

Применение физического движка Unity для создания игр

Шайдуров Александр Александрович

Приамурский государственный университет им. Шолом-Алейхема

студент

Аннотация

В данной статье рассматривается возможность применения физического движка для создания трёхмерных компьютерных игр, на примере Unity.

Ключевые слова: 3D игра, Unity, Havok, PhysX, разработка игры, физический движок.

The use of physical Unity engine for creating games

Shaidurov Aleksandr Aleksandrovich

Sholom-Aleichem Priamursky State University

student

Abstract

This article discusses the possibility of using a physical engine to create three-dimensional computer games, using the example of Unity.

Keywords: 3D game, Unity, Havok, PhysX, game development, physics engine.

В наши дни видео игры стали повседневным увлечением многих людей, они позволяют развлечься им в свободное время. Разнообразие жанров открывает большой выбор, присутствуют такие как: шутеры, гонки, приключения, аркады, квесты, симуляторы и другие. Для игр, где требуется взаимодействие, и проявление физических свойств объекта, для используется физический движок.

Целью данной статьи является анализ нескольких существующих продуктов, которые позволяют создать симуляцию физического взаимодействия, для разработки видео игр и реализация взаимодействия на одном из них.

В последнее время используется физические движки такие как Havok, Digital Molecular Master, PhysX, Bullet Physics Library, Open Dynamics Engine, Unity, Tokamak, Newton Game Dynamics, Vox2D. Первые два представленных являются проприетарными, остальные бесплатные и находятся в свободном доступе.

Havok Physics или более известный как Havok - физический движок, разработанный ирландской компанией Havok. Движок создаёт симуляцию физического взаимодействия в реальном времени, что делает мир игры более живым и реалистичным, подобно физике Ragdoll. Havok Physics является кроссплатформенным движком. Он использовался более чем в 150

компьютерных играх. Физическая часть движка Source, созданного Valve Corporation и использующегося в игре Half-Life 2 и её дополнениях, включает в себя часть переработанного кода Havok Physics.

PhysX первоначально разрабатывался компанией Ageia для своего физического процессора PhysX. После того, как Ageia была приобретена nVidia, движок перешёл в собственность компании nVidia, которая продолжает его дальнейшую разработку. nVidia адаптировала движок для ускорения физических расчётов на своих графических чипах с архитектурой CUDA. PhysX может также производить вычисления с использованием обычного процессора. В настоящее время PhysX доступен на следующих платформах: Windows, Linux, Mac OS X, Wii, PlayStation 3, Xbox 360, PlayStation 4, Xbox One, Android, IOS (аппаратное ускорение возможно только на платформе Windows). Движок используется во многих играх и активно предлагается для продажи (лицензирования) всем желающим.

Движок Unity один из наиболее простых в обращении, большим преимуществом является его доступность и поддержка платформами: Windows, Linux, Mac OS, iOS, Android. Выпуск Unity состоялся в 2005 году, и с того времени постоянно обновляется и недавно получил поддержку VR, для реализации игр с виртуальной реальностью. Также Unity поддерживает физику твёрдых тел и ткани, а также физику типа Ragdoll (тряпичная кукла). В редакторе имеется система наследования объектов; дочерние объекты будут повторять все изменения позиции, поворота и масштаба родительского объекта. Скрипты в редакторе прикрепляются к объектам в виде отдельных компонентов. При импорте текстуры в Unity можно сгенерировать alpha-канал, mip-уровни, normal-map, light-map, карту отражений, однако непосредственно на модель текстуру прикрепить нельзя — будет создан материал, которому будет назначен шейдер, и затем материал прикрепится к модели. Редактор Unity поддерживает написание и редактирование шейдеров. Редактор Unity имеет компонент для создания анимации, но также анимацию можно создать предварительно в 3D-редакторе и импортировать вместе с моделью, а затем разбить на файлы.

В качестве примера демонстрации физического взаимодействия, был выбран Unity, по причине его простоты и доступности. Такая игра как Firewatch, была написана на физическом движке Unity. Firewatch это однопользовательская квест игра, разработанная компанией Campo Santo в 2016 году. Игра имеет достаточно реалистичную графику и физику внутри игры.



Рис. 1. Скриншот из игры Firewatch

В работе Р.И. Баженова, О.С. Ходоса [1] хорошо расписаны возможности применения платформы Unity. А.Б. Свириденко и Б.А. Гладкий в своей статье описали все преимущества применения движка Unity для создания игр [2]. В книге W. Goldstone описываются основы работы с средой Unity и приводится пример создания полноценной трехмерной игры [3]. С.А.Винокуров и Р.И.Баженов в своей статье показали реализацию Unity для создания игры «Танк на острове» [4]. Статья М.Е.Кочитова и Д.В.Лучанинова показывает реализацию Unity для создания 3d игры [5].

Скачать Unity можно с официального сайта, установка не занимает много времени. Интерфейс этого физического движка достаточно прост.

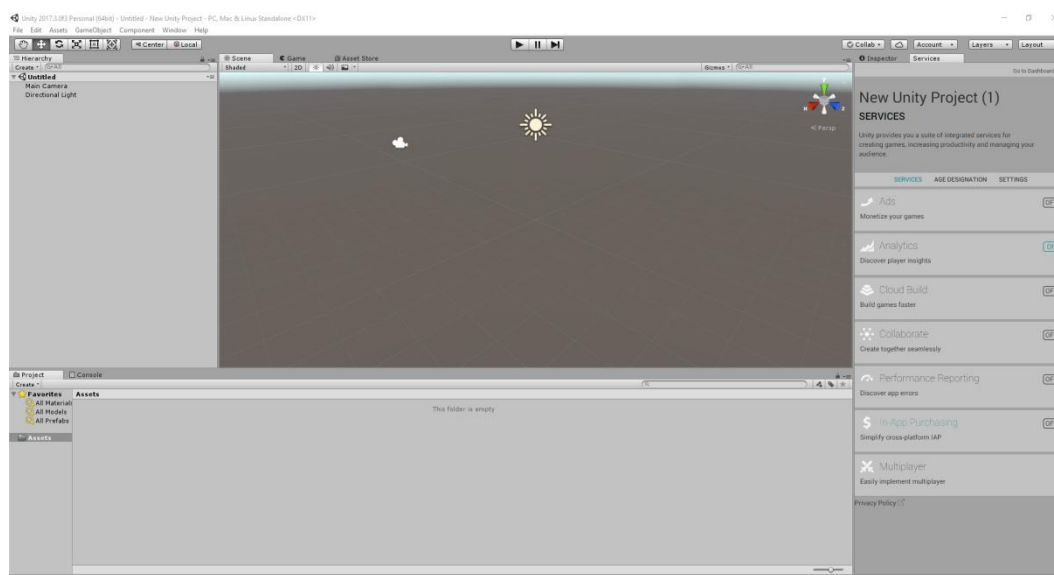


Рис. 2. Интерфейс среды Unity

На рисунке 2 представлен интерфейс среды разработки Unity, в нем есть все необходимые элементы, окна, инструменты, наборы для разработки своей будущей игры. Справа находится окно «Inspector», которое содержит свойства различных объектов, инструментов, файлов и прочего. По центру

расположилось окно «Scene», в которой видно в перспективе саму сцену и что в ней находится, также вверху есть вкладка «Game», показывающая сам игровой процесс на сцене. Окно «Hierarchy» имеет список всех добавленных объектов на сцену, которые можно выбирать, если сложно найти в окне «Scene» сам этот нужный игровой объект. Окно «Project» представляет собой проводник всех файлов для проекта: скрипты, изображения, текстуры, модели и т.д.

Для демонстрации физического взаимодействия, была создана локация, где будет происходить действие. Для это был добавлен Terrain объект.

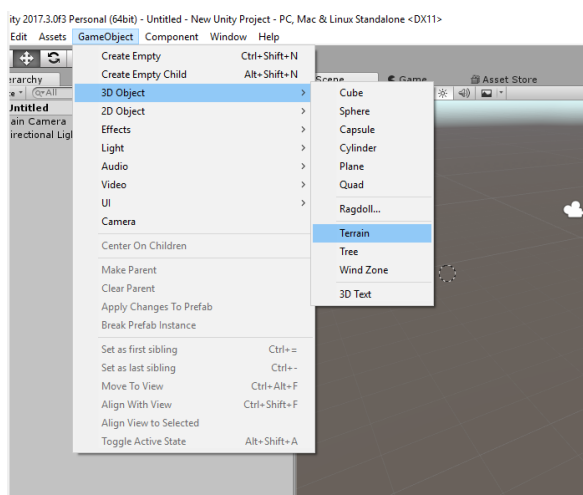


Рис.3. Добавление Terrain

С помощью параметров во вкладке Inspection, объекту была придана нужная форму и цвет. Все нужные текстуры были скачены в Assent Store, там присутствуют как бесплатные, так и платные текстуры. Так же там можно найти готовые объекты. Необходимо просто задать в строке поиска то, что необходимо найти и задать параметр Free, если необходимо найти бесплатный вариант.

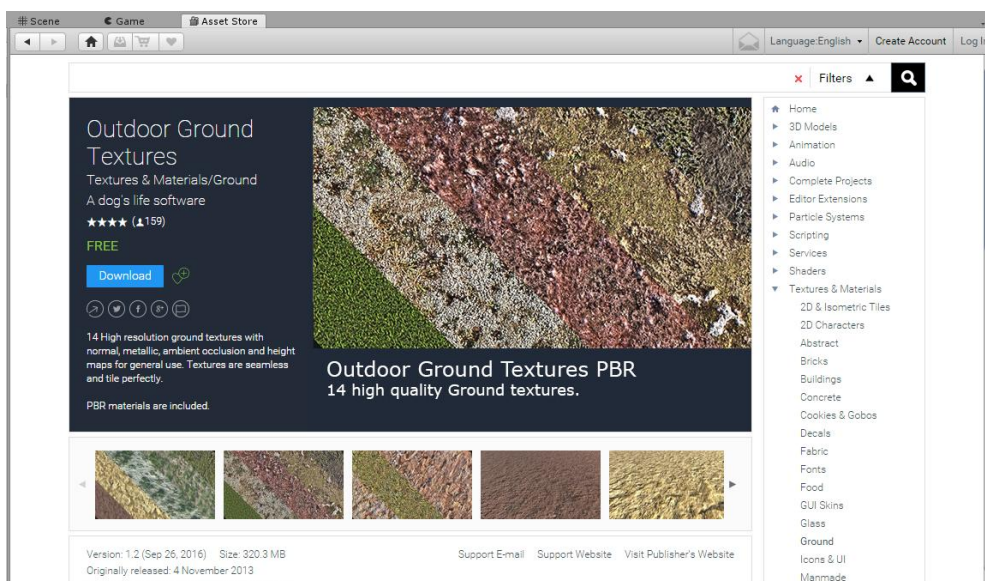


Рис.4. Загрузка необходимого во вкладке Assent Store

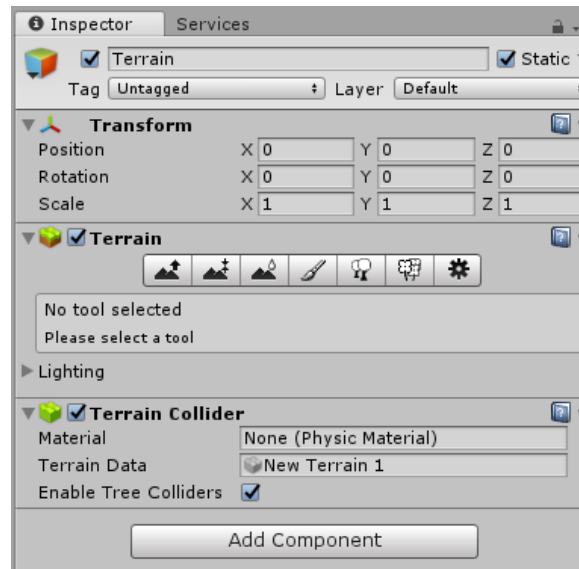


Рис.5. Параметры Inspection для изменения Terrain

В качестве персонажа, которым можно управлять был сделан автомобиль. Для это в самом Unity есть готовая заготовка, чтобы её получить был загружен нужный пакет в Unity, через Import Package. После выбора Vehicles, и нажатия Import в окне, пакет был установлен в Unity и готов к использованию.

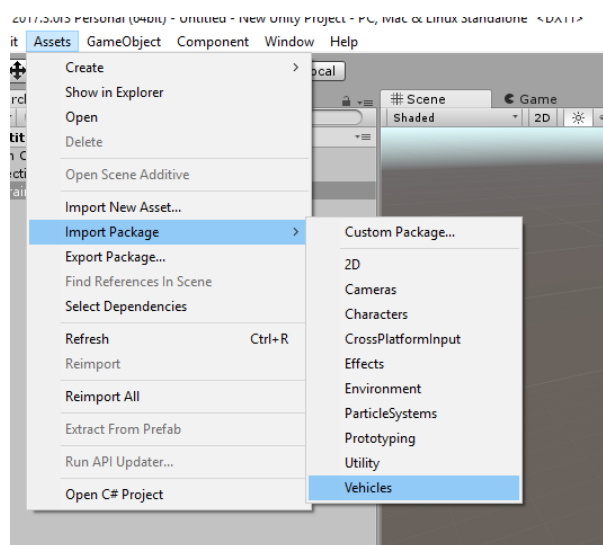


Рис. 6. Расположение дополнительного пакета Vehicles

При создании неба, был изменён компонент SkyBox. Он находится во вкладке Window -> Lighting-> Settings.

В результате нескольких действий, была получена локация, по которой можно передвигаться на автомобиле.

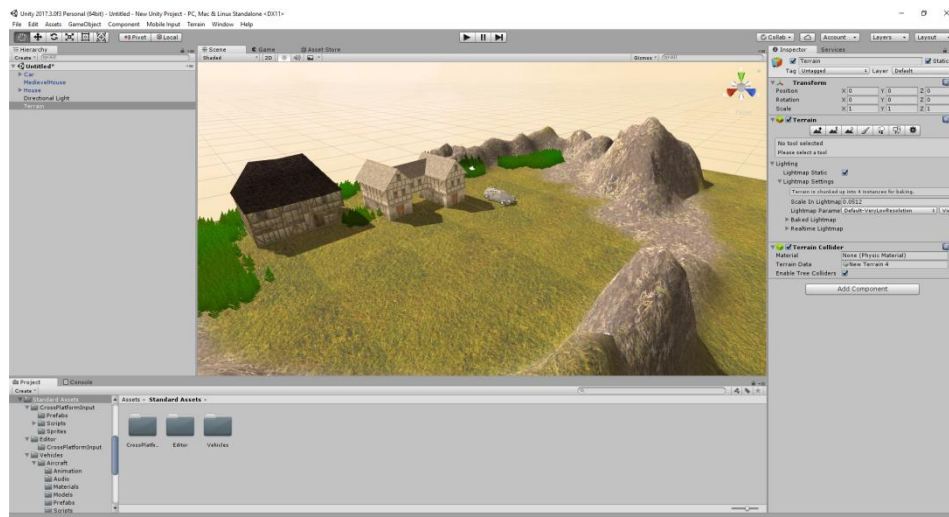


Рис. 7. Готовая локация в окне Scene

Чтобы проверить физическое взаимодействие, была создана стена из блоков. Каждому модулю присвоено физическое свойство с помощью компонента Rigidbody. Была произведена попытка разрушить её с помощью автомобиля.

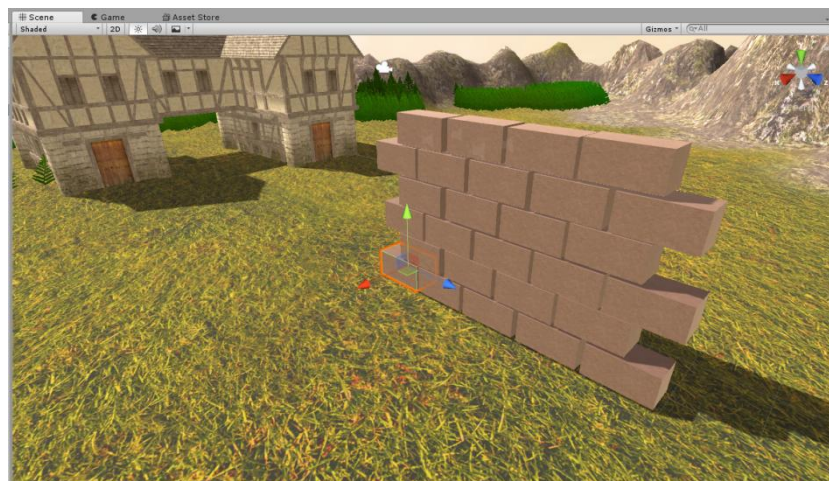


Рис. 8. Стена из модулей до столкновения

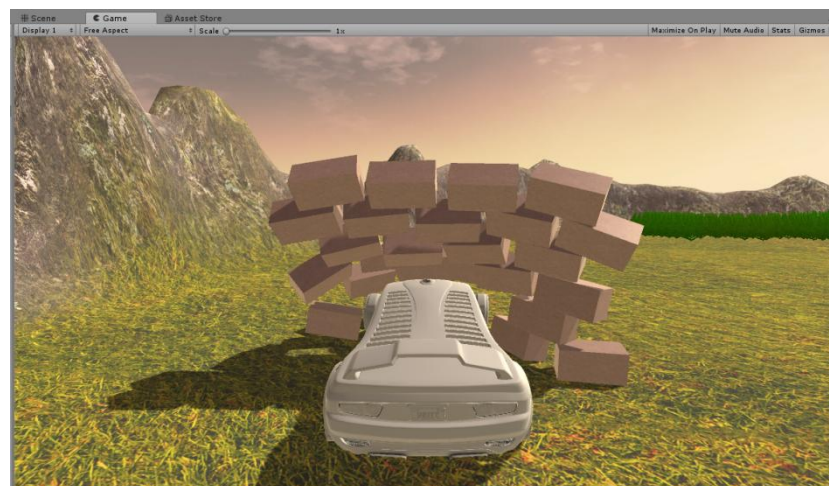


Рис. 9. Разрушение стены при столкновении автомобиля с ней

В результате столкновения тяжёлого автомобиля со стеной, состоящей из лёгких модулей стена моментально была разрушена. Unity полностью передал физическое взаимодействие нескольких предметов. Так же движок позволяет передать предмету упругость, что позволит предмету при его падении с высоты отскочить от поверхности, на которую он упадёт.

Физический движок является одним из эффективных способов, для создания игр и сцен с разрушениями объектов. Создатели движков постоянно работают над улучшением и добавляют новые технологии в свои продукты.

Библиографический список

1. Баженов Р.И., Ходос О.С. Лабораторный практикум по трехмерному моделированию в Unity3D // Биробиджан: Изд-во ФГБОУ ВПО «ПГУ им. Шолом-Алейхема», 2014. 143 с.
2. Свириденко А.Б., Гладкий Б.А. Unity3d – обзор возможностей и преимуществ использования в игровой индустрии // Научно-практическая конференция. 2014. Краснодар: Академия знаний, 2014. с. 211-214.
3. Goldstone W. Unity Game Development Essentials. Packt Publishing, 2009.
4. Винокуров С.А., Баженов Р.И. Создание проекта «Танк на острове» в Unity3d // Современная техника и технологии. 2015. № 7 (47). С. 53-59.
5. Кочитов М.Е., Лучанинов Д.В. Разработка игры «FlappyBird 3D» на Unity3D // Постулат. 2018. №1. URL: <http://e-postulat.ru/index.php/Postulat/article/view/1089>
6. Havok Physics URL: <https://www.havok.com/> (дата обращения 31.01.18)
7. Havok // Wikipedia URL: <https://ru.wikipedia.org/wiki/Havok> (дата обращения 31.01.18)
8. PhysX URL: <http://www.nvidia.ru/object/nvidia-physx-ru.html>
9. PhysX // Wikipedia URL: <https://ru.wikipedia.org/wiki/PhysX>
- 10.Unity URL: <https://unity3d.com/ru> (дата обращения 31.01.18)
- 11.Unity // Wikipedia URL: <https://goo.gl/w6hST5> (дата обращения 31.01.18)
- 12.Firewatch // Wikipedia URL: <http://www.firewatchgame.com/> (дата обращения 31.01.18)