

## Реализация механики разрушения игровых объектов в Unity 3D

*Ульянов Егор Андреевич*

*Приамурский государственный университет имени Шолом-Алейхема*

*Студент*

### Аннотация

В данной статье рассматривается и описывается процесс реализации механики разрушения игровых объектов прикосновением с помощью программирования в Unity 3D. Работа механики осуществляется посредством стандартных средств среды разработки. Практическим результатом является созданная и протестированная механика.

**Ключевые слова:** Unity 3D, механика, разрушение, ассет

## Implementation of the mechanics of destruction of game objects in Unity 3D

*Ulianov Egor Andreevich*

*Sholom-Aleichem Priamursky State University*

*Student*

### Abstract

This article discusses and describes the process of implementing the mechanics of destroying game objects by touch using programming in Unity 3D. The mechanics work is carried out through standard development environment tools. The practical result is created and tested mechanics.

**Keywords:** Unity 3D, mechanics, destruction, asset

Игровые механики имеют важную цель, потому что определяют, как игрок будет взаимодействовать с игровым миром и другими игроками. Хорошие игровые механики обеспечивают увлекательный игровой процесс, который будет захватывать внимание игрока и давать чувство достижения цели. Кроме того, игровые механики могут помочь определить жанр игры и уникальные особенности. Так, игровые механики в шутерах могут включать в себя стрельбу из оружия и использование тактики на поле боя, в то время как игровые механики в RPG (ролевых играх) могут включать в себя систему развития персонажа и выбор действий в диалогах. Хорошие, продуманные игровые механики могут сделать игру незабываемой и интересной для игроков.

Цель данной статьи рассмотреть возможности игрового движка Unity 3D в реализации различных игровых механик на сцене.

И. А. Савин, О. В. Батенькина рассмотрели процесс написания скриптовых сценариев при разработке виртуального тренажера [1]. С. А. Суродин в своей статье представил сценарий углубленного изучения одного

из лучших движков, существующих на данный момент, для создания красивых 2D и 3D игр [2]. В своей работе Р. Ф. Гайнуллин, В. А. Захаров, Е. А. Аксенова изучили инструмент для разработки двух- и трёхмерных игр – Unity 3D[3]. К. В. Богданов, П. Р. Михеев, И. Н. Суворов в своей работе описали развитие игровых движков, а именно провели обзор от примитивной графики до высокоуровневых инструментариев [4].

Начинаем реализацию механики с создания нового 3D проекта см. рисунок 1.

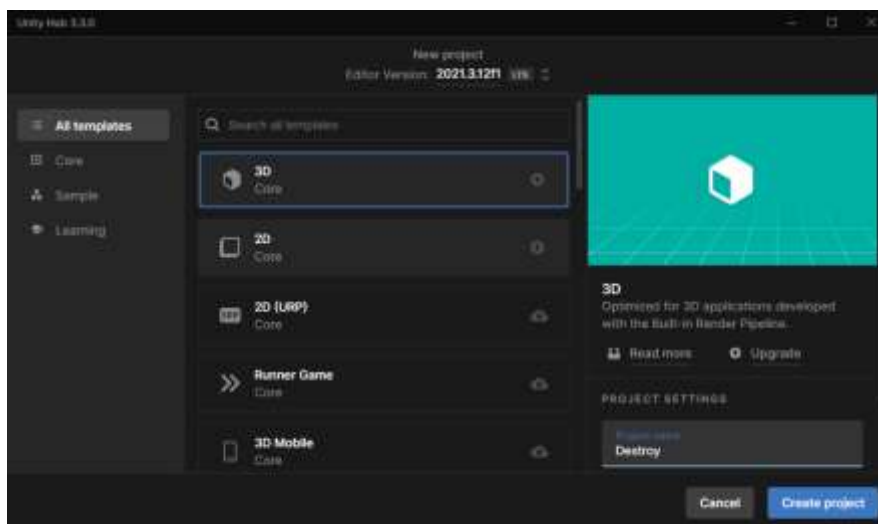


Рис. 1. Создание проекта

Для реализации механики необходимо подготовить ассеты. Нужен сам объект и разрушенная копия, поделенная на части, как на рисунках 2-3. Для примера возьмем такой ящик из открытых источников [5].



Рис. 2. Игровая заготовка ящика

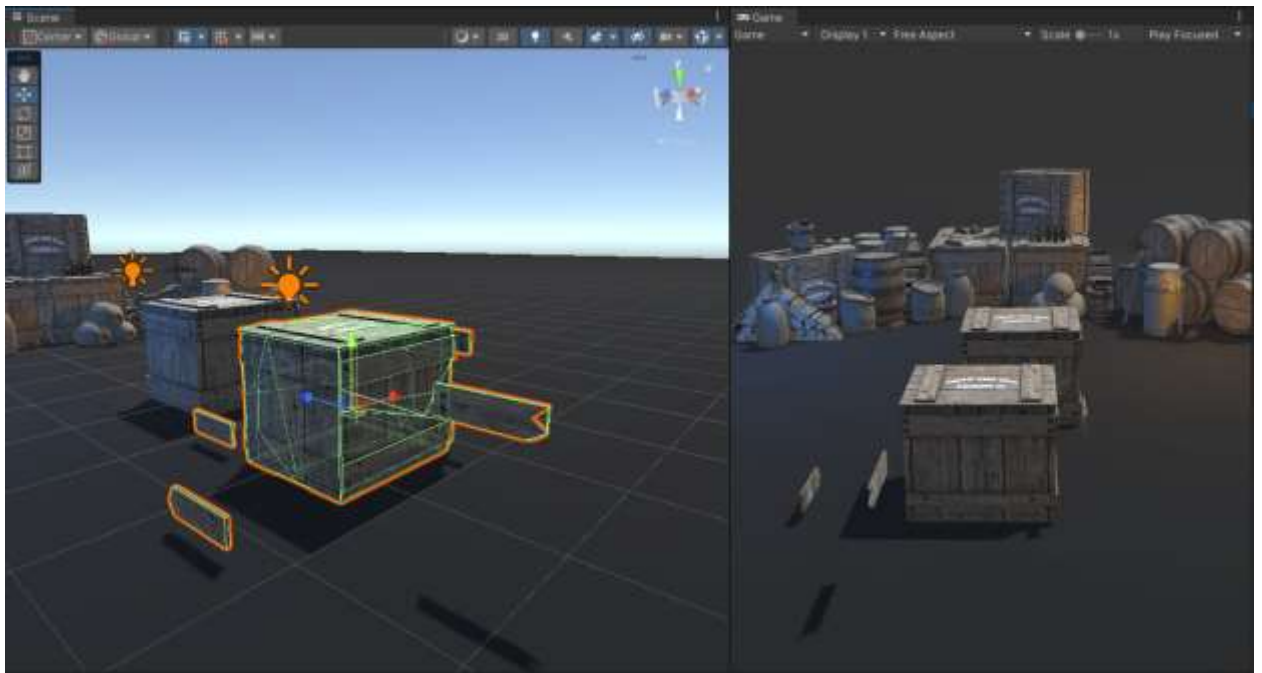


Рис. 3. Поделенный на части ящик

Обычному ящику добавляем компоненты необходимые для работы физики движка: «Box Collider», «Rigidbody» см. рисунок 4.

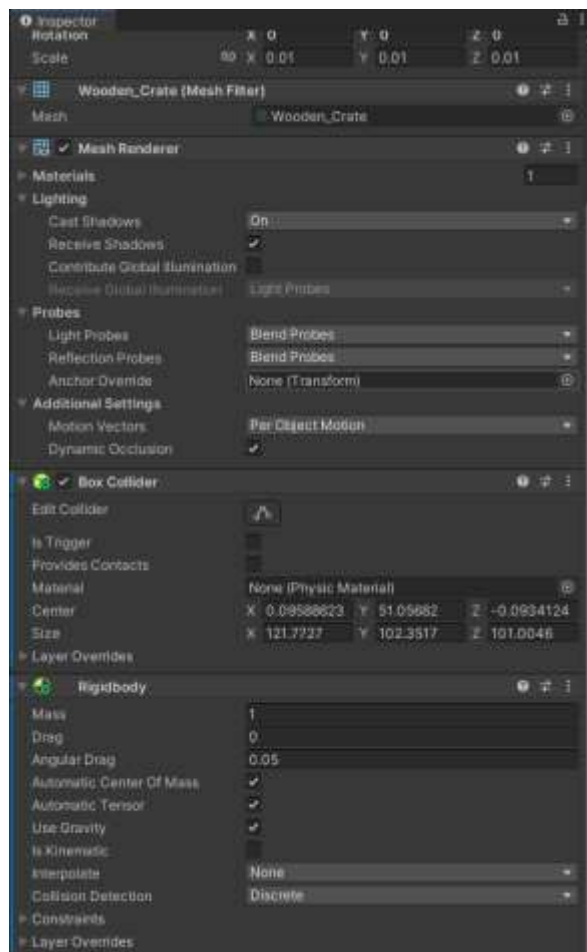


Рис. 4. Добавление компонентов

Для работы механики разрушения ящика всем частям разрушенного ящика добавляем компоненты «Mesh Collider» и «Rigidbody» рисунок 5-6.

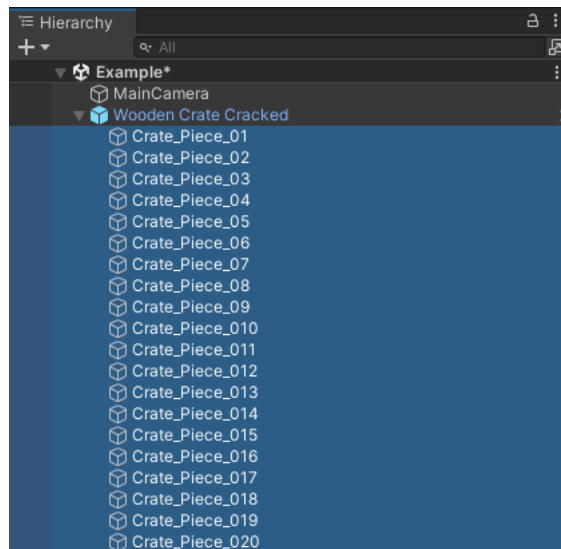


Рис. 5. Всем части ящика



Рис. 6. Добавление компонентов

Далее удаляем со сцены разрушенную копию ящика и добавляем к ящику скрипт «DestroyCube» см. рисунок 7.

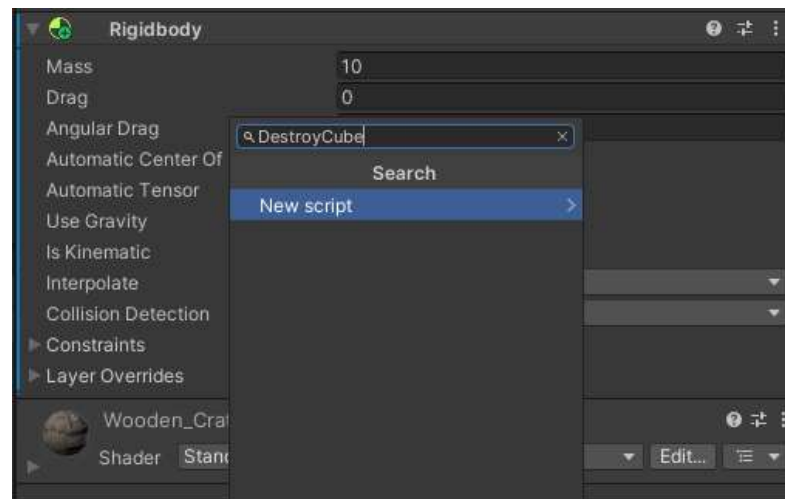


Рис. 7. Добавление скрипта

Переходим к написанию кода. Сначала объявляем публичную переменную «destroyedVersion» как «GameObject». Это добавляет возможность выбрать, какой объект использовать в качестве "разрушенной версии" куба в окне редактирования «Unity». Затем используем метод «OnMouseDown()», который срабатывает при нажатии кнопки мыши на объекте. Внутри этого метода используем «Instantiate()», чтобы создать экземпляр "разрушенной версии" на том же месте, где находится куб. Затем вызываем «Destroy()» на самом кубе, чтобы удалить из сцены. Таким образом, при клике на куб, объект исчезнет, а на этом же месте возникнет "разрушенная версия" см. рисунок 8.

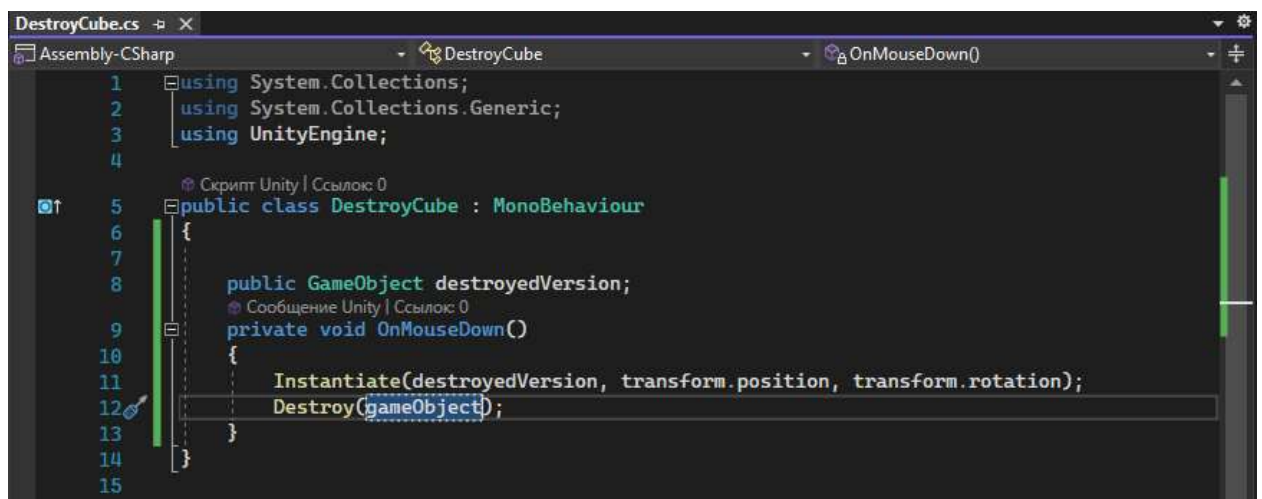


Рис. 8. Скрипт замены версий ящика

В компоненте скрипта добавляем разрушенную версию см. рисунок 9.

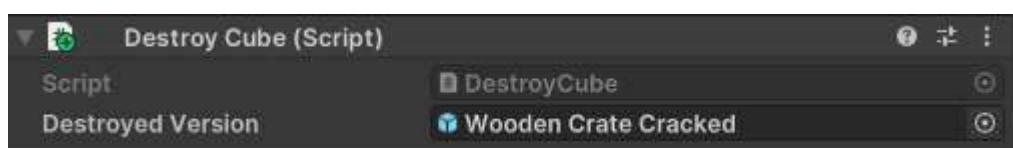


Рис. 9. Добавление разрушенной версии в скрипт

Проверяем работу скрипта и ассета см. рисунок 10.



Рис. 10. Демонстрация работы скрипта и механики

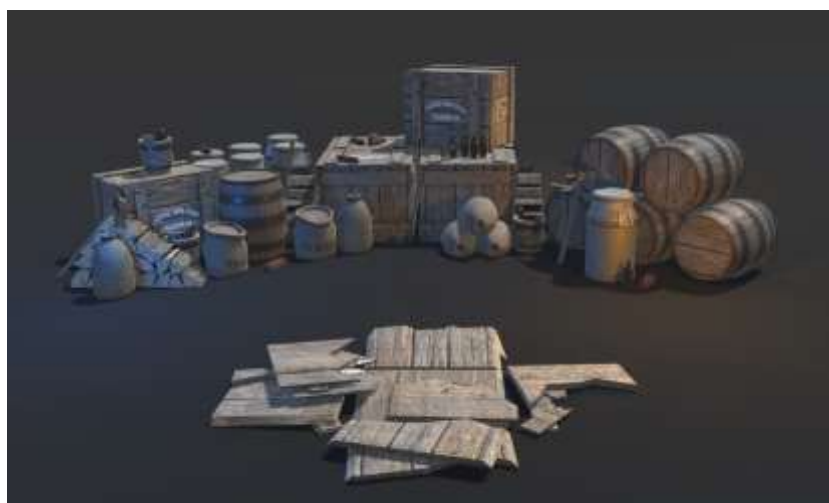


Рис. 11. Демонстрация работы скрипта и механики

Как показано на скриншотах механика разрушения игровых объектов работает в соответствии с указанными настройками.

### **Библиографический список**

1. Савин И. А., Батенькина О. В. Написание скриптов для трехмерного графического движка // Визуальная культура: дизайн, реклама, информационные технологии. 2018. № 12-7 (28). С. 7-15.
2. Суродин С. А. Unity 3D. разработка сценария проектирования в среде Unity 3D// Информатика и вычислительная техника. 2015. №3. С. 504-511.
3. Гайнуллин Р. Ф., Захаров В. А., Аксенова Е. А. Создание 2d игры на Unity 3D 5.4 // Вестник современных исследований. 2018. №4. С. 78-82.
4. Богданов К. В., Михеев П. Р., Суворов И. Н. Развитие игровых движков// Актуальные научные исследования в современном мире. 2021. №4. С. 24-29.
5. Devassets URL: <https://devassets.com/assets/western-props-pack/> (Дата обращения 03.07.2023)