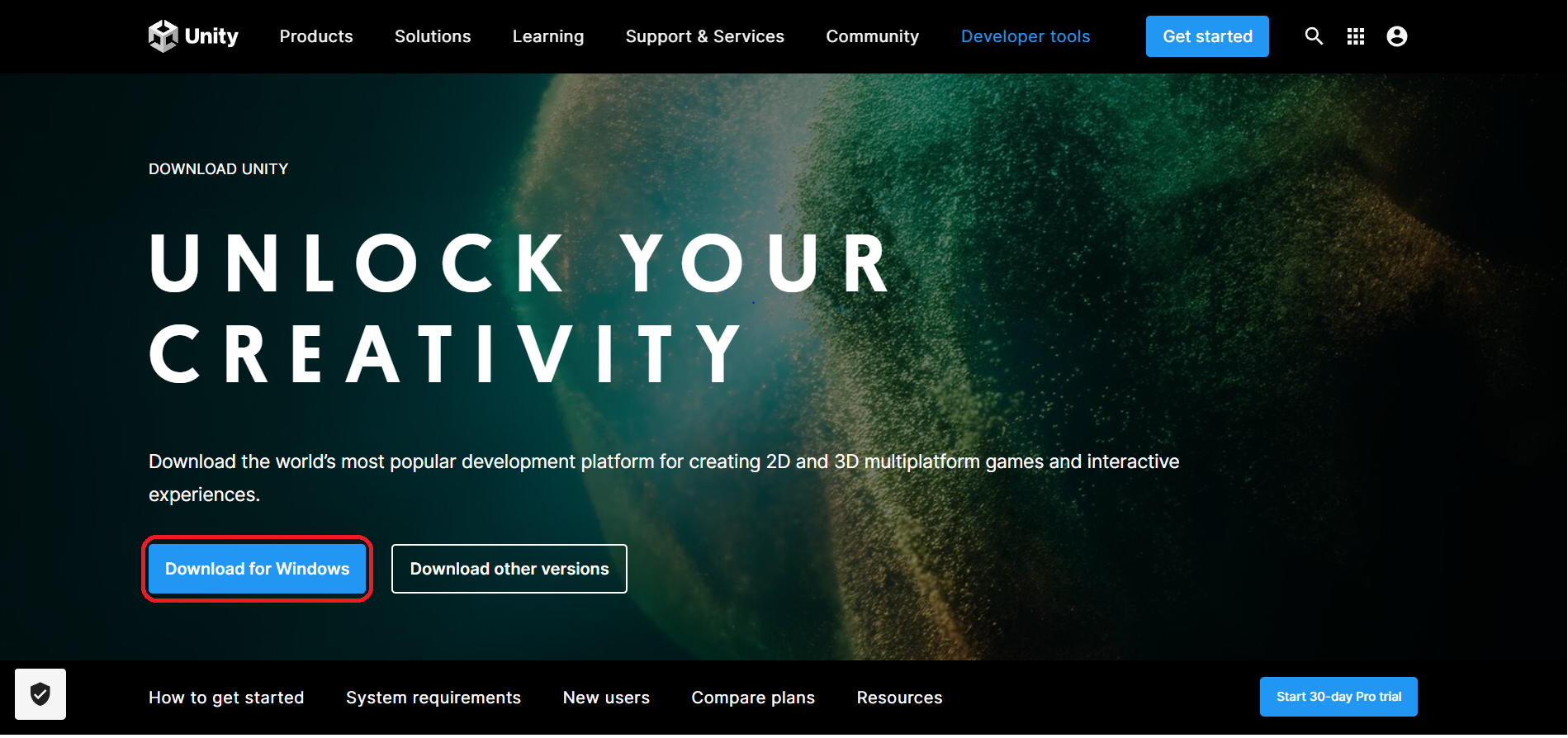


Рисунок Скрин из браузера открытого сайта Unity выбор установки

Таким образом мы скачаем Unity Hub

Unity Hub – это приложение, разработанное для упрощения рабочего процесса.

3.После установки запускаем Unity Hub

4. Чтобы начать что-то делать вам надо зарегистрироваться

В правом верхнем углу нарисован кружочек, это и есть регистрация.

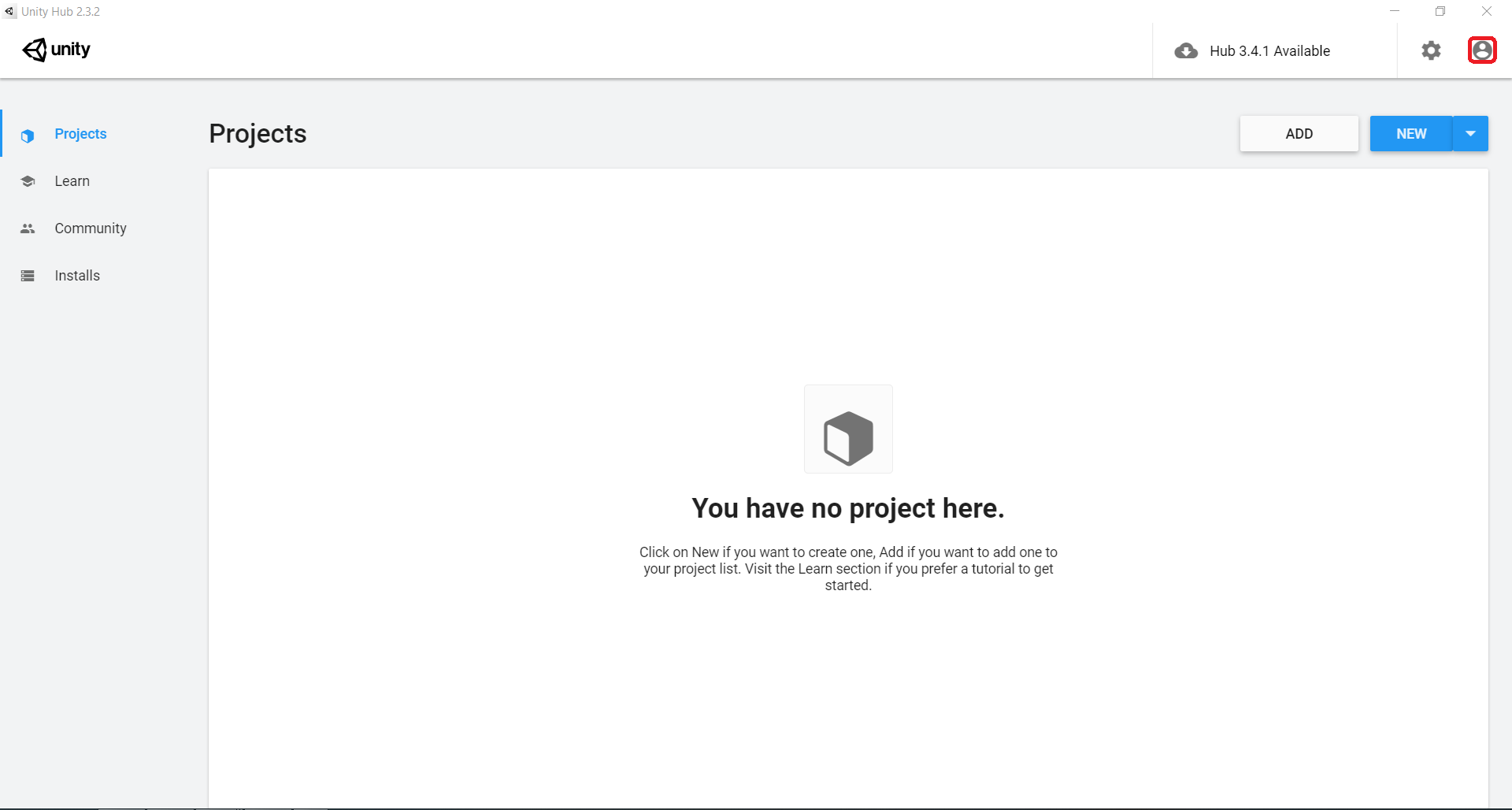


Рисунок 3 Меню проектов

5. После того как мы зарегистрировались нам надо вернутся в Unity Hub

6. Слева присутствуют кнопки, нам нужна Installs, нажимаем.

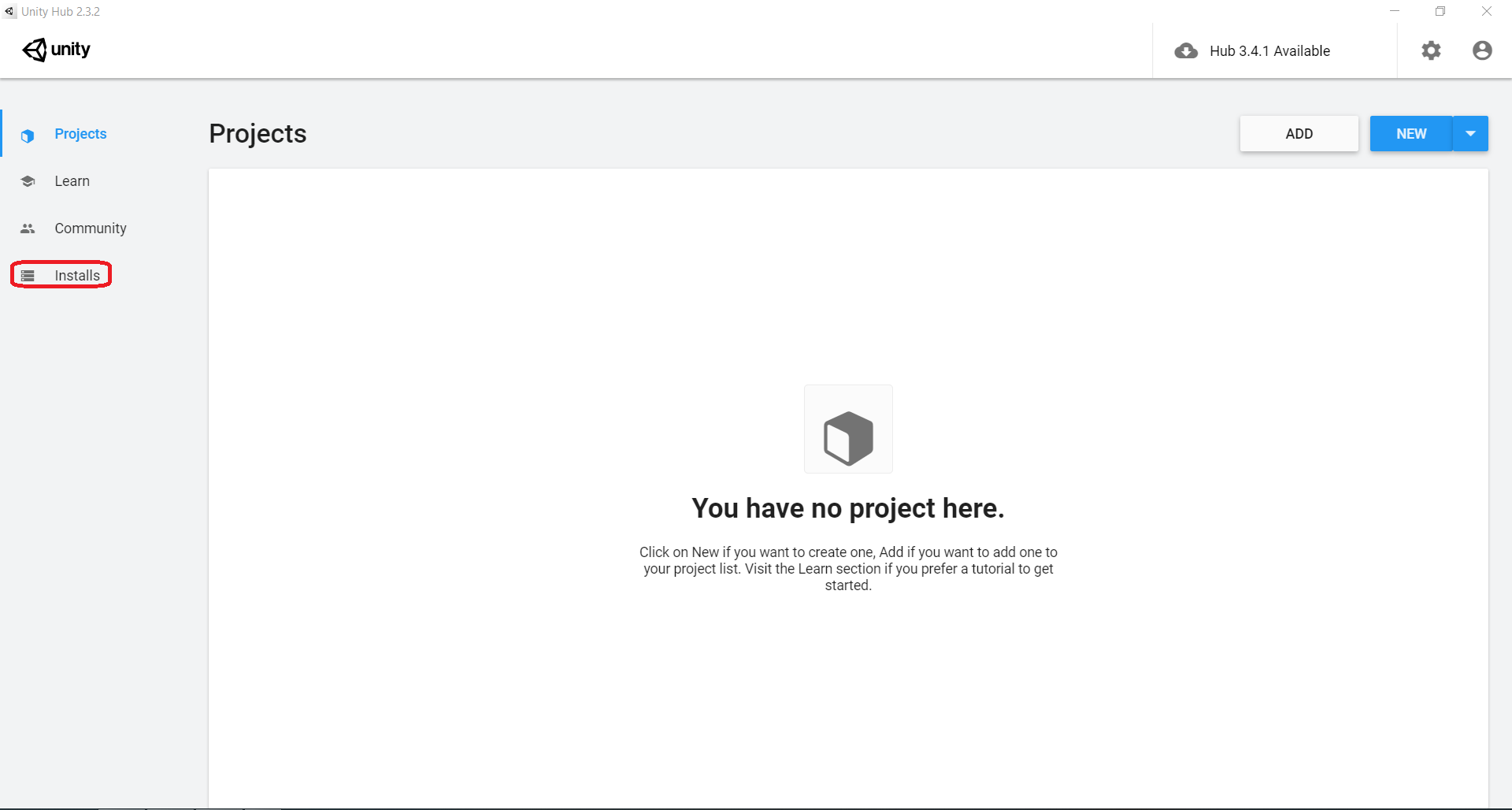


Рисунок 4 Кнопка установки Unity

7. Теперь нажимаем на кнопку ADD

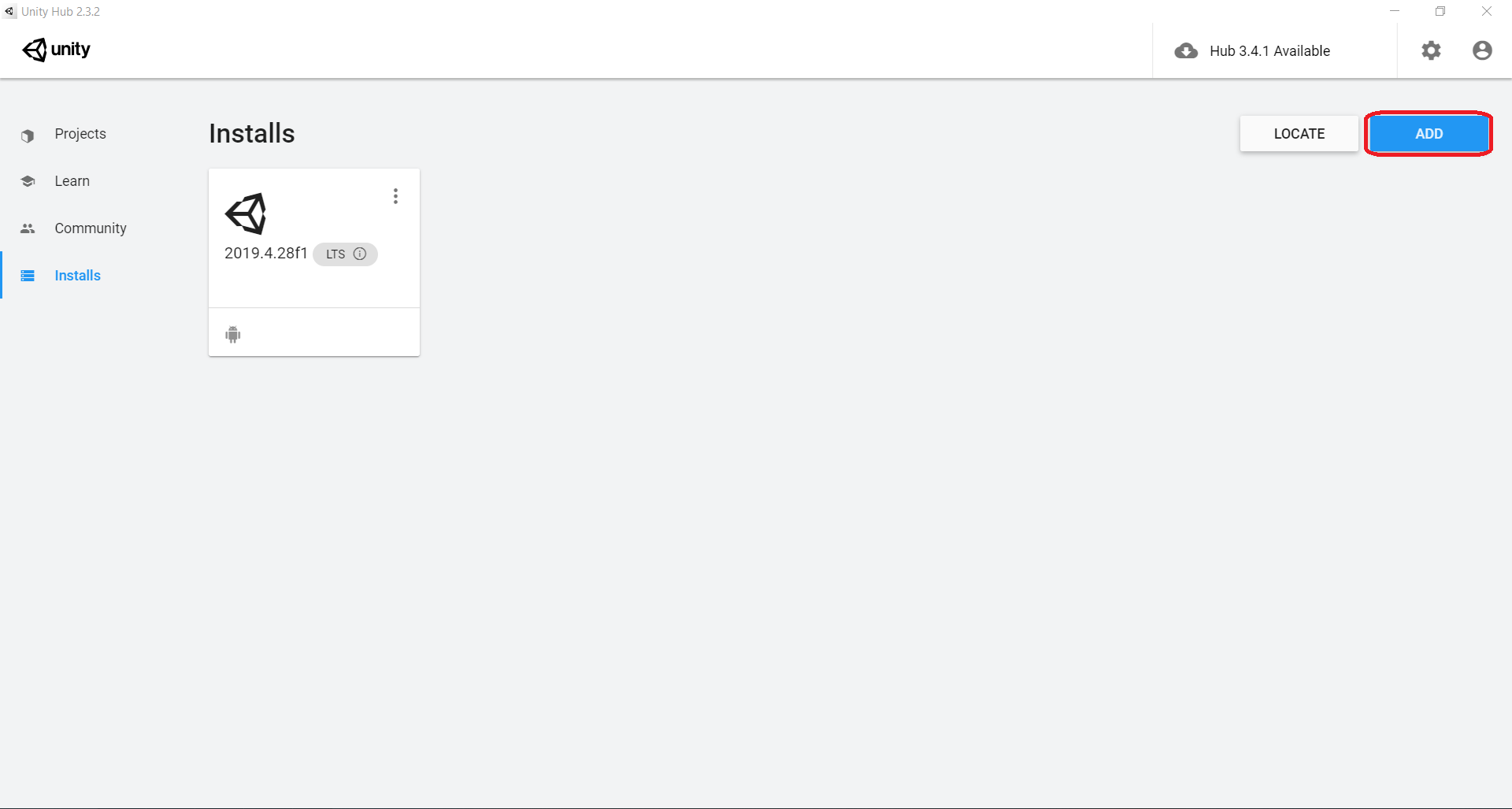


Рисунок Меню версий

8. В открывшимся меню выбираем версию Unity и

устанавливаем на ваш жёсткий диск. У меня установлена 2019.4.28f1.

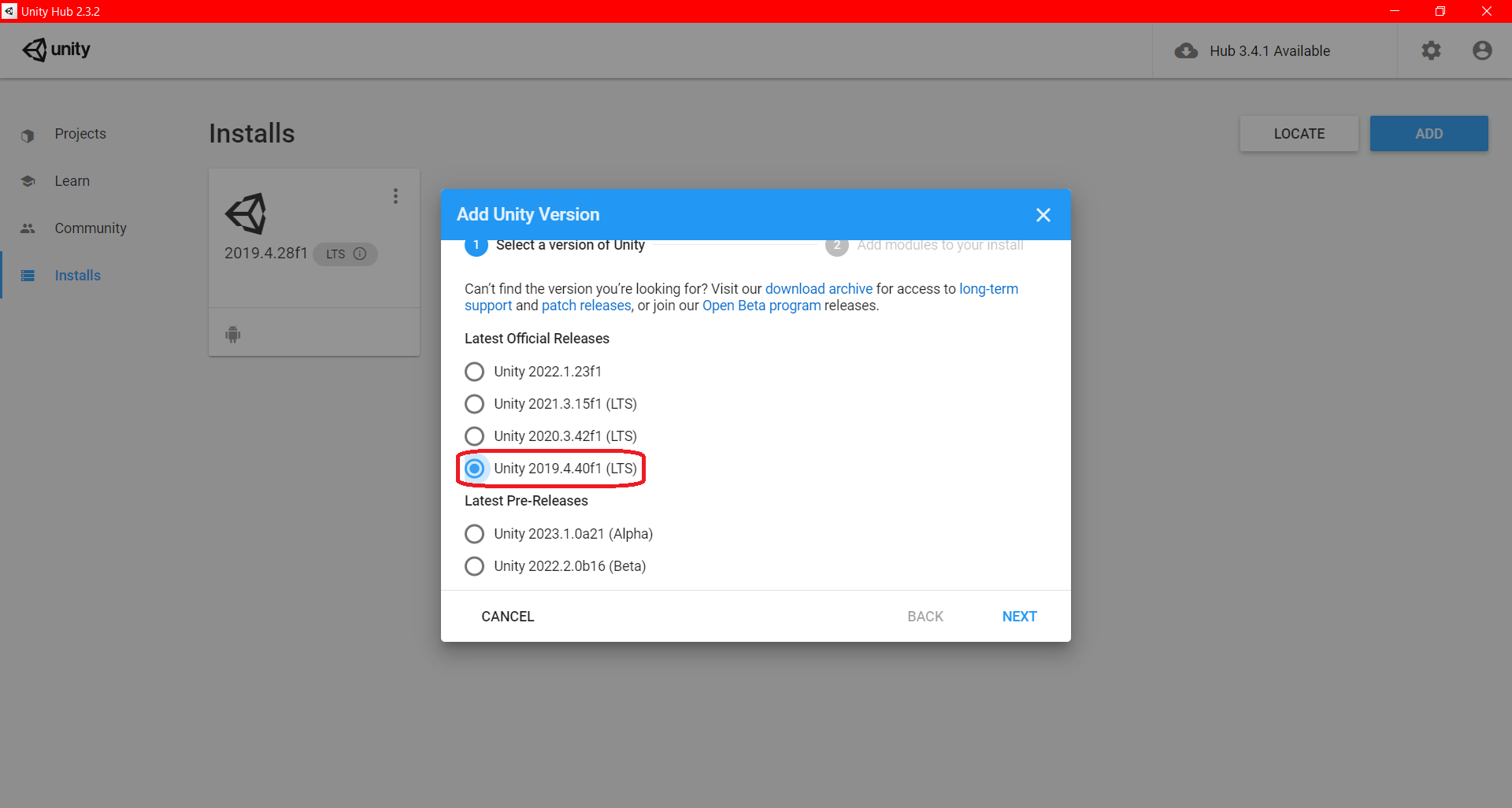


Рисунок Выбор версии Unity

Мы установили Unity, теперь мы можем начать создавать игру.

**Этапы создание проекта**

1. В Unity hub открываем меню Projects.

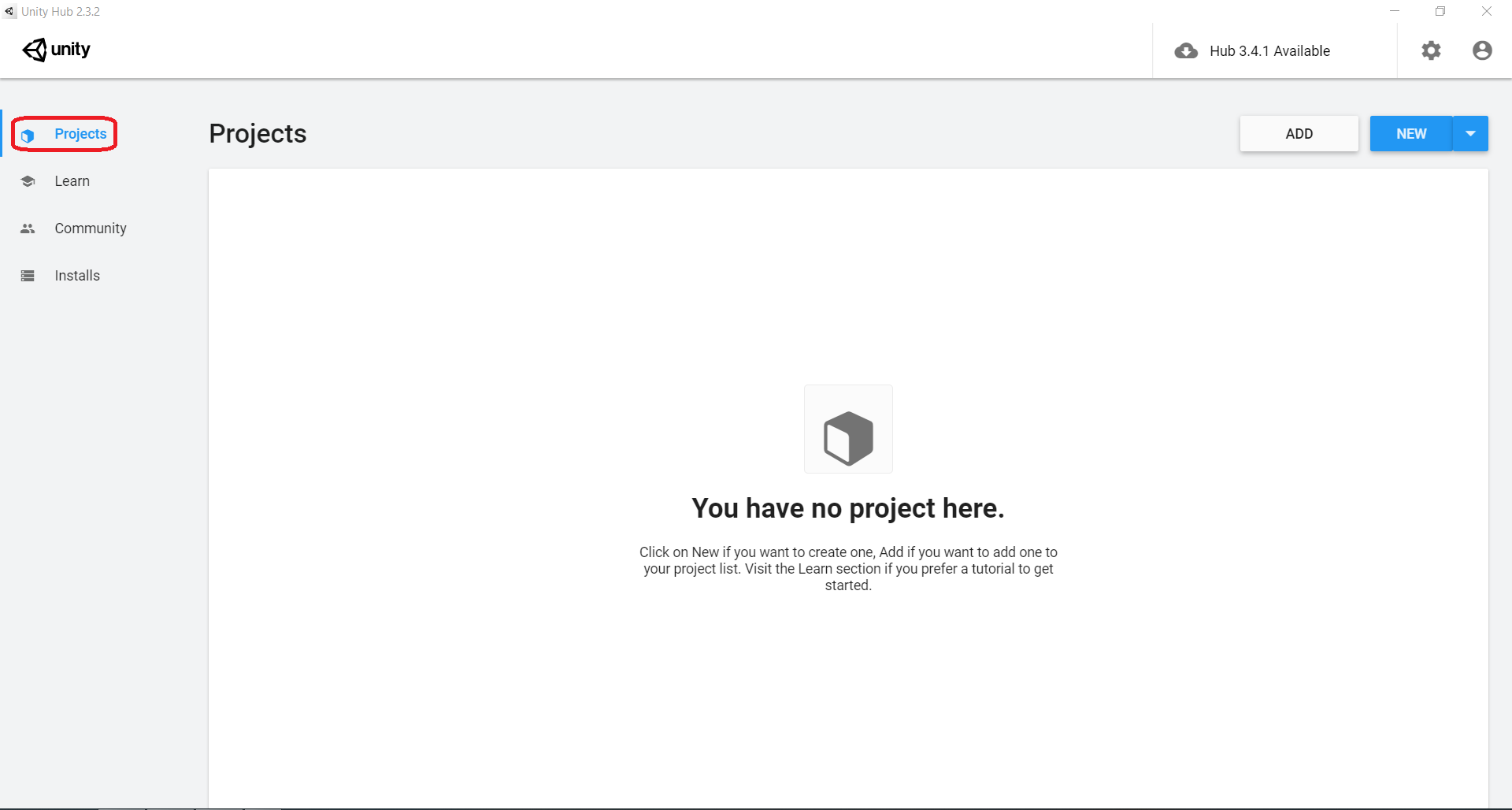


Рисунок 7 Меню создания проекта

2. Далее нажимаем кнопку NEW

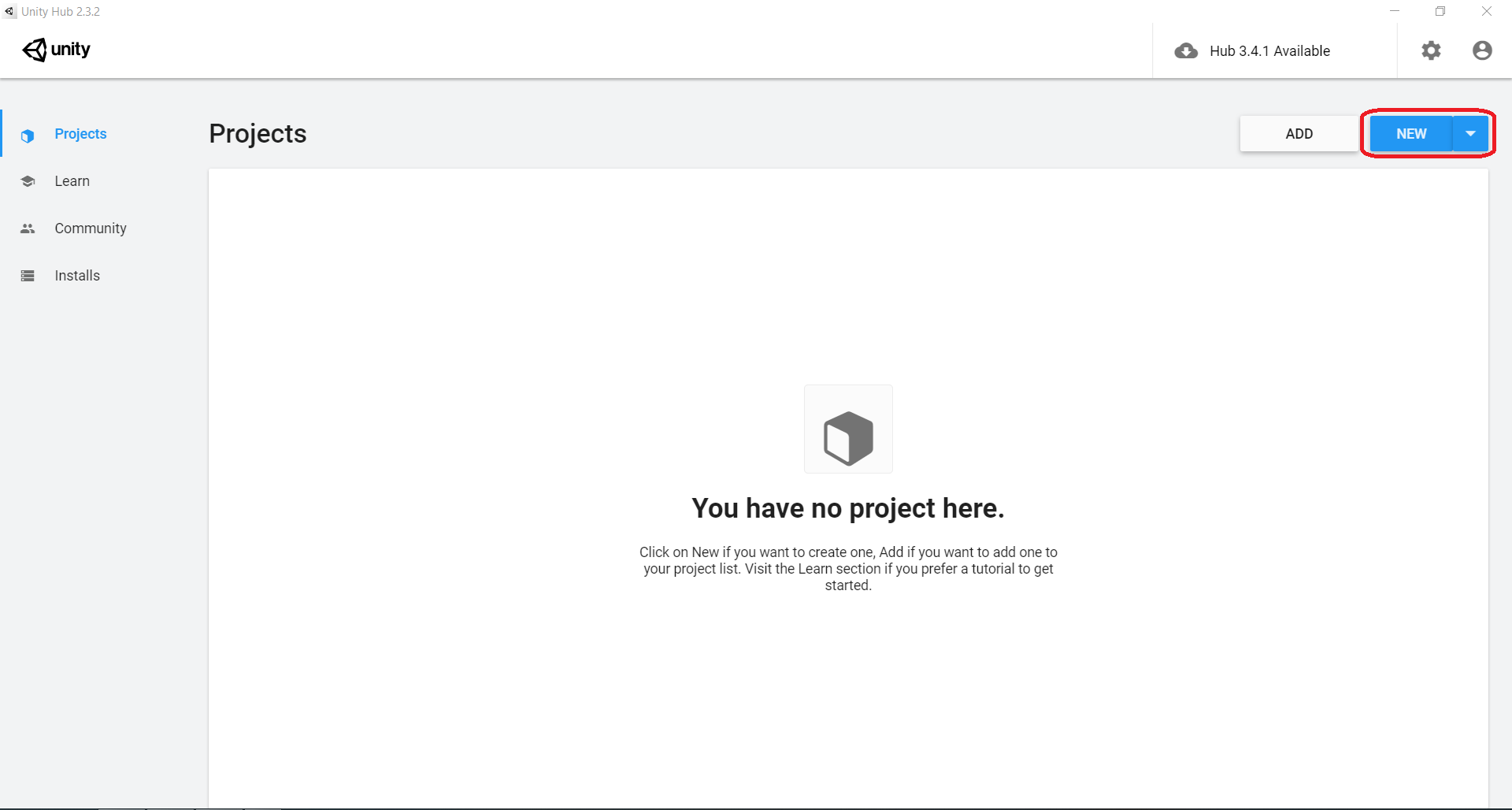


Рисунок 8 Кнопка создания проекта

3. В открывшимся окошке нам надо выбрать

- Стиль игры: 2D или 3D

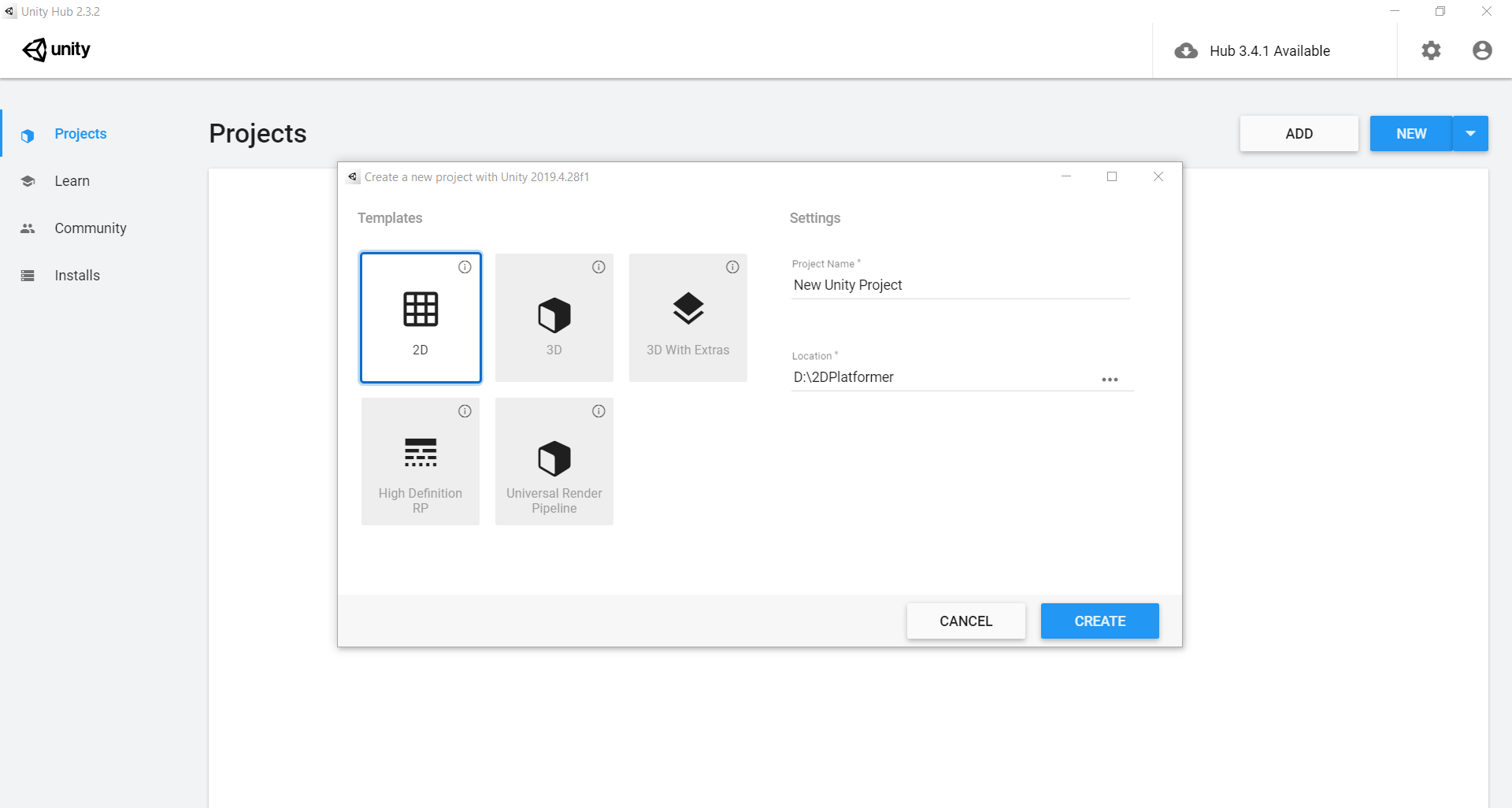


Рисунок 9 Выбор стиля игры

- Название проекта

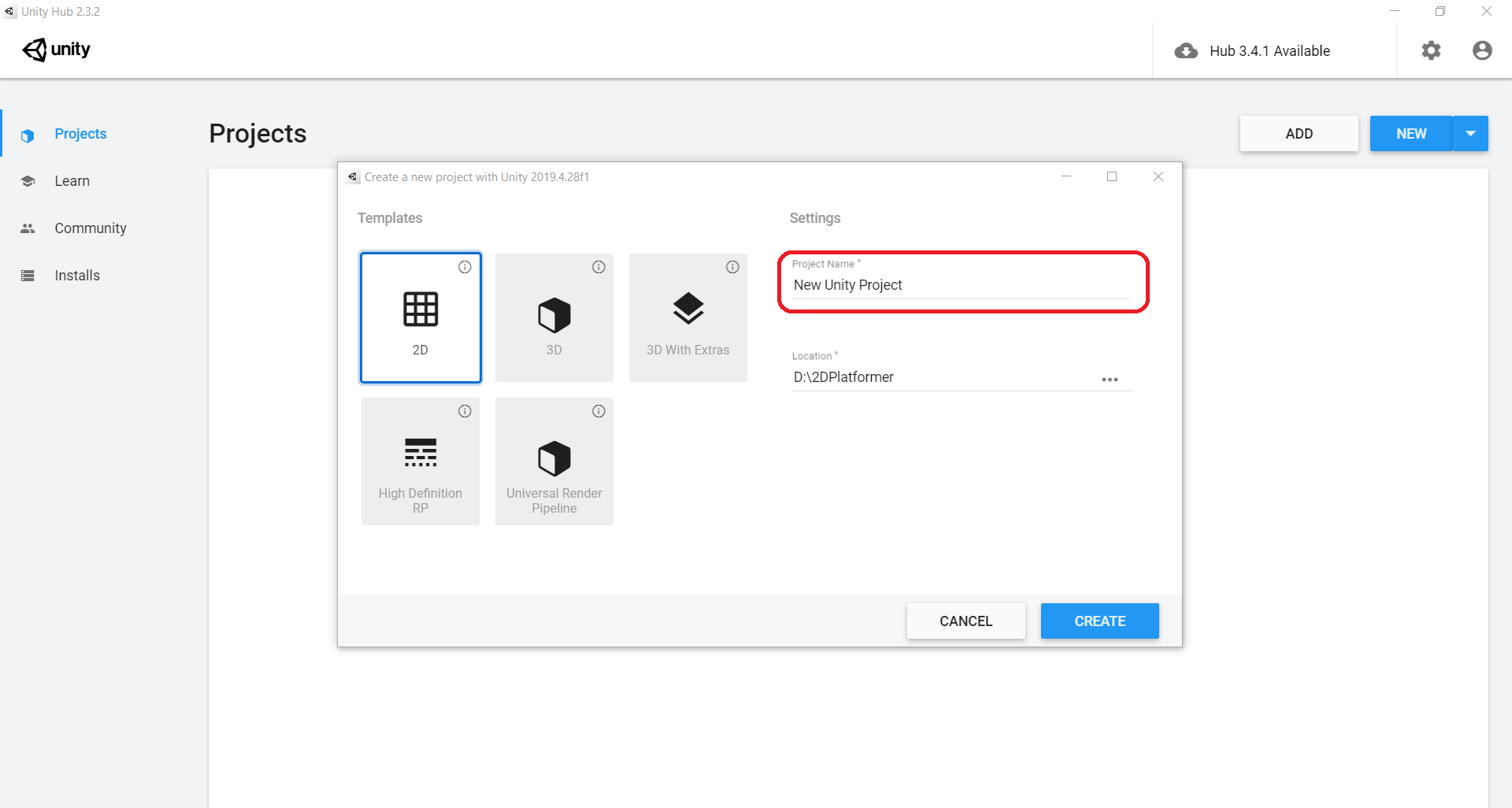


Рисунок 10 Названия проекта

- Путь, куда вы хотите сохранить свой проект

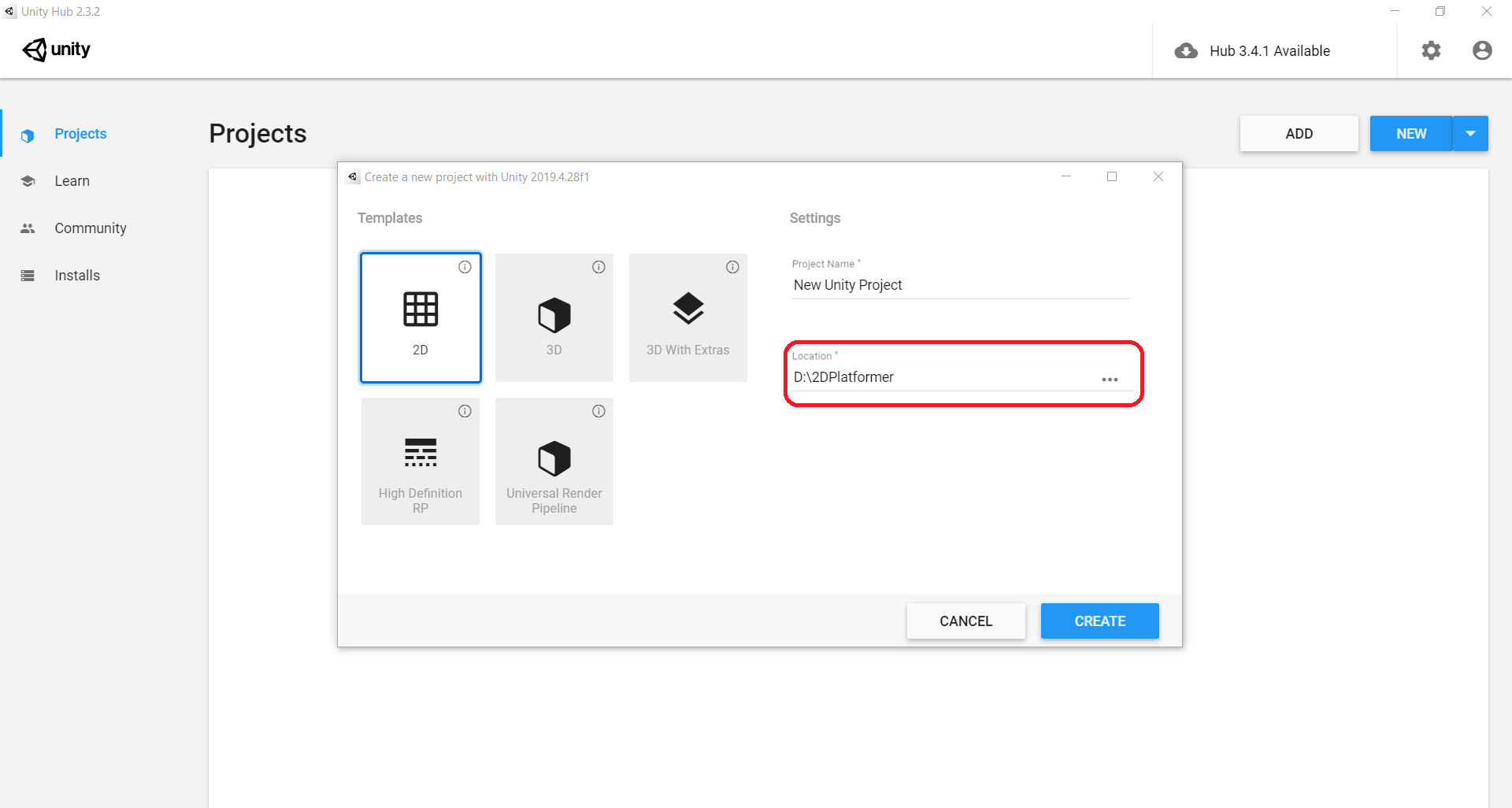


Рисунок 11 Путь сохранения проекта на жёстком диске

4. После того как всё сделали нажимаем кнопку CREATE

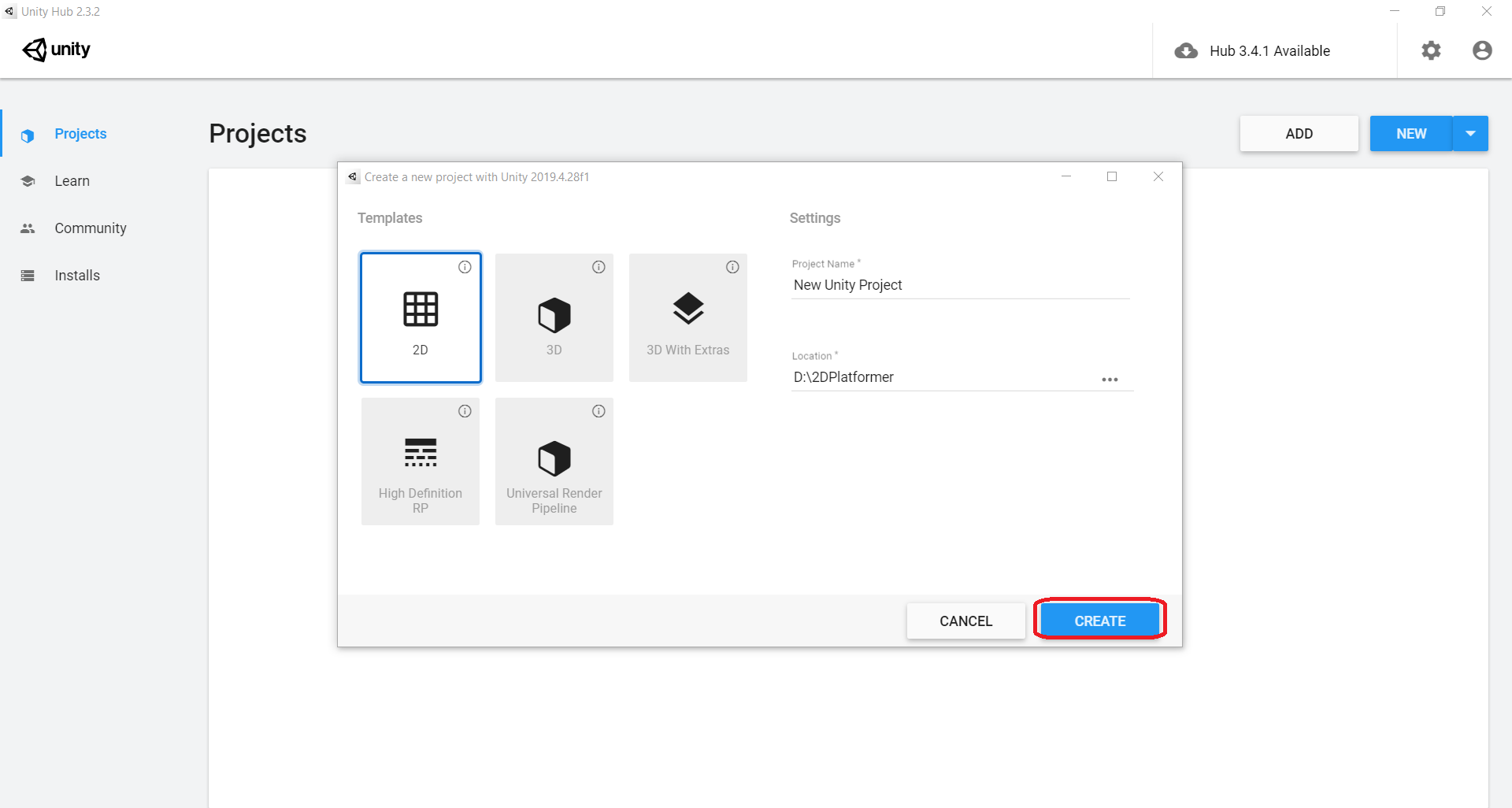


Рисунок 12 Кнопка создания проекта

У нас начнётся создание проекта, это займёт какое-то время.

**Описание среды Unity из документации Unity[[2]](#footnote-2)**

После создания проекта у нас появляется интерфейс Unity.



Рисунок 13 Интерфейс Unity

Окно проекта[[3]](#footnote-3) – отображает вашу библиотеку активов, доступных для использования в вашем проекте. Когда вы импортируете активы в свой проект, они появляются здесь.

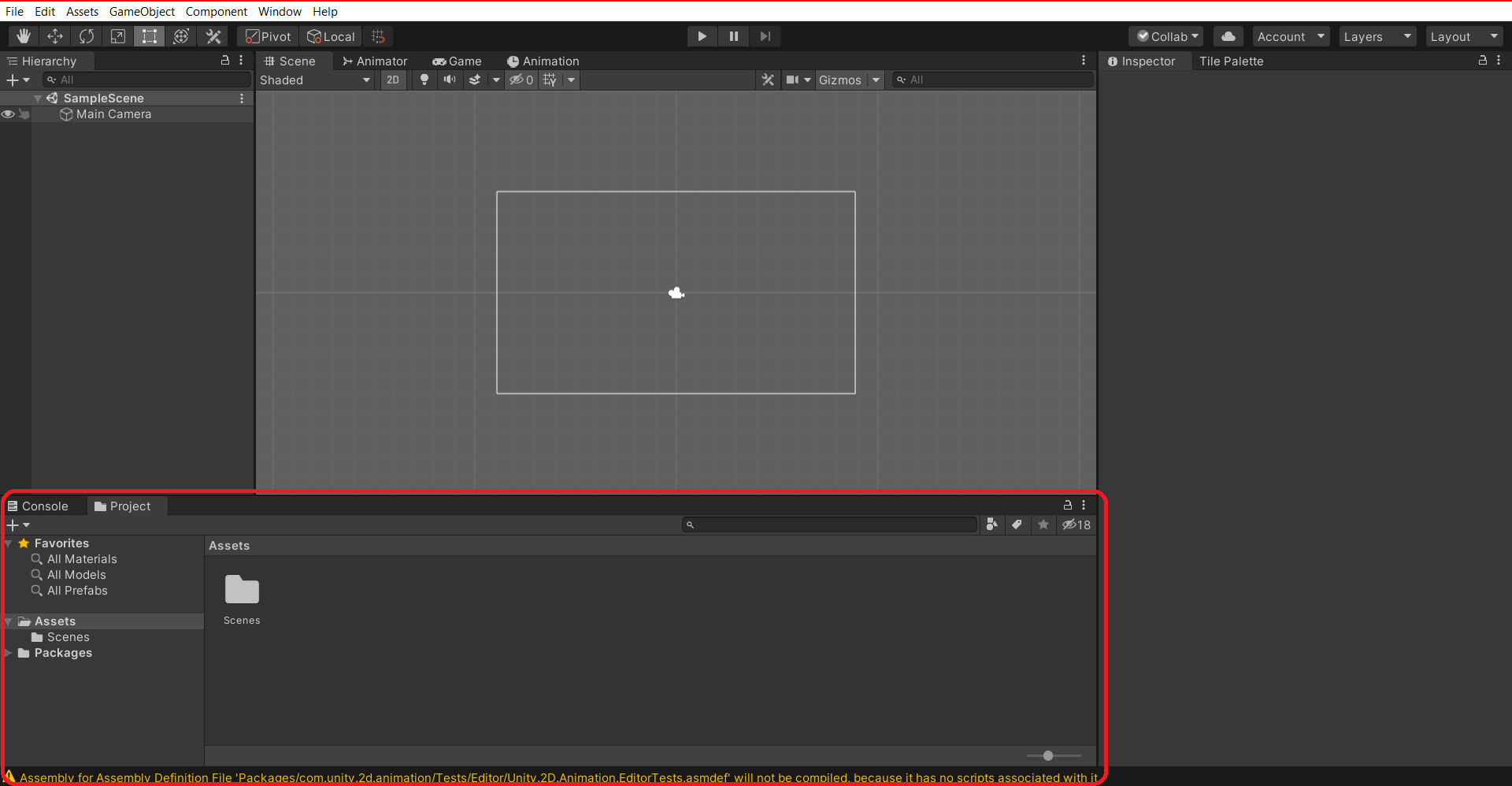


Рисунок 14 Окно проекта

Вид сцены[[4]](#footnote-4) – позволяет визуально перемещаться по сцене и редактировать её. Вид сцены может отображать трёхмерную или двухмерную перспективу, в зависимости от типа проекта, надо которым вы работаете.

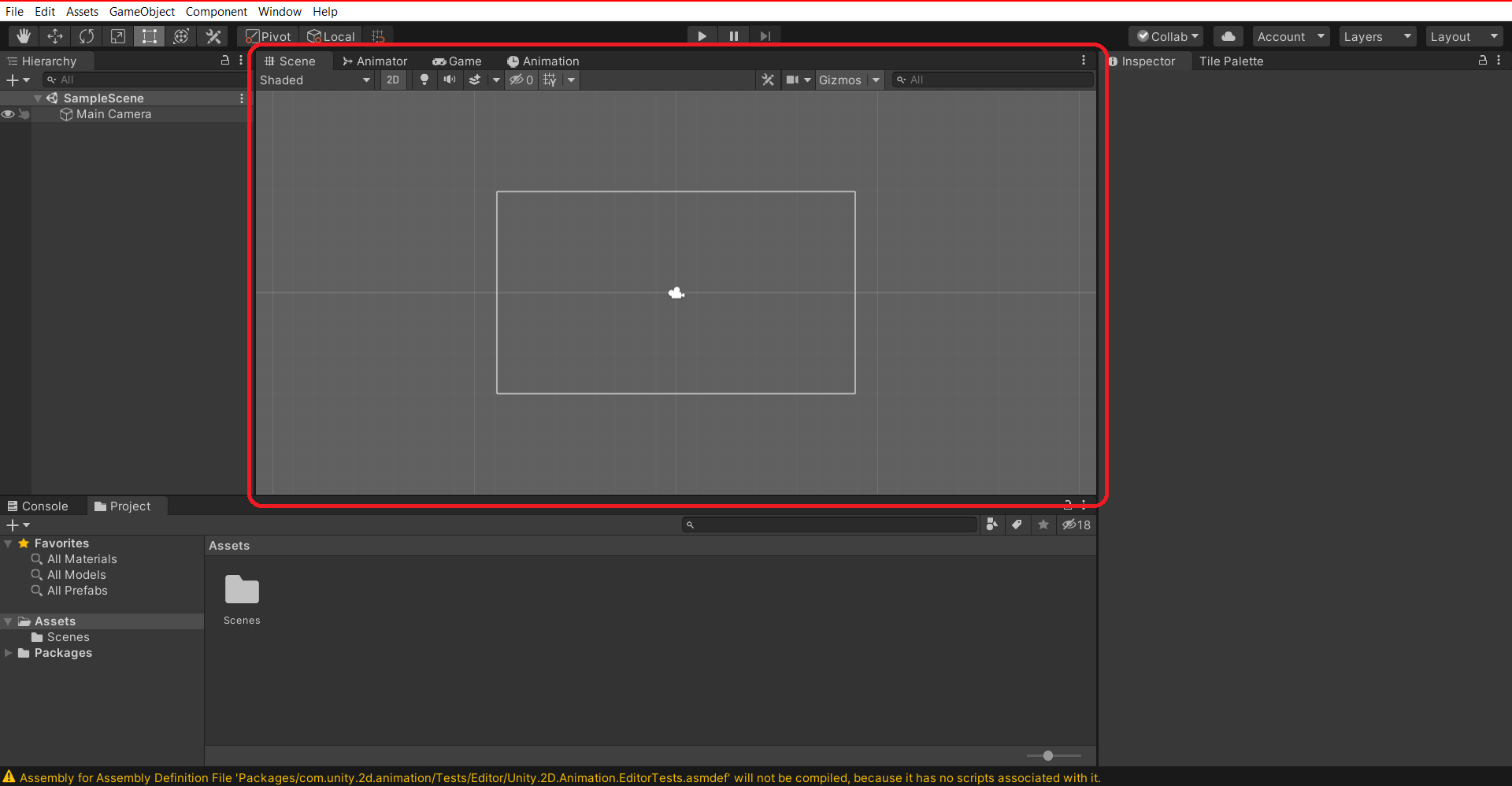


Рисунок 15 Вид сцены

Окно иерархии[[5]](#footnote-5) – представляет собой иерархическое текстовое представление каждого объекта в сцене. У каждого элемента сцены есть запись в иерархии, поэтому два окна неразрывно связаны. Иерархия раскрывает структуру того, как объекты связаны друг с другом.

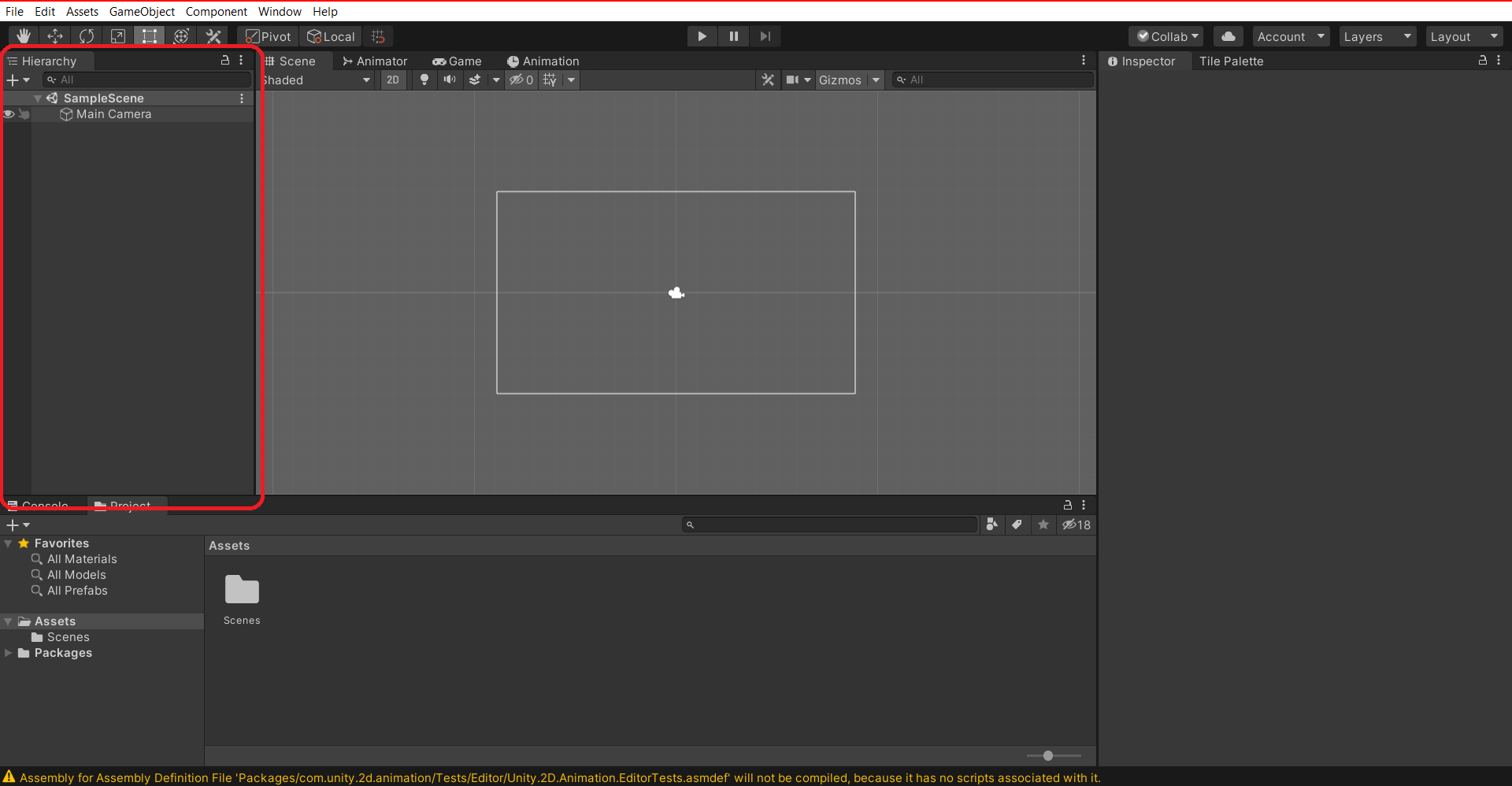


Рисунок 16 Окно иерархии

Окно инспектора[[6]](#footnote-6) – позволяет вам просматривать и редактировать все свойства выбранного в данный момент объекта. Поскольку разные типы объектов имеют разные наборы свойств, макет и содержимое окна инспектора будут различаться.

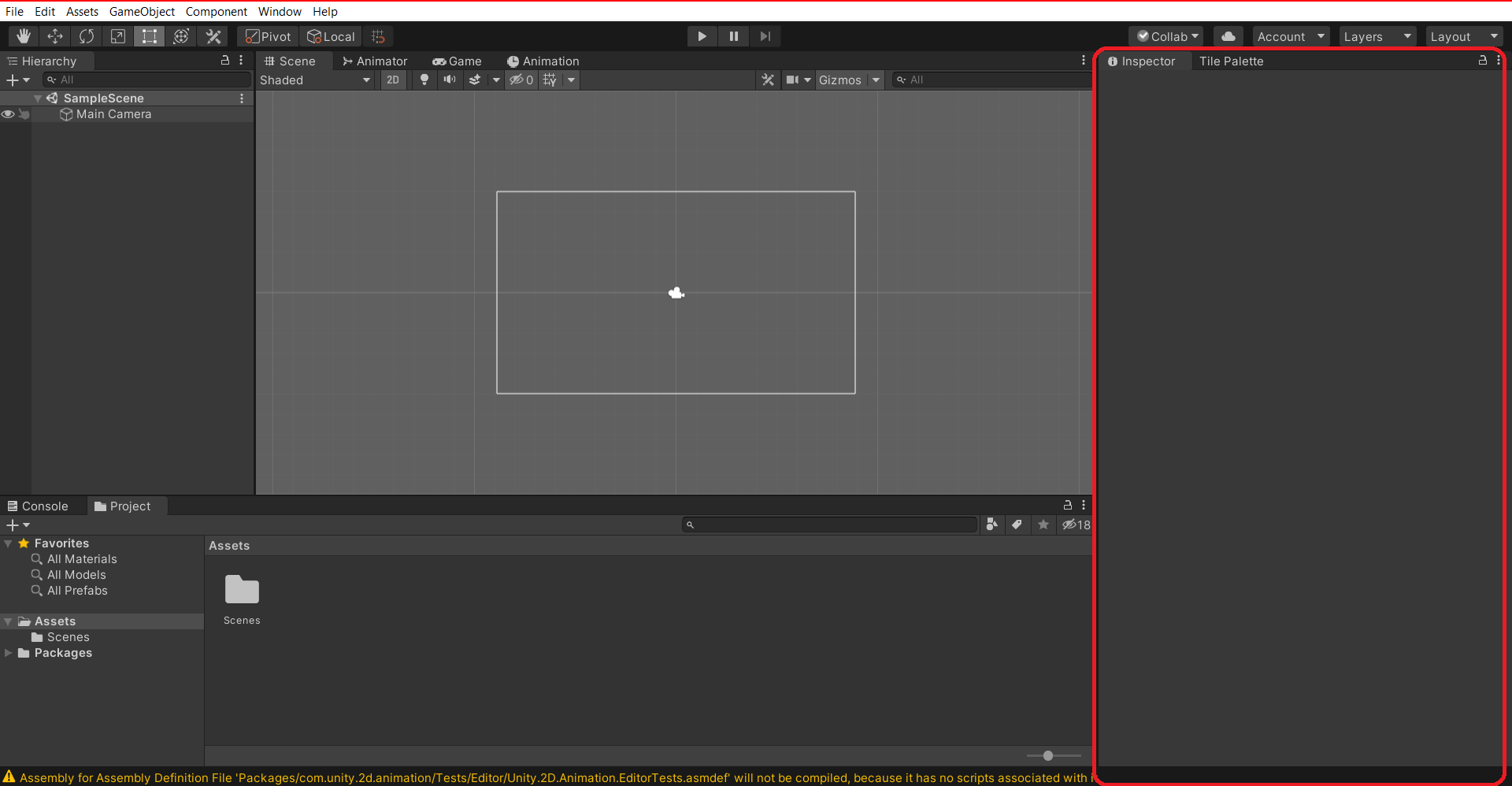


Рисунок 17 Окно инспектора

Панель инструментов[[7]](#footnote-7) – обеспечивает доступ к наиболее важным рабочим функциям. Слева он содержит основные инструменты для управления видом сцены и объектами внутри него. В центре находятся элементы управления воспроизведением, паузой и шагом. Кнопки справа дают вас доступ к вашим облачным службам Unity и вашей учётной записи Unity, за которыми следует меню видимости слоев и, наконец, меню макета редактора (которое предоставляет несколько альтернативных макетов для окон редактора и позволяет вам сохранять свои собственные настройки). макеты).

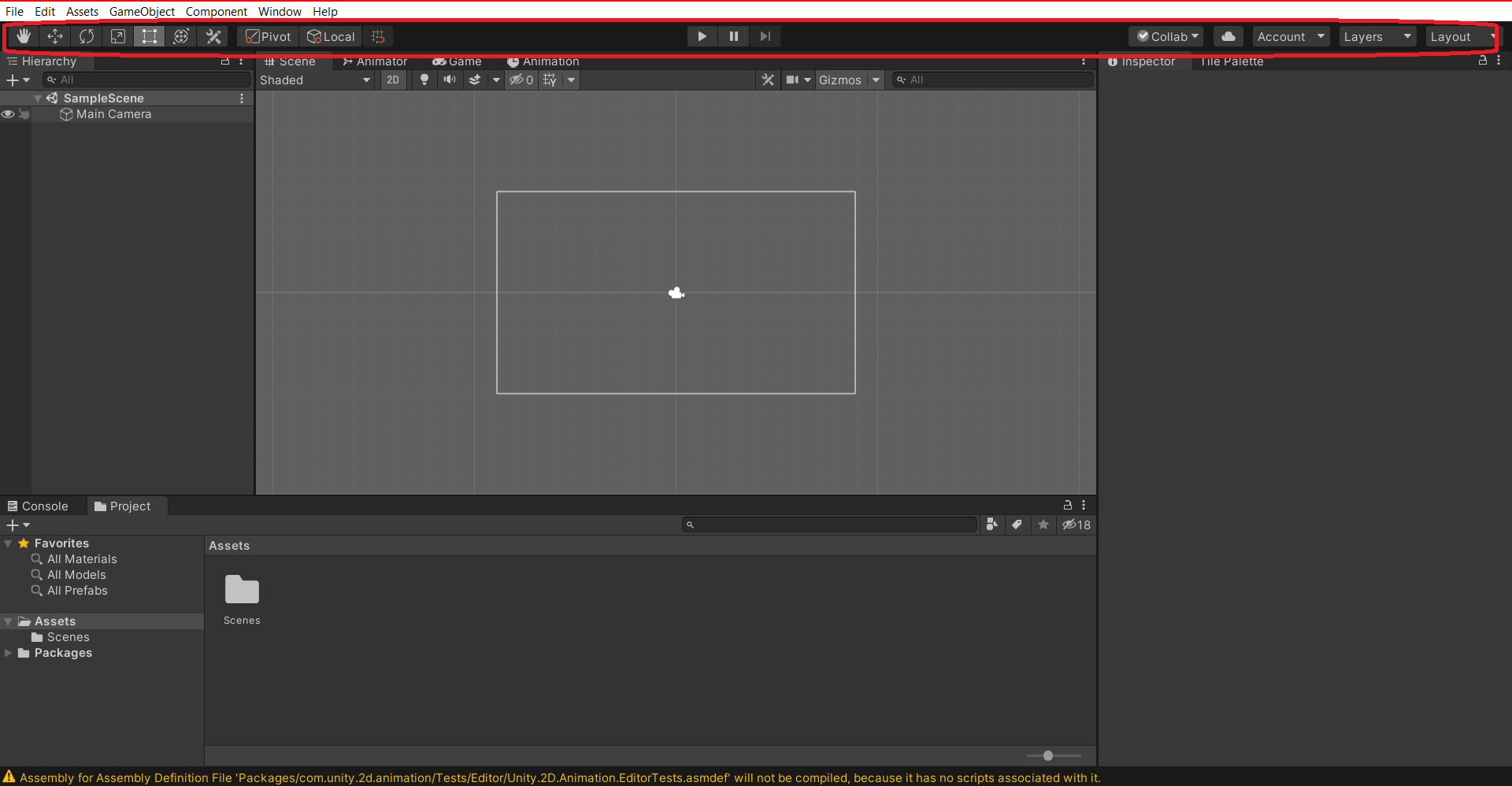


Рисунок 18 Панель инструментов

**Возможности Unity[[8]](#footnote-8)**

В первую очередь Unity дает возможность разрабатывать игры, не требуя для этого каких-то особых знаний. Здесь используется компонентно-ориентированный подход, в рамках которого разработчик создает объекты (например, главного героя) и к ним добавляет различные компоненты (например, визуальное отображение персонажа и способы управления им).

Благодаря удобному Drag & Drop интерфейсу и функциональному графическому редактору движок позволяет рисовать карты и расставлять объекты в реальном времени и сразу же тестировать получившийся результат.

Второе преимущество движка – наличие огромной библиотеки ассетов и плагинов, с помощью которых можно значительно ускорить процесс разработки игры. Их можно импортировать и экспортировать, добавлять в игру целые заготовки – уровни, врагов, паттерны поведения ИИ и так далее. Никакой возни с программированием.

Многие ассеты доступны бесплатно, другие предлагаются за небольшую сумму, и при желании можно создавать собственный контент, публиковать его в Unity Asset Store и получать от этого прибыль.

Третья сильная сторона Unity – поддержка огромного количества платформ, технологий, API. Созданные на движке игры можно легко портировать между ОС Windows, Linux, OS X, Android, iOS, на консоли семейств PlayStation, Xbox, Nintendo, на VR- и AR-устройства.

Unity поддерживает DirectX и OpenGL, работает со всеми современными эффектами рендеринга, включая новейшую технологию *трассировки лучей* в реальном времени.

Физика твердых тел, ragdoll и тканей, система Level of Detail, коллизии между объектами, сложные анимации – все это можно реализовать силами движка.

Стереотипное мнение о том, что движок пригоден только для небольших *инди-игр* и неспособен выдавать красивую картинку, давно уже не актуально: достаточно посмотреть технодемо ADAM, The Blacksmith и Book of the Dead от создателей среды Unity, чтобы убедиться в ее выдающихся способностях.

Наконец, Unity доступен бесплатно, что открывает перед независимыми разработчиками дверь в игровую индустрию. Конечно, существуют ограничения: бесплатная версия движка демонстрирует лого Unity перед запуском игры, а проект, созданный с ее помощью, не должен приносить разработчику больше $100 тысяч в год.

Впрочем, тарифы на подписку не опустошат кошельки даже начинающей команды: Про-версия стоит $150 в месяц, что не так уж много в сравнении с другими движками, причем базовая версия содержит ровно тот же функционал, что и профессиональная.

**Практическая часть**

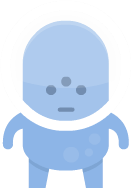
**Описание игры**

Игру, которую буду создавать является 2D платформером.

Платформер[[9]](#footnote-9) - жанр компьютерных игр, в которых основу игрового процесса составляют прыжки по платформам, лазанье по лестницам, сбор предметов, необходимых для победы над врагами или завершения уровня.

В игре будут враги, головоломки и магазин. Наша задача будет пройти игру выполнив все головоломки и пройдя все уровни.

**Создание персонажа**



Для начала нам нужен персонаж.

В окне иерархии нажимаем правую кнопку мыши -> Create Empty

Мы создали пустой объект на сцене.

Далее мы должны дать объекту физику.

В иерархии нажимаем на нашего персонажа -> в окне инспектора кнопку Add Component -> в поиске ищем Capsule Collider 2D, RigidBody 2D, Sprite Renderer.

Collider 2D - 2D-компоненты коллайдера определяют форму 2D-игрового объекта для физических столкновений. Невидимый коллайдер не обязательно должен быть точно такой же форма, как и сетка игрового объекта; на самом деле грубое приближение часто более эффективно и неразличимо в игровом процессе

Все коллайдеры для 2D GameObjects имеют имена, оканчивающиеся на "2D". Коллайдер, в названии которого нет "2D", предназначен для использования в 3D GameObject. Обратите внимание, что вы не можете смешивать 3D-игровые объекты и 2D-коллайдеры или 2D-игровые объекты и 3D-коллайдеры.

RigidBody 2D - класс Rigidbody2D по существу обеспечивает те же функции в 2D, что и Rigidbody в 3D. Добавление компонента Rigidbody2D к спрайту ставит его под контроль физического движка. Само по себе это означает, что на спрайт будет воздействовать гравитация и им можно будет управлять из скриптов с помощью сил. Добавив соответствующий компонент коллайдера, спрайт также будет реагировать на столкновения с другими спрайтами. Такое поведение полностью обусловлено физическое системой Unity; требуется очень мало кода, чтобы получить впечатляющее и аутентичное физическое поведение, и позволяет реализовать "эмерджентрный" игровой процесс, который не был явно закодирован в игре.

Sprite Renderer - визуализирует Sprite и управляет его визуальным отображением в сцене как для 2D, так и для 3D проектов.

**Создание камеры на игрока**

Создаём новый скрипт – слежения за персонажем

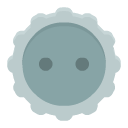
В скрипте пишем данный код (Приложение 1)

Сохраняем скрипт.

Данный скрипт надо присвоить объекту Main Camera. Перенести скрипт прямо на данный объект.

Дальше выбираем Main Camera и в окне инспектора ищем наш скрипт -> в переменную Target надо из иерархии переместить нашего игрового персонажа.

**Создание врага**



В иерархии создаём пустой объект и накладываем Collider 2D.

Нашему врагу надо присвоить Tag

В иерархии выбираем врага -> в инспекторе сверху меню Tag -> Add Tag... -> Tags и нажимаем на плюсик -> любое название и кнопку Save.

Возвращаемся на нашего врага и в инспекторе присваиваем наш добавленный Tag

**Основные блоки кода**

Реализация ходьбы персонажа (Приложение 2)

Метод FixedUpdate - может вызываться несколько раз в кадре. Вызывается регулярно и нормированно, через фиксированные интервалы времени. Используется для эмуляции физических характеристик объектов (св-ва компонента RigiBody) LateUpdate. Вызывается в каждом кадре как Update но только после событий Update и FixedUpdate.

if (isClimb) - проверка на лестнице ли игрок или нет.

Дальше идёт код перемещения игрока по вертикальной оси

Ниже передвижение игрока

Реализация прыжка (Приложение 3)

if (isGrounded && !isClimb) - проверка на земле ли игрок И не на лестницы ли он

Выполняется прыжок

Реализация поворота (Приложение 4)

if (joystick.Horizontal > 0) - если джойстик больше нуля то игрок поворачивается вправо

if (joystick.Horizontal > 0) - если джойстик меньше нуля то игрок поворачивается влево

Реализация подбора монет и ключей (Приложение 5)

OnCollisionEnter2D(Collision2D other) - передается когда входящий коллайдер контактирует с коллайдером данного объекта (только 2D физика).

if (other.gameObject.CompareTag("Coin")) - проверка название коллайдера при соприкосновение имеет название Coin.

Coin + Coin + 1; - прибавляет 1 монетку.

CoinText.text = Coin + ""; - текстом показывает количество коинов.

Destroy(other.gameObject); - удаляет подобранную монету.

if (other.gameObject.CompareTag("keyGreen")) - проверка название коллайдера при соприкосновение имеет название keyGreen.

GreenKey++; - добавляет 1 зелёный ключ

Destroy(other.gameObject); - удаляет подобранный зелёный ключ.

if (other.gameObject.CompareTag("keyRed") - проверка название коллайдера при соприкосновение имеет название keyRed.

GreenRed++; - добавляет 1 красный ключ

Destroy(other.gameObject); - удаляет подобранный красный ключ.

Реализация врага (Приложение 5)

if (other.gameObject.CompareTag("Saw")) - проверка название коллайдера при соприкосновении имеет название Saw.

maxHeart = maxHeart -0.5f; - отнимаем здоровье у персонажа

**Сбор игры под Android**

Созданную игру осталось «перекинуть» на андроид.

Нажимаем в левом верхнем углу File -> Build Settings

Выбираем систему Android и ждём.

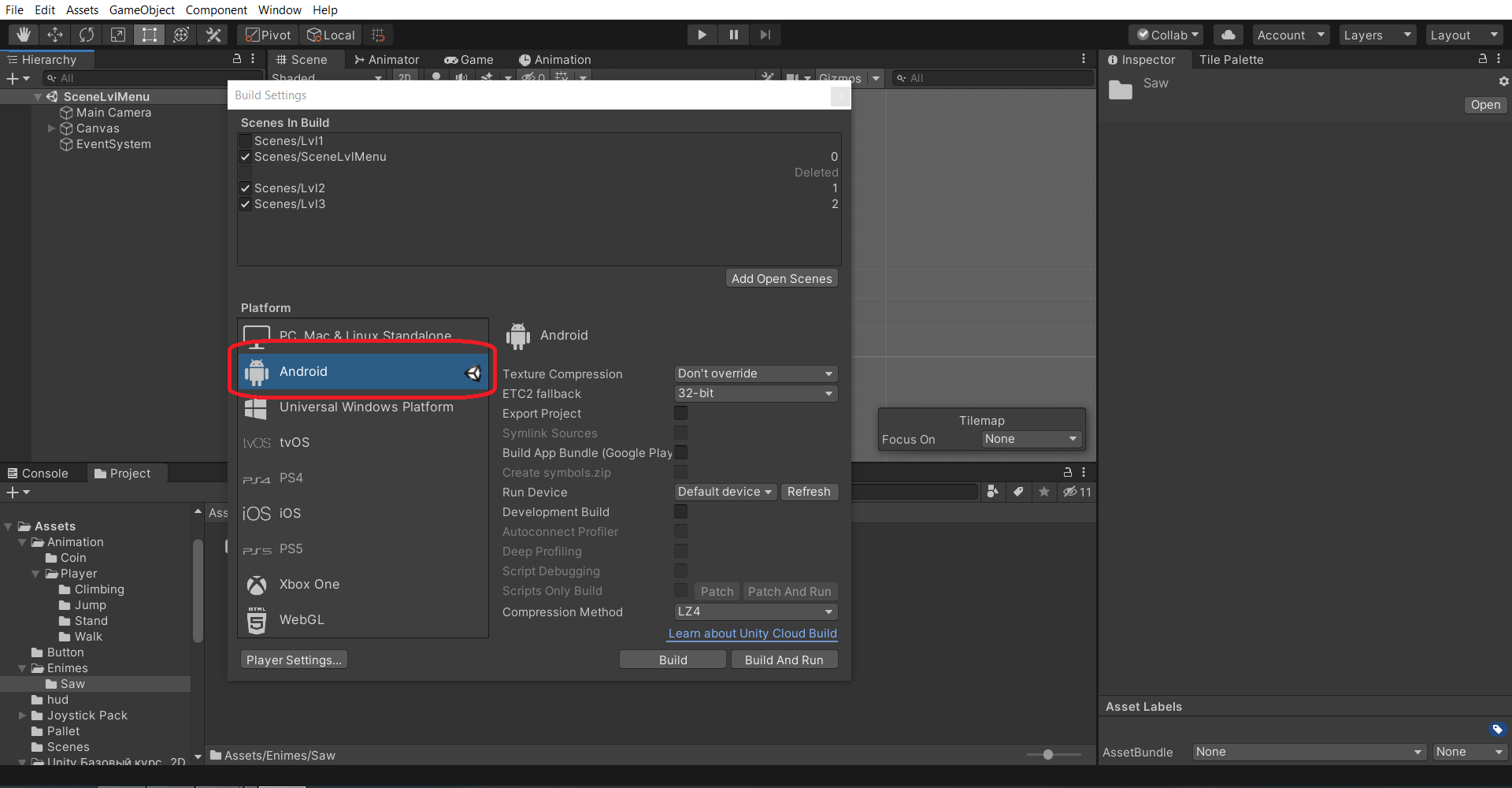


Рисунок 19 Выбор Android системы

После того как приложение пересобралось под андроид, нажимаем Build.

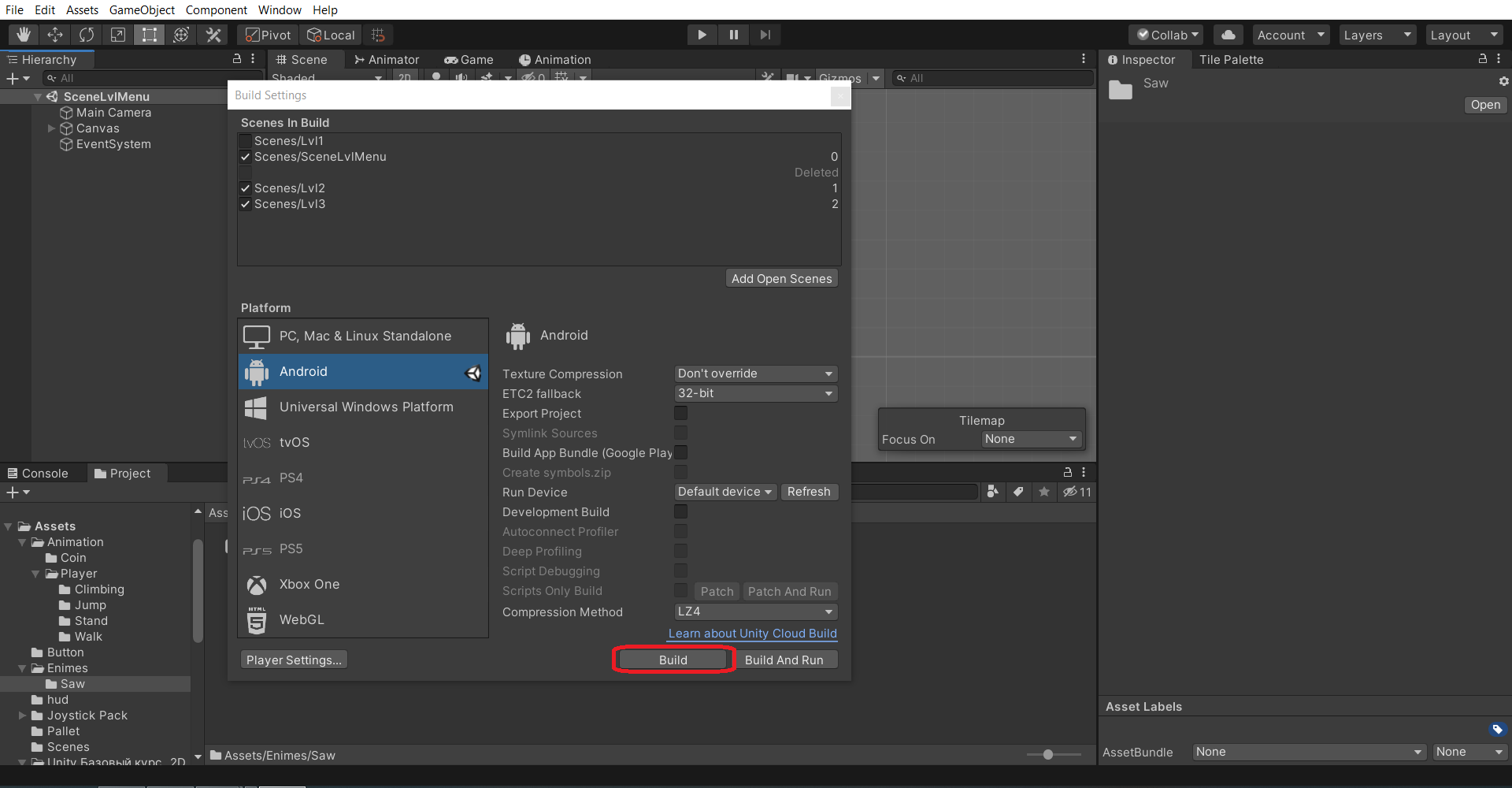


Рисунок 20 Кнопка компилирования

Компилируем приложение как APK файл на телефон.

**Заключение**

В результате создания игры 2D на платформе Unity, был реализован проект - приключения Кенни, создан персонаж - Кенни и уровни.

Решены следующие задачи:

1. Изучен интерфейс и функционал Unity.

2. Создана 2D 3-х уровневый платформер в среде Unity.

3. Скомпилирован код для использования в мобильной системе на базе Android.

В современном мире создание видеоигр является одним из наиболее крупных сегментов индустрии развлечений. Масштабы игровой индустрии сопоставимы, например, с киноиндустрией. А по скорости роста за последние пять лет индустрия видеоигр существенно ее опережала. По степени влияния на потребителей и вовлеченности их в интерактивное окружение, предлагаемое видеоиграми, этот сегмент уже давно выделяется среди других видов развлечений. Геймдев или разработку игр невозможно рассматривать обособленно от индустрии компьютерных игр в целом. Непосредственно создание игр – это только часть комплексной «экосистемы», обеспечивающей полный  жизненный цикл производства, распространения  и потребления таких сложных продуктов, как компьютерные игры.

Перспектива

Есть два направления развития профессионального геймдизайнера:

Вертикальное, предполагающее карьерный рост с изменением должности. Обычно геймдизайнеры вырастают в креативных продюсеров, креативных литературных редакторов, руководителей направлений, руководителей студий.

Горизонтальное, которое предполагает повышение авторитета и расширение зоны ответственности геймдизайнера. Рост до старшего или ведущего гейм-дизайнера. Становление как эксперта, а не как руководителя.

**Источники**

1. Руководство пользователя Unity 2021.3 (LTS) <https://docs.unity3d.com/Manual/>

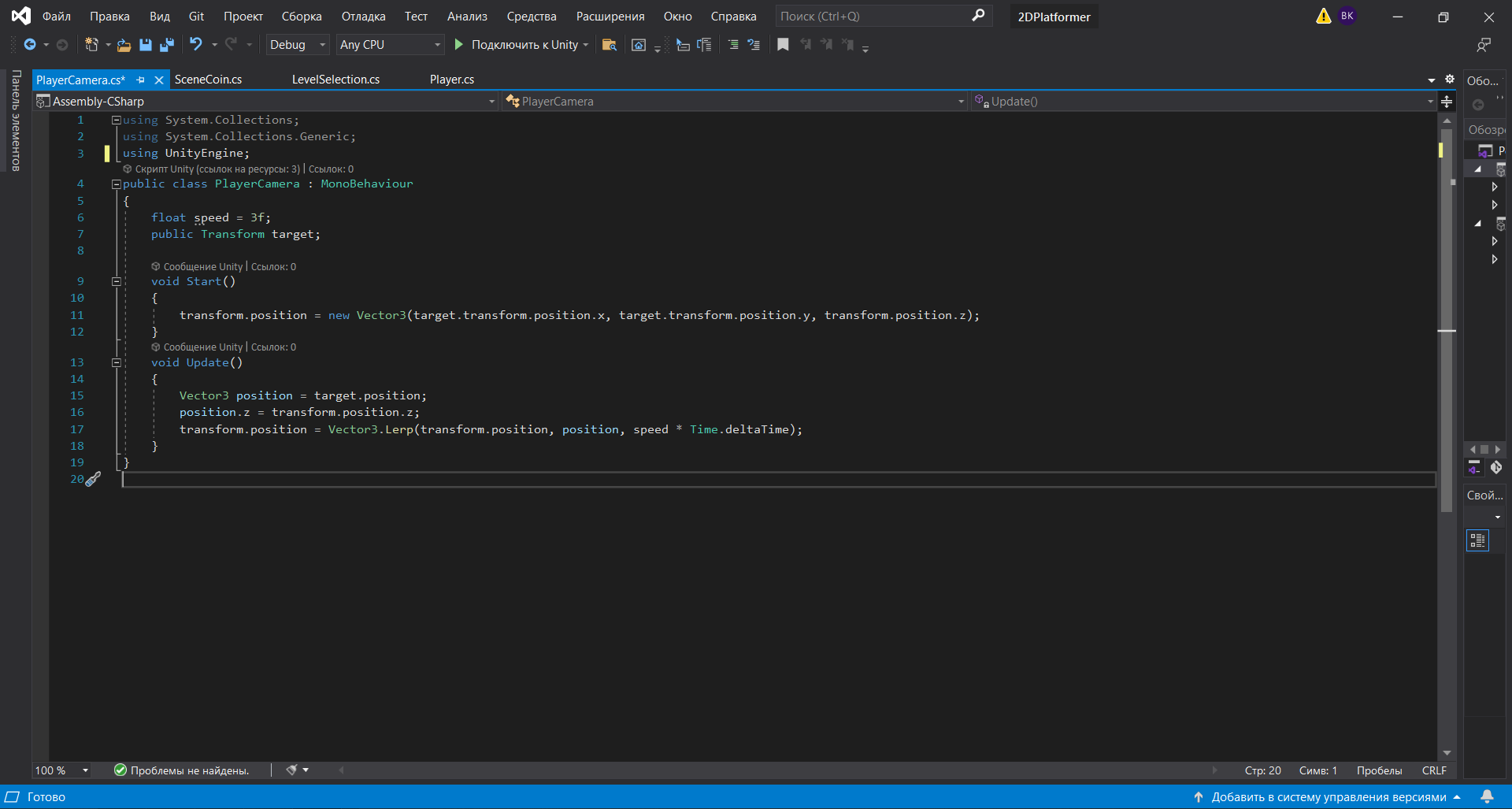
2. Unity (Игровой движок) <https://ru.wikipedia.org/wiki/Unity_(игровой_движок)>

3. (8) Официальные уроки Unity Юнити на русском <https://www.youtube.com/playlist?list=PLpyssslyeRz6Yd4SdrY-O_kyFiyeK8w6l>

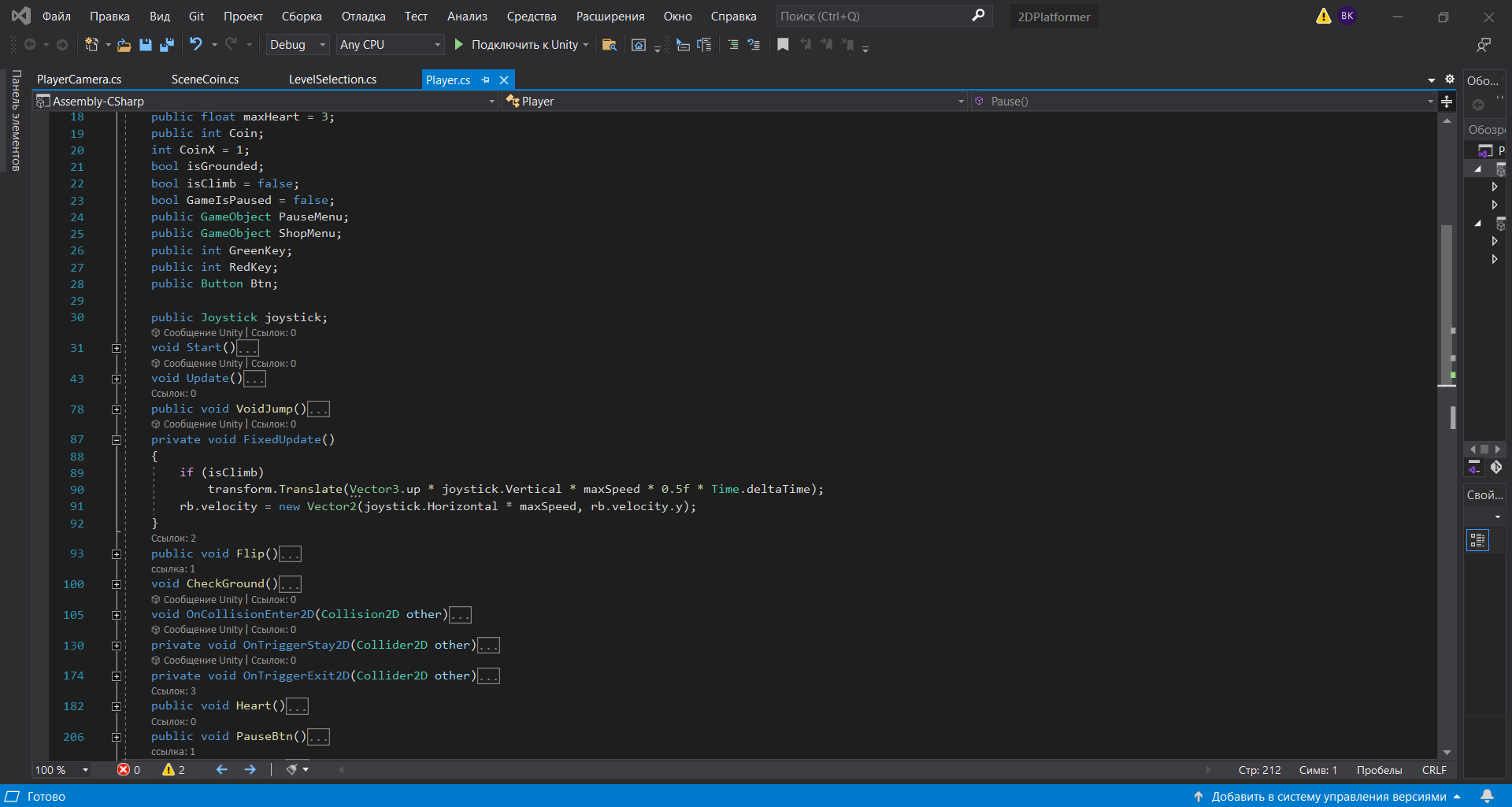
4. Книга Бонд Д. "Unity и C#. Геймдев от идеи до реализации. 2-е изд."

**Приложение**

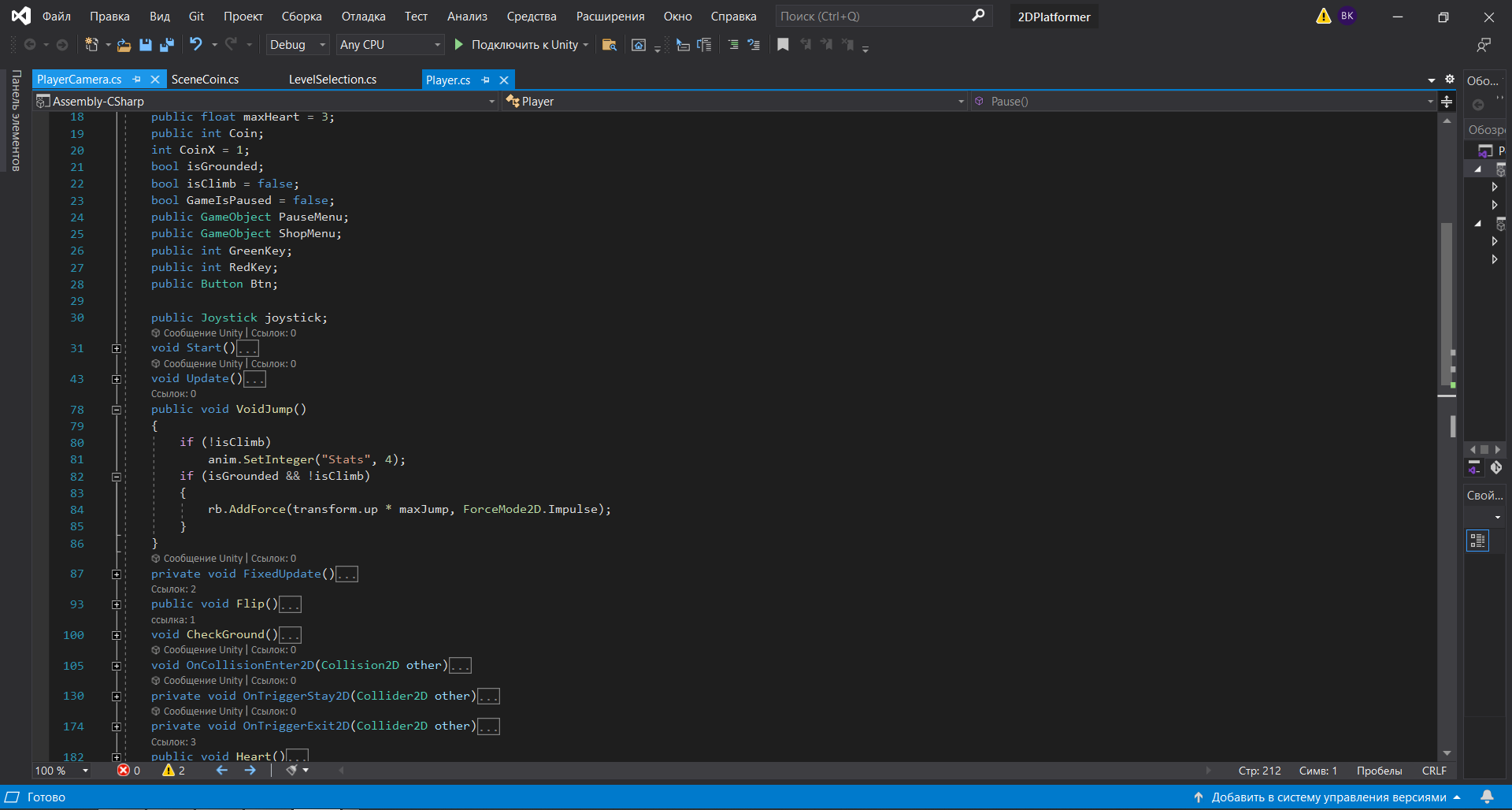
Приложение 1



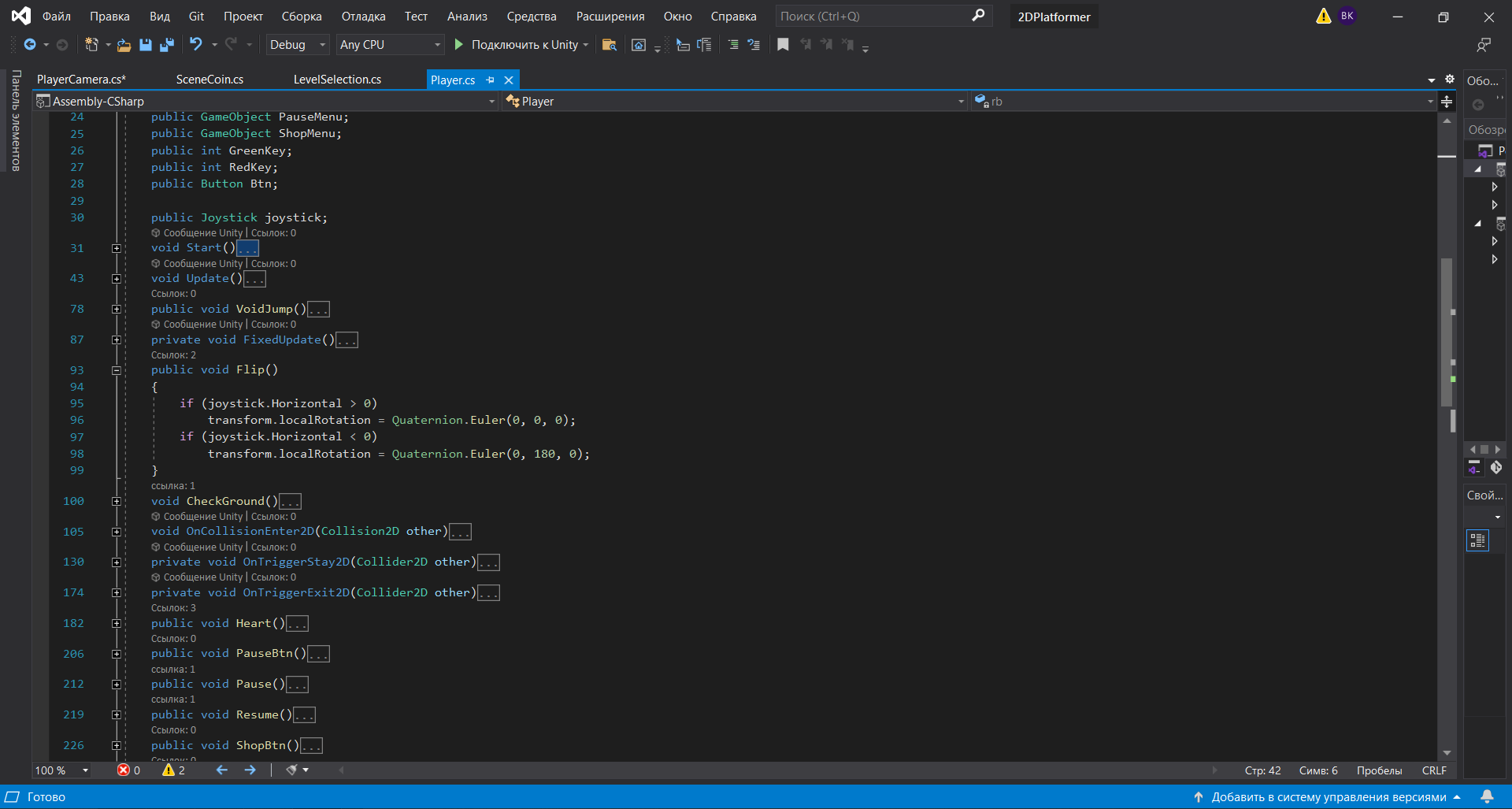
Приложение 2



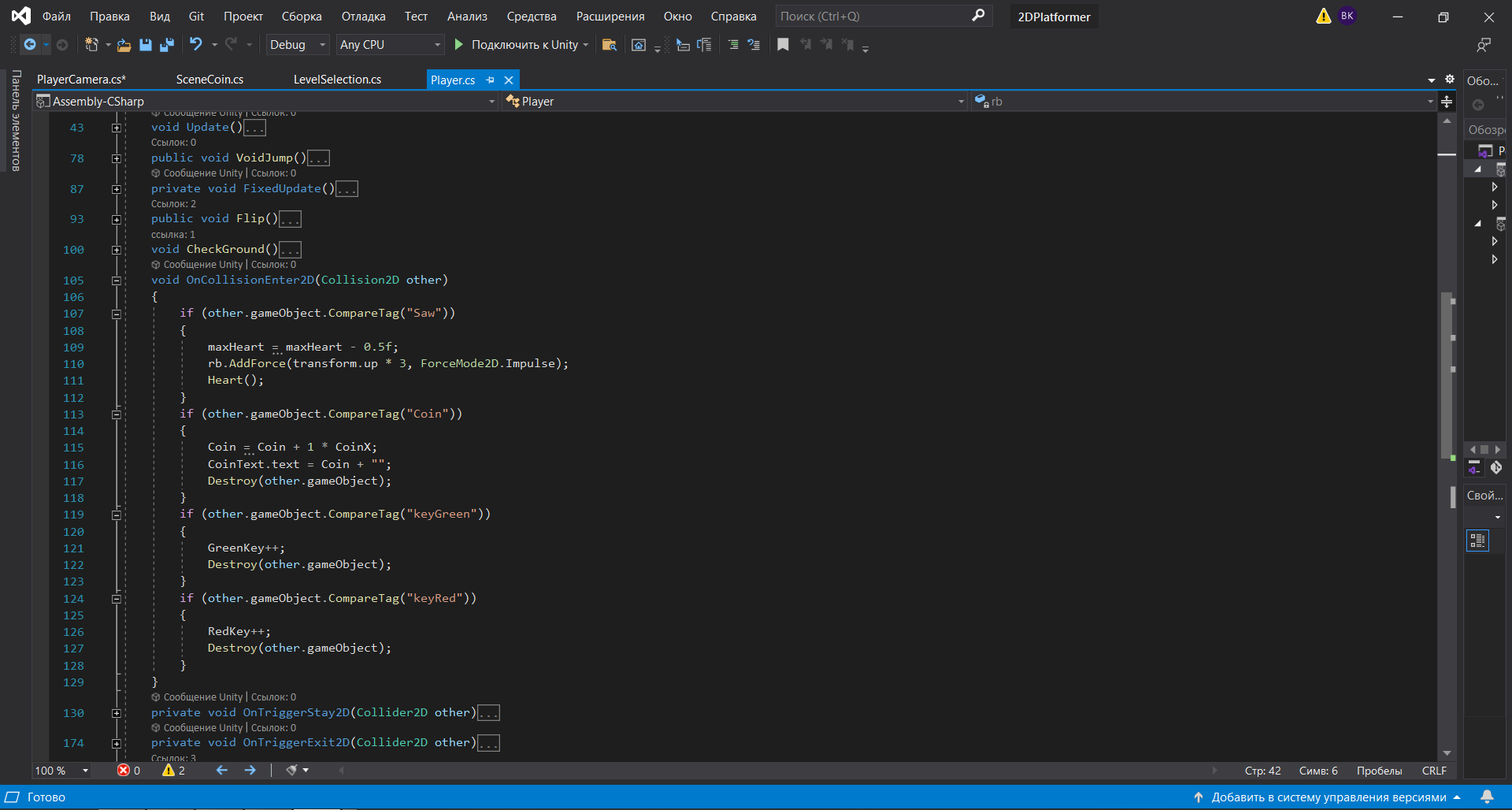
Приложение 3



Приложение 4



Приложение 5



1. https://unity.com/ru [↑](#footnote-ref-1)
2. https://docs.unity3d.com/Manual/ [↑](#footnote-ref-2)
3. https://docs.unity3d.com/ru/2018.4/Manual/ProjectView.html [↑](#footnote-ref-3)
4. https://docs.unity3d.com/Manual/CreatingScenes.html [↑](#footnote-ref-4)
5. https://docs.unity3d.com/Manual/Hierarchy.html [↑](#footnote-ref-5)
6. https://docs.unity3d.com/Manual/UsingTheInspector.html [↑](#footnote-ref-6)
7. https://docs.unity3d.com/ru/2021.1/Manual/Toolbar.html [↑](#footnote-ref-7)
8. https://unity.com/ru/pages/more-than-an-engine [↑](#footnote-ref-8)
9. https://ru.wikipedia.org/wiki/Платформер [↑](#footnote-ref-9)