

Реализация простого, случайного генератора объектов на уровнях в Unity 3D

Ульянов Егор Андреевич

*Приамурский государственный университет имени Шолом-Алейхема
Студент*

Аннотация

В данной статье рассматривается и описывается реализация простого, случайного генератора объектов на уровнях при помощи стандартных средств Unity. Генерация осуществляется посредством специально написанного скрипта на языке C#. Практическим результатом является случайно сгенерированные объекты на уровне.

Ключевые слова: Unity 3D, случайный генератор объектов, скрипт

Implementation of a simple, random object generator at levels in Unity 3D

Ulianov Egor Andreevich

*Sholom-Aleichem Priamursky State University
Student*

Abstract

This article discusses and describes the implementation of a simple, random object generator at levels using standard Unity tools. The generation is carried out by means of a specially written script in the C# language. The practical result is randomly generated objects at the level.

Keywords: Unity 3D, random object generator, script

Для повышения реиграбельности компьютерных и мобильных игр многие разработчики стремятся добавить элемент случайности, делая каждый перезапуск игры более увлекательным и уникальным. Красивый и правильно заполненный уровень дает игроку почувствовать атмосферу, представить себя на месте главного героя.

Цель данной статьи рассмотреть возможности игрового движка Unity 3D в создании случайного генератора объектов на сцене.

И.А. Савин, О.В. Батенькина в своей статье рассмотрели процессы написания скриптовых сценариев при разработке виртуального тренажера «Учебно-производственные мастерские кафедры «Гидромеханика и транспортные машины» ОмГТУ[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 создание генератора с создания проекта, а также интеграции скачанных 2D моделей в проект[5] см. рисунок 1-2.

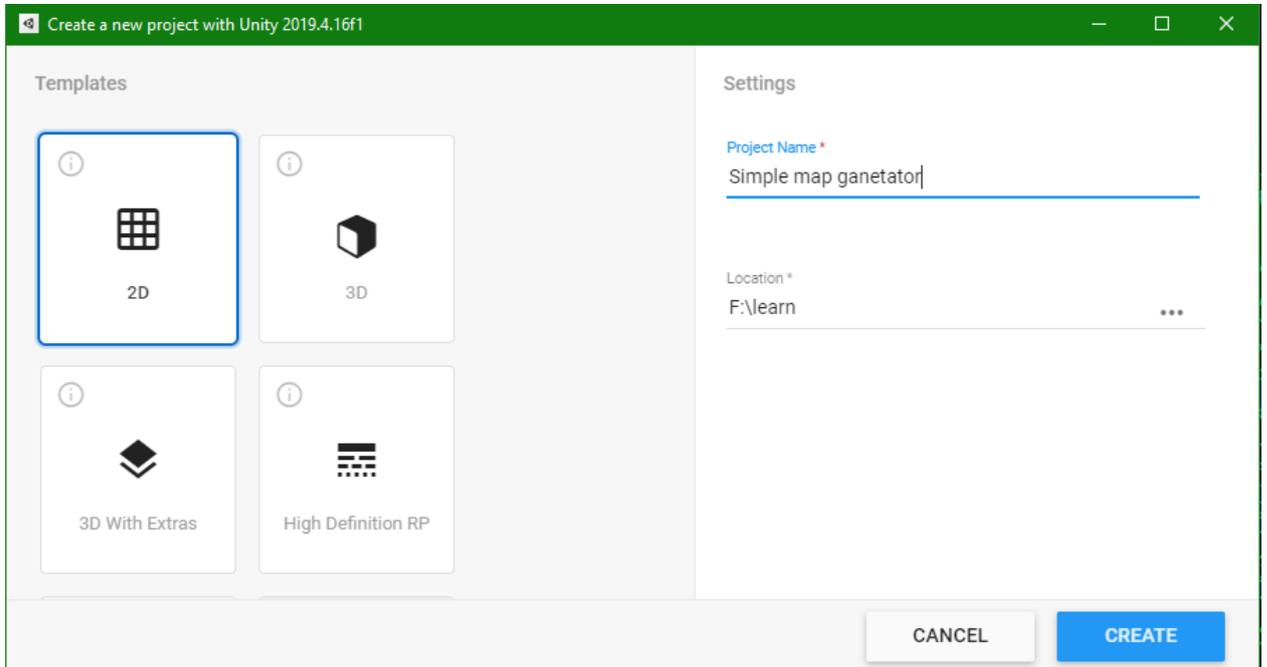


Рис. 1. Создание проекта

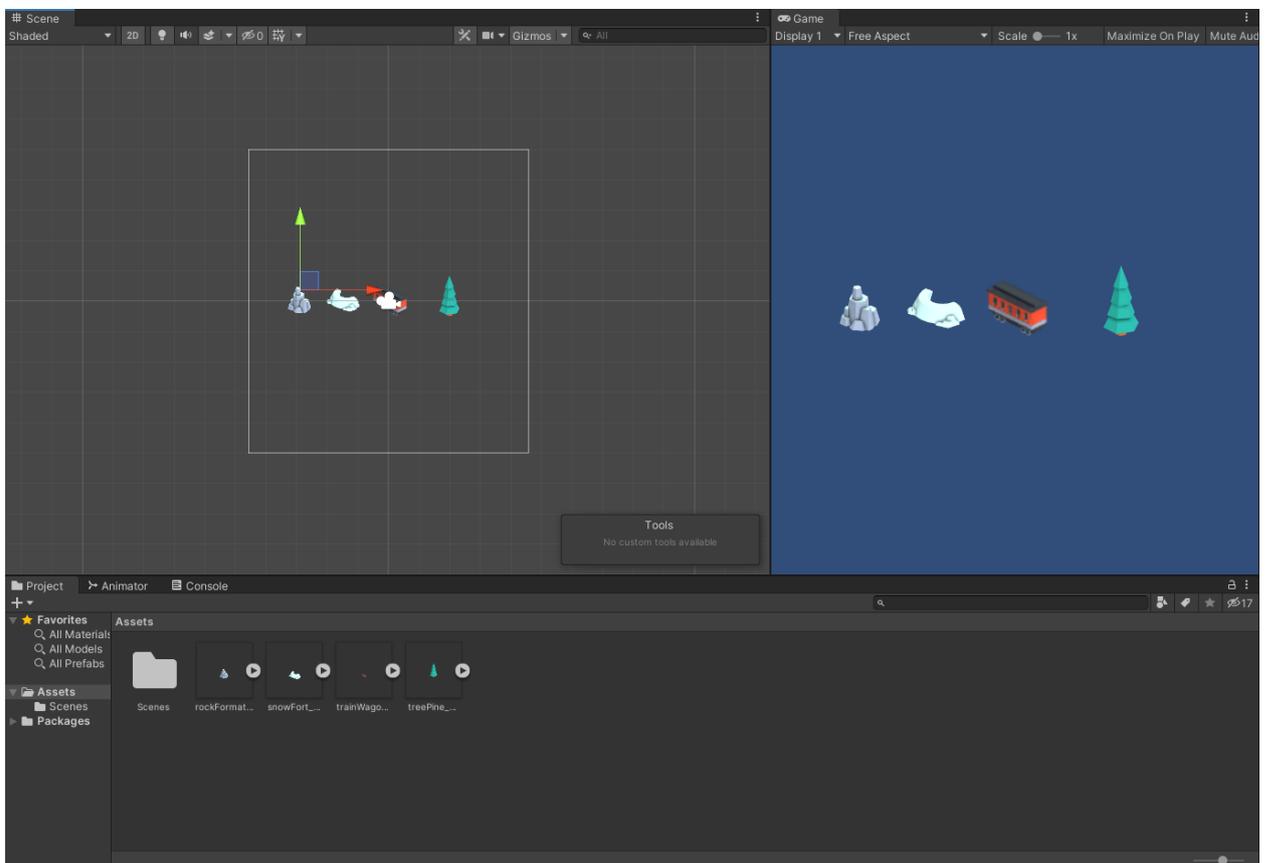


Рис. 2. Импорт моделей в проект

Далее необходимо создать префабы (префаб - это особый тип ассетов, позволяющий хранить весь GameObject со всеми компонентами и значениями свойств. Префаб выступает в роли шаблона для создания экземпляров хранимого объекта в сцене. Любые изменения в префабе немедленно отражаются и на всех экземплярах, при этом возможно переопределять компоненты и настройки для каждого экземпляра в отдельности) этих объектов, для этого создаем новую папку и называем ее «prefabs», перетаскиваем все спрайты в папку и удаляем со сцены. В окне «inspector» должен появиться синий куб см. рисунок 3.

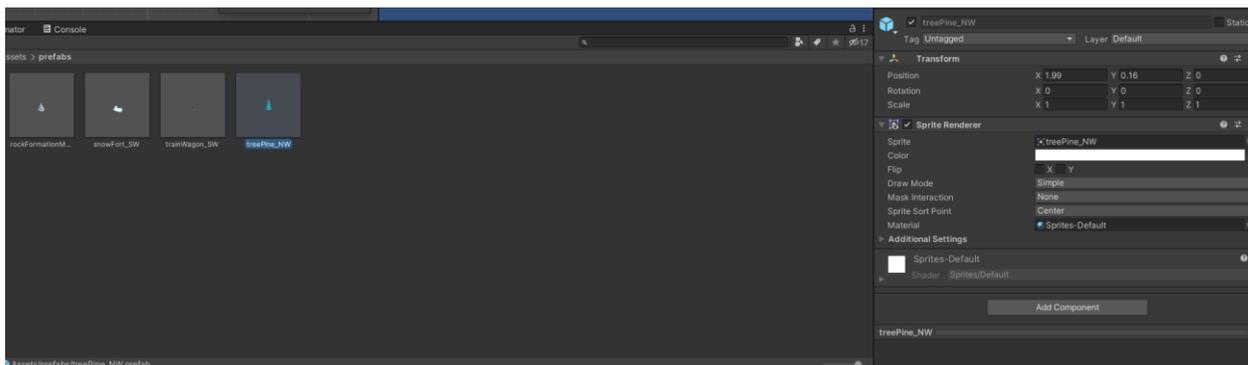


Рис. 3. Создание префабов спрайтов

Далее создаем два «Empty Game Object» называем «LevelTemplate01» (выбор объекта) и «SpawnPoint» (точка появления) соответственно, меняем значки для облегчения поиска на сцене. «SpawnPoint» расставляем в случайном порядке см. рисунок 4.

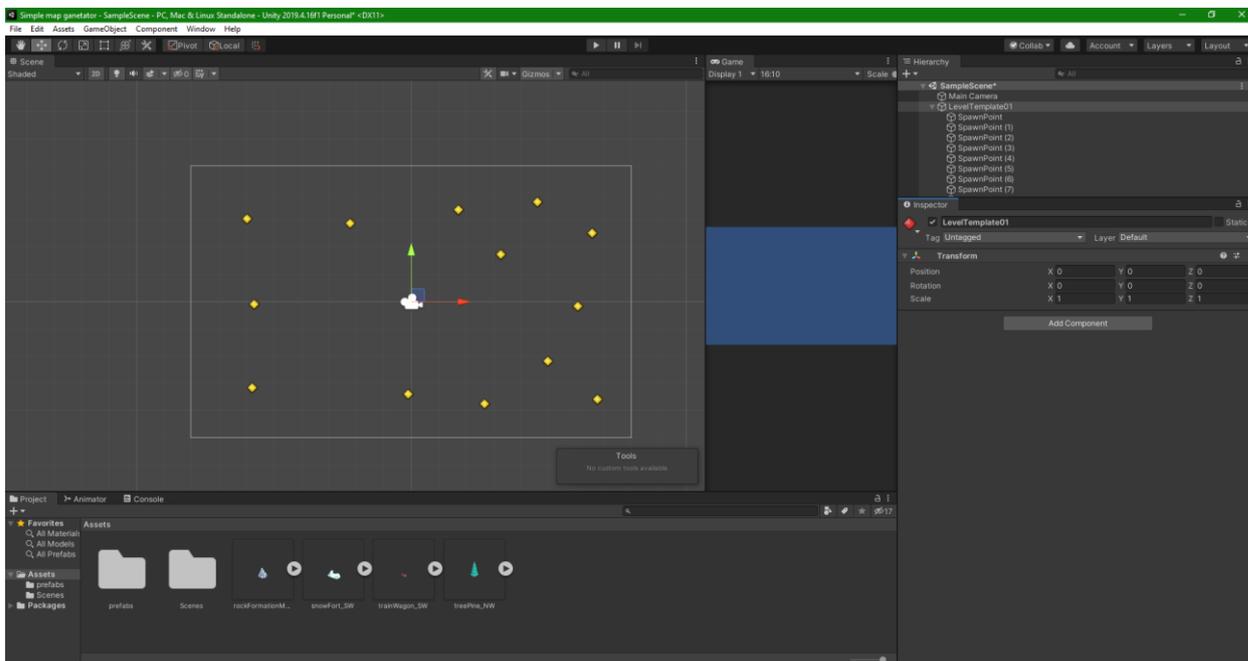


Рис. 4. Создание шаблона генератора

Приступаем к написанию скрипта, создаем папку для скриптов и создаем скрипт с названием «LevelGeneration» см. рисунок 5.

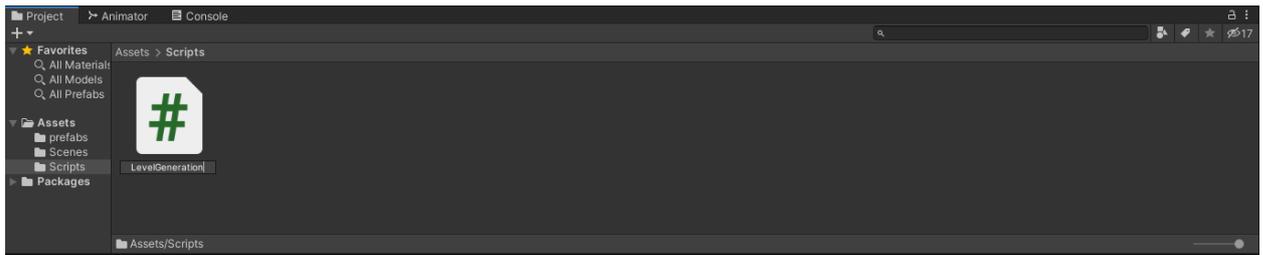


Рис. 5. Создание скрипта

Начнем с создания массива игровых объектов, который будет хранить префабы спрайтов. Далее в стартовой функции, создаем случайный интервал с именем «rand» и приравниваем к случайному числу от 0 до суммы загруженных префабов. Осталось только указать программе какие объекты создавать и в каком месте, см. рисунок 6.

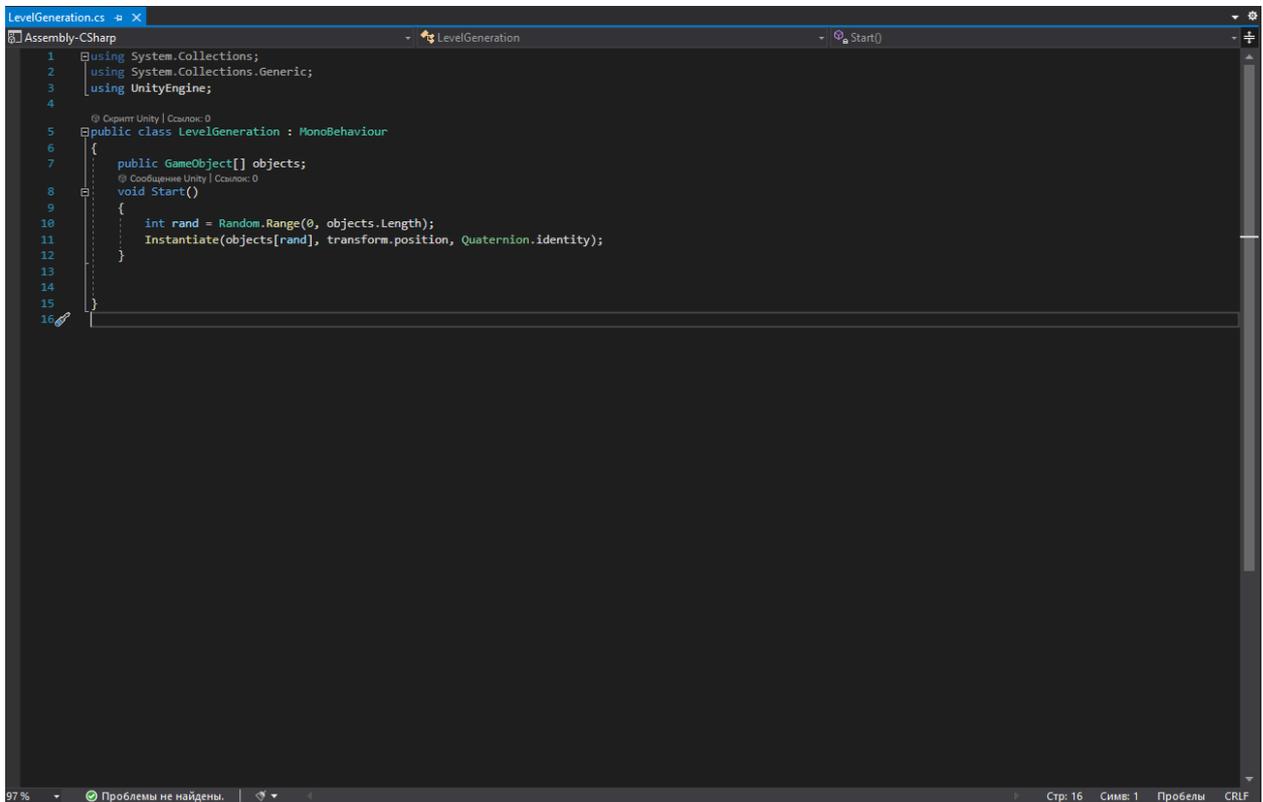


Рис. 6. Код генератора

С зажатой клавишей «Shift» выбираем все точки появления и посредством «Drag-and-drop» добавляем скрипт см. рисунок 7.

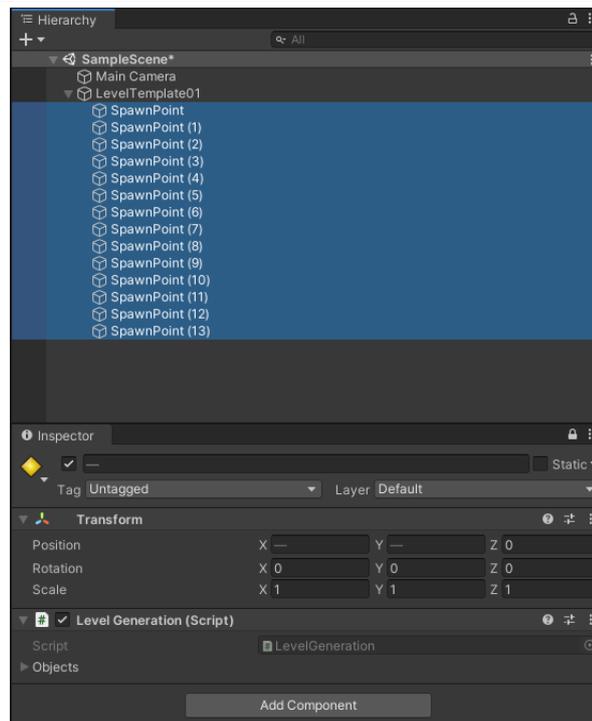


Рис. 7. Добавление скрипта к точкам появления

Далее выделяем все префабы и посредством того же «Drag-and-drop» добавляем в объекты скрипта см. рисунок 8.

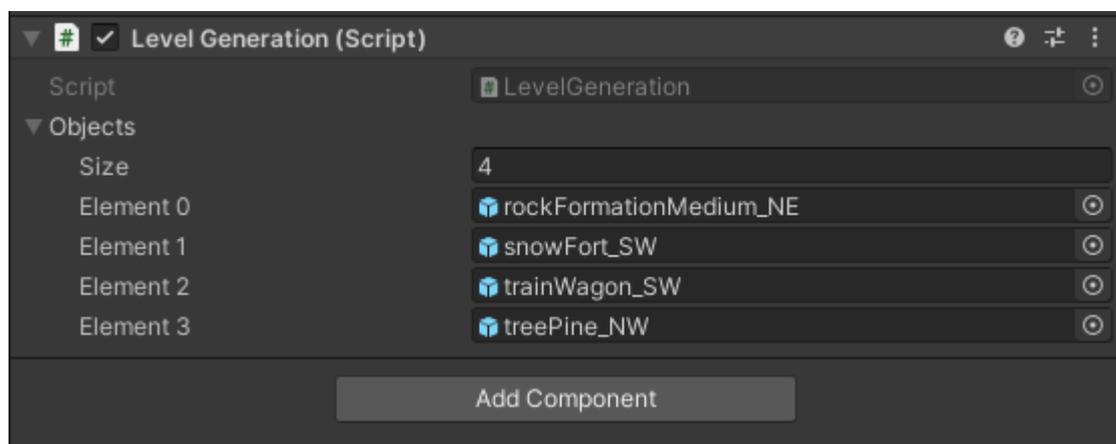


Рис. 8. Добавление префабов в скрипт

Теперь можно нажать на клавишу «Play» и убедиться в работе скрипта см. рисунок 9-10.

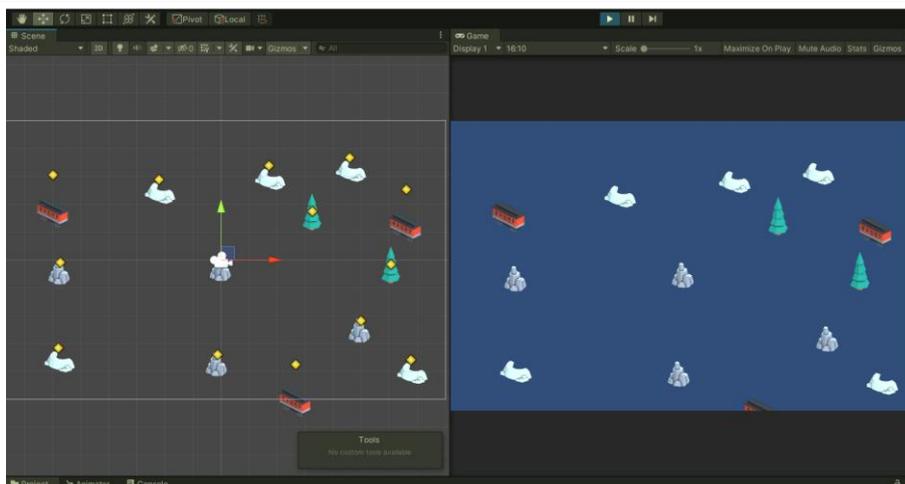


Рис. 9. Демонстрации работы скрипта

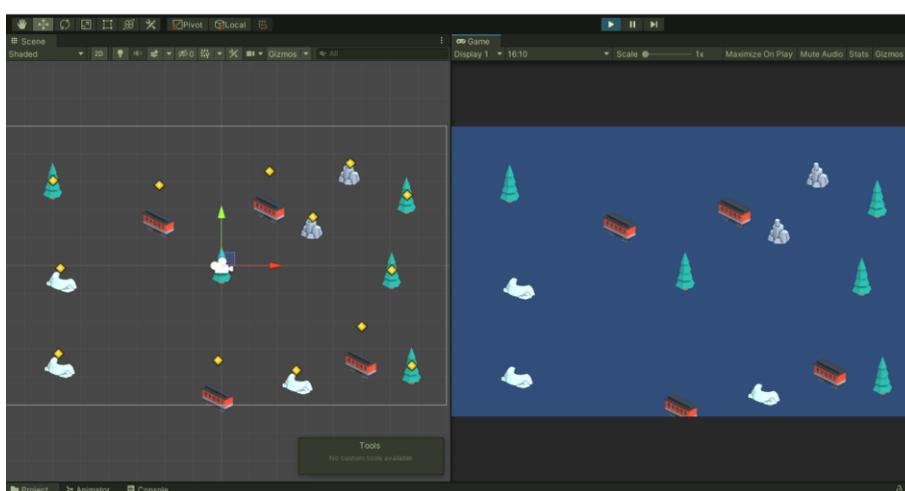


Рис. 10. Демонстрации работы скрипта

Как показано на скриншотах случайная генерация объектов на сцене работает, и зависит только от количества добавленных объектов, а также от расположение точек появления. В данной статье был разработан простой генератор случайных объектов на сцене.

Библиографический список

1. Савин И.А., Батенькина О.В. Написание скриптов для трехмерного графического движка//Визуальная культура: дизайн, реклама, информационные технологии. 2014. №5. С. 91-95.
2. Сурудин 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. <https://kenney.nl/assets/holiday-kit>