

## Внедрение системы разрушения объектов в Unity 3D

*Бородулин Андрей Вадимович*

*Приамурский государственный университет им. 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 object destruction system in Unity 3D

*Borodulin Andrey Vadimovich*

*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

## 1. Введение

### Актуальность исследования

Внедрение системы разрушения объектов в Unity 3D значительно увеличивает реализм и интерактивность игровых проектов, позволяя игрокам взаимодействовать с окружающей средой на новом уровне. Эта статья исследует основные методы и инструменты, используемые для реализации механики разрушения, а также их влияние на игровой процесс. Понимание этих аспектов поможет разработчикам создавать более увлекательные и динамичные игры.

Актуальность исследования внедрения системы разрушения объектов в Unity 3D обоснована растущими требованиями к реалистичности и интерактивности в современных видеоиграх. Эффективная механика разрушения не только улучшает пользовательский опыт, но и открывает новые возможности для геймдизайнеров в создании динамичного игрового процесса. Понимание и применение таких систем становится ключевым

фактором для успешной разработки игр, соответствующих современным стандартам индустрии.

### Обзор исследований

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].

### Цель исследования

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

## 2. Методы исследования

В работе был применен аналитический метод: были проанализированы материалы и обучающие ресурсы по работе с Unity 3D.

## 3. Результат

Начинаем реализацию механики с создания нового 3D проекта (Рис. 1).

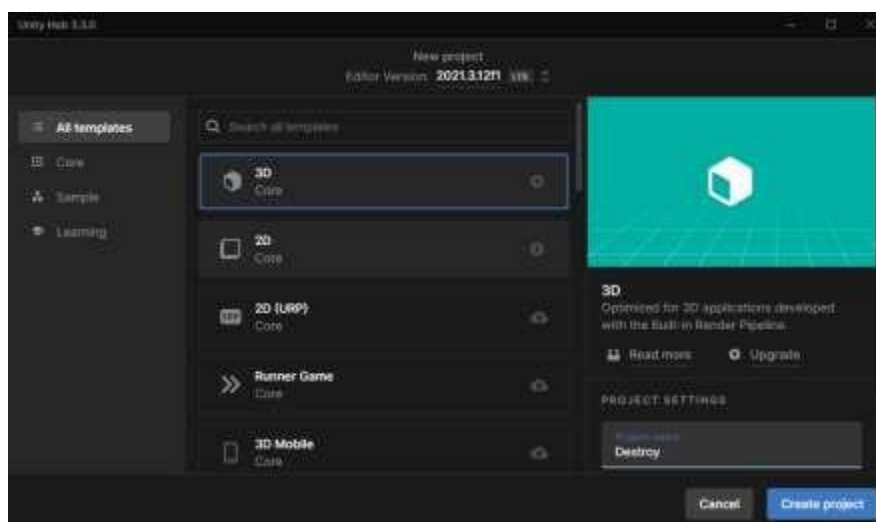


Рисунок 1 - Создание проекта

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



Рисунок 2 - Игровая заготовка ящика

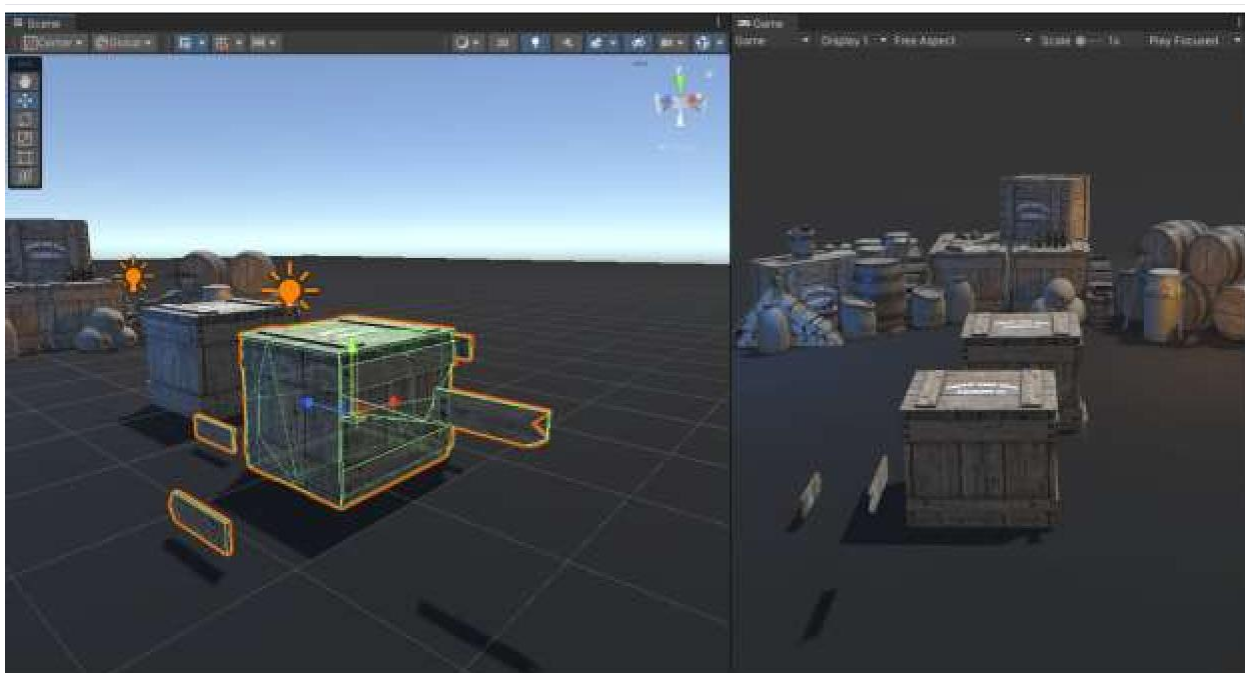


Рисунок 3 - Поделенный на части ящик

Обычному ящику добавляем компоненты необходимые для работы физики движка: «Box Collider», «Rigidbody» (Рис. 4).

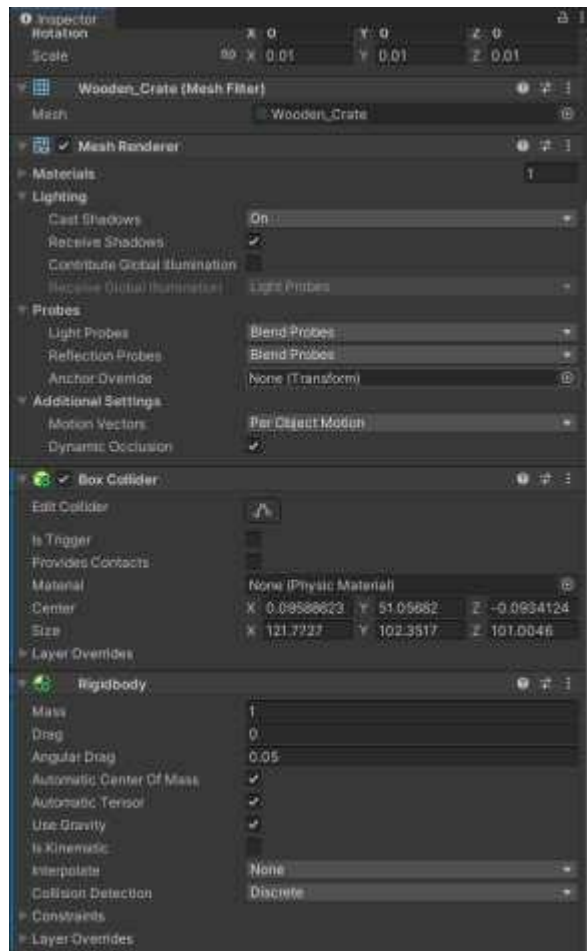


Рисунок 4 - Добавление компонентов

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

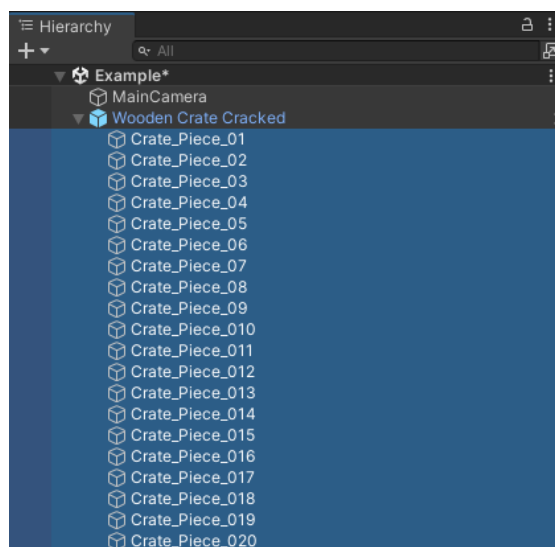


Рисунок 5 - Все части ящика



Рисунок 6 - Добавление компонентов

Далее удаляем со сцены разрушенную копию ящика и добавляем к ящику скрипт «DestroyCube» (Рис.7).

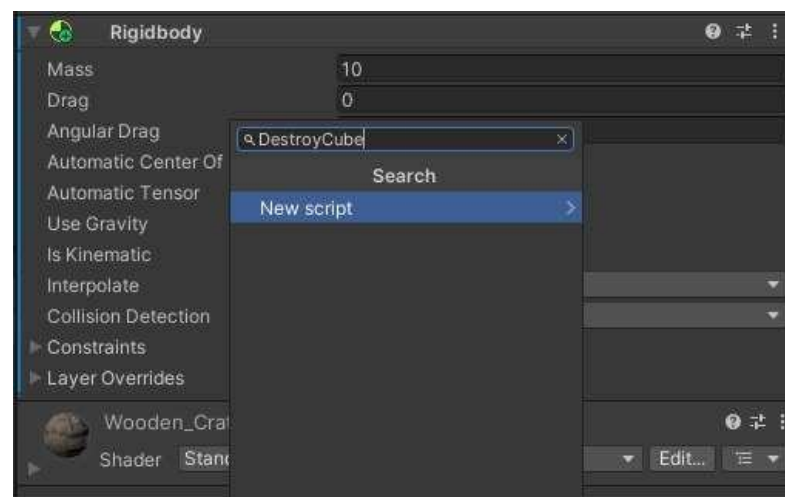
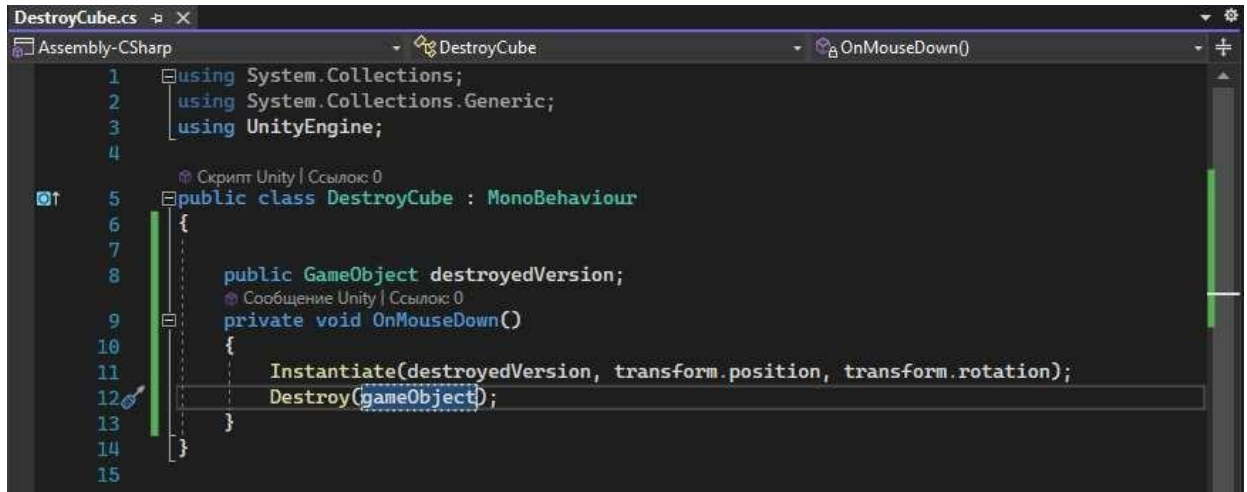


Рисунок 7 - Добавление скрипта

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" (Рис. 8).



```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class DestroyCube : MonoBehaviour
6 {
7
8     public GameObject destroyedVersion;
9     private void OnMouseDown()
10    {
11        Instantiate(destroyedVersion, transform.position, transform.rotation);
12        Destroy(gameObject);
13    }
14 }
15
```

Рисунок 8 - Скрипт замены версий ящика

В компоненте скрипта добавляем разрушенную версию (Рис. 9).



Рисунок 9 - Добавление разрушенной версии в скрипт

Проверяем работу скрипта и ассета (Рис. 10-11).

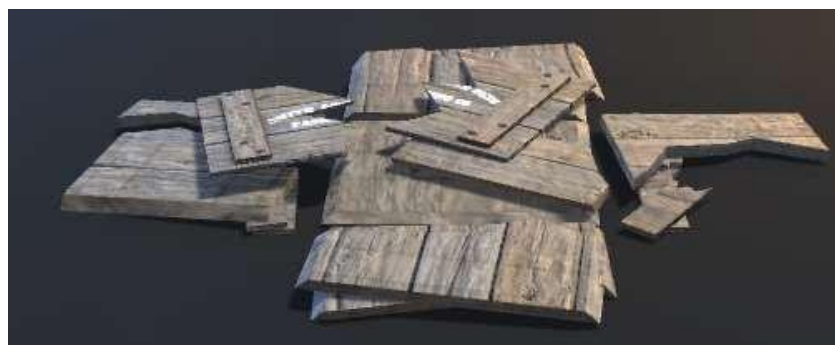


Рисунок 10 - Демонстрация работы скрипта и механики



Рисунок 11 - Демонстрация работы скрипта и механики

#### **4. Выводы**

В данной статье показан процесс настройки и использования механики разрушения объектов, которая работает в соответствии с 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 03.09.2024)