Министерство науки и высшего образования Российской Федерации

федеральное государственное бюджетное образовательное учреждение высшего образования

«Российский экономический университет имени Г.В. Плеханова»

**МОСКОВСКИЙ ПРИБОРОСТРОИТЕЛЬНЫЙ ТЕХНИКУМ.**

Специальность: 09.02.07 Информационные системы и программирование

Квалификация: Программист

ОТЧЕТ

ПО ОБЩЕПРОФЕССИОНАЛЬНОЙ ДИСЦИПЛИНЕ: МДК 01.01 РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ

Тема: создание палитр и процедурных генераций

Листов: 14

|  |  |
| --- | --- |
| Выполнил студент  Группы П50-7-20  Антонова М.М.  «\_\_\_\_» \_\_\_\_\_\_\_\_\_ 2023 года | Проверил преподаватель  \_\_\_\_\_\_\_\_\_\_\_А.Ю.Бушин  «\_\_\_\_» \_\_\_\_\_\_\_2023 года |

Москва 2023

ПРАКТИЧЕСКАЯ РАБОТА №15.

Цель работы: при помощи базовых знаний интерфейса Unity и языка программирования С#, скачать инструменты для Android разработки, создать несколько платформ при помощи палитр и разработать скрипт для процедурной генерации.

1. Разархивируем архив (не убираем никакие галочки)



Рисунок 1. Файл для разархивации (не убирая галочек соглашаемся на операцию)

1. Открываем новое окно генерации ландшафта игры

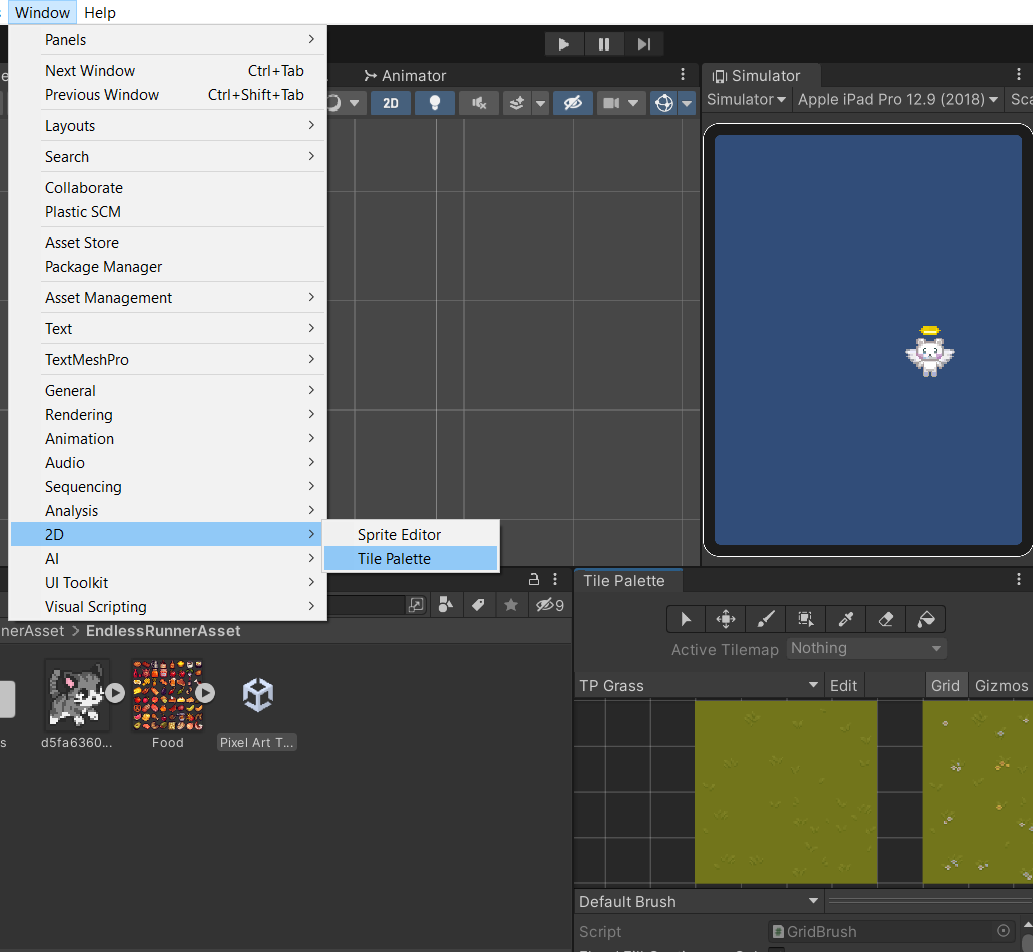


Рисунок 2. Открываем окно

1. Создаем пустой объект, сбрасываем его координаты



Рисунок 3. Создаем объект и называем его



Рисунок 4. Сбрасываем координаты объекта

1. Создаем по цепочке новые объекты (4 штуки: Ground, Wall, Background, Props)

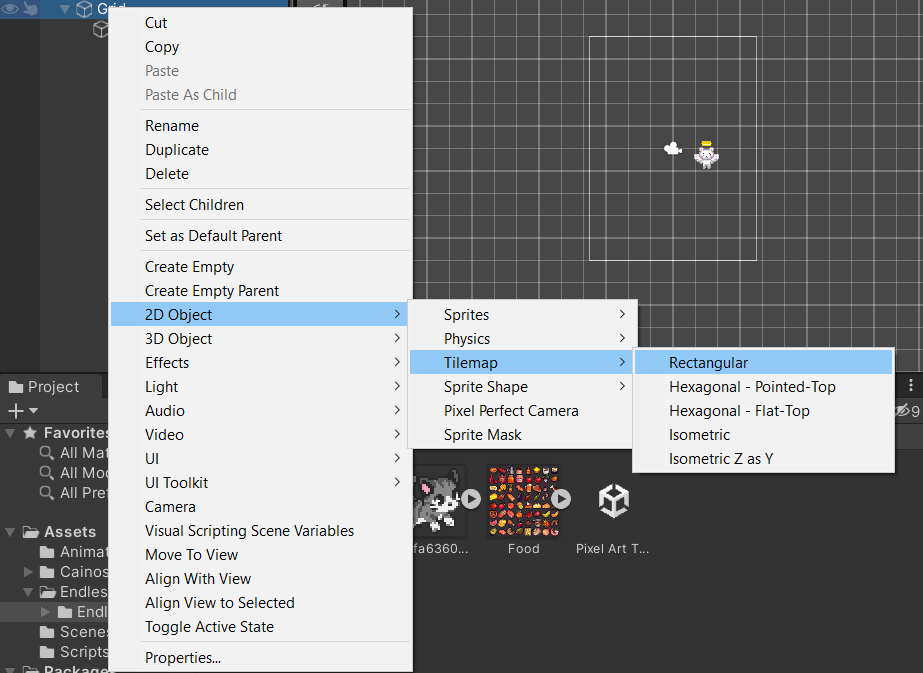


Рисунок 5. Путь для создания слоя



Рисунок 6. Все слои должны находиться в объекте Platform1

1. Настраиваем (Ground 1, wall 2, background 0, props 3)

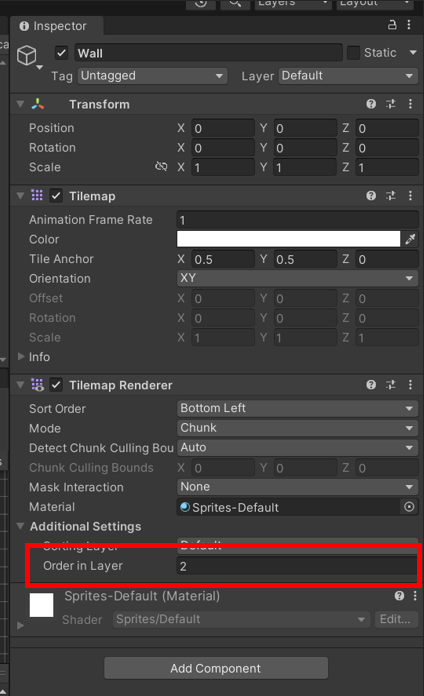


Рисунок 7. Меняем значение (эти значения будут определять порядок слоев)

1. В отдельном окне есть примеры текстур, берем и рисуем

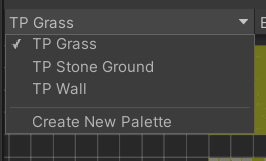


Рисунок 8. Выбираем какой слой будем рисовать (землю, стены и тд)

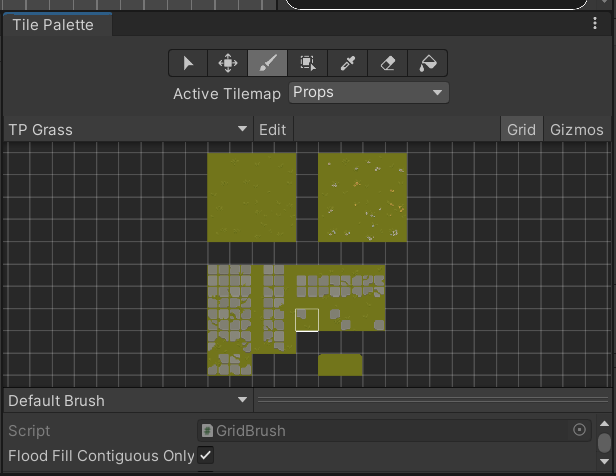


Рисунок 9. Так выглядит панелька с текстурами (выбрав третий инструмент сверху – кисточка, можно рисовать как по одной клеточке, так и всей площадью текстурки)

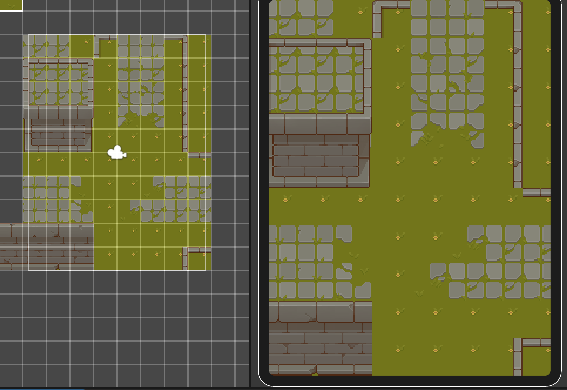


Рисунок 10. После рисования получается такая картина (на сцене есть ограничения размера эмулятор, для наглядности)

1. По пути идем к нужному компоненту, чтобы создать свою текстурку

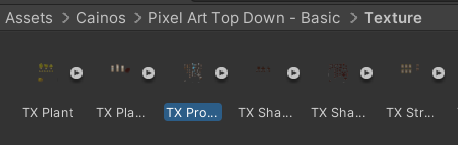


Рисунок 11. Находим по такому пути данный файл

1. Создаем новый объект, далее открываем объект в окне текстур и перетаскиваем туда файл с текстурными объектами, далее просто расставляем их на сцене

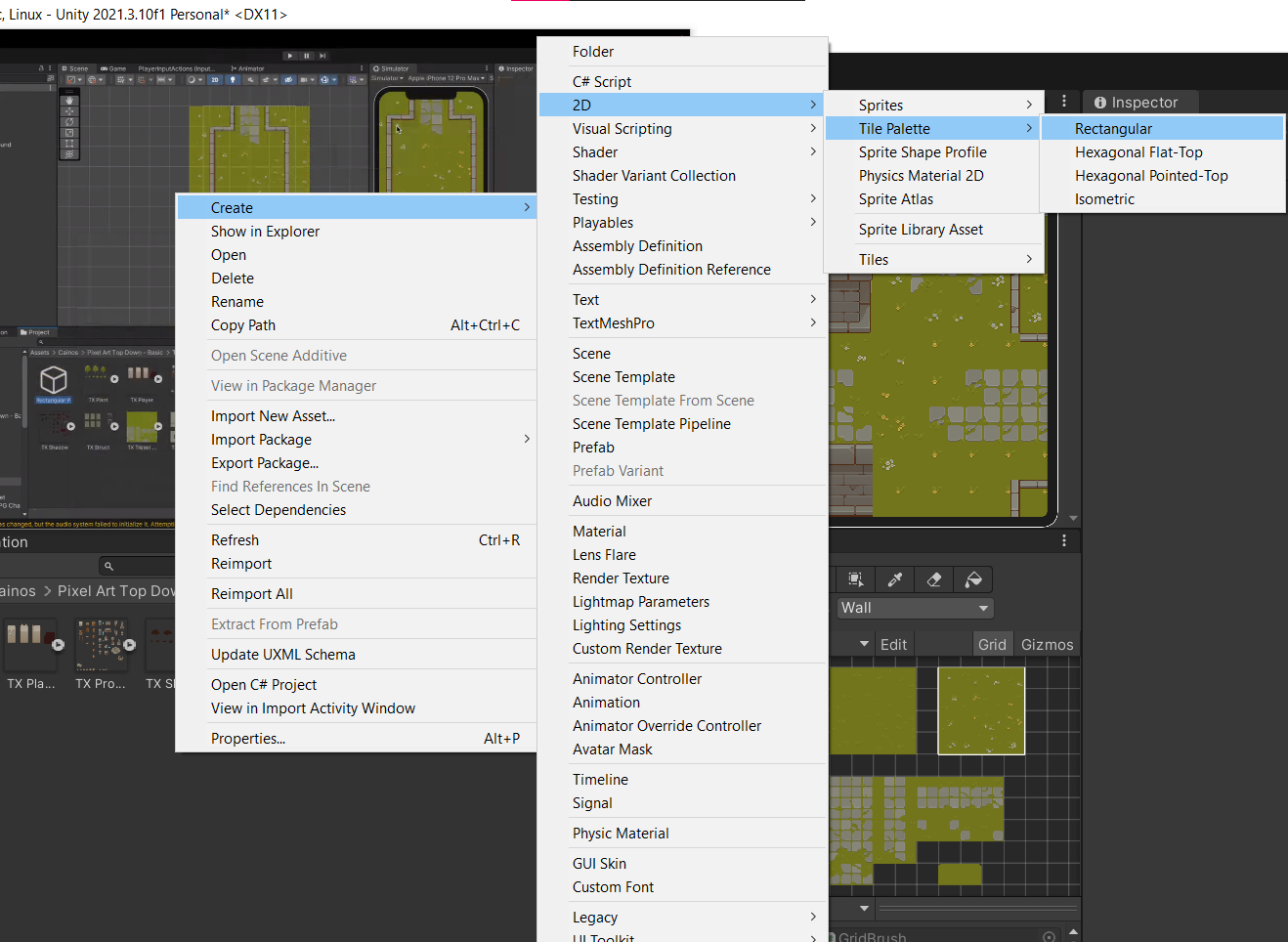


Рисунок 12. Создаем новый объект

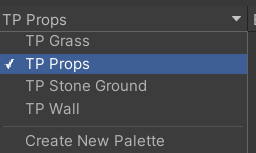


Рисунок 13. Выбираем объект, который только что создали

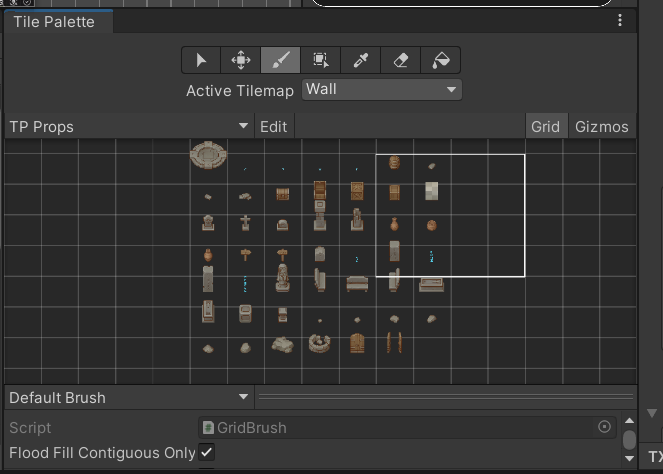


Рисунок 14. Перетаскиваем ранее найденные текстуры в окошко

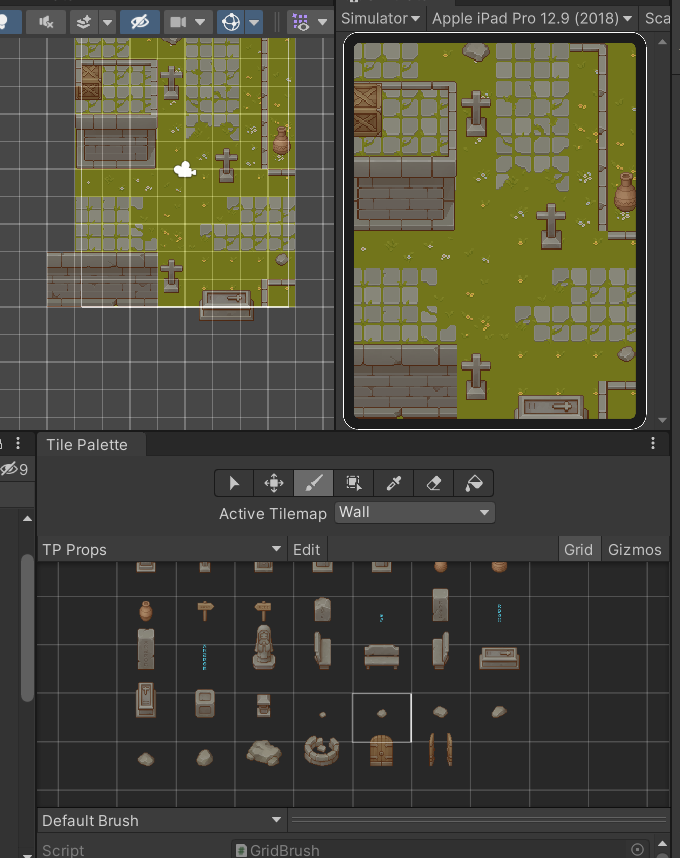


Рисунок 15. Добавляем объекты на сцену

1. Создаем пустой объект, в его настройках ставим ему отображение красной звездочкой и перетаскиваем его наверх сцены

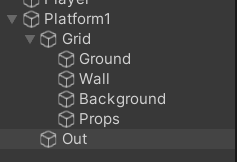


Рисунок 16. Создаем объект



Рисунок 17. Меняем его отображение

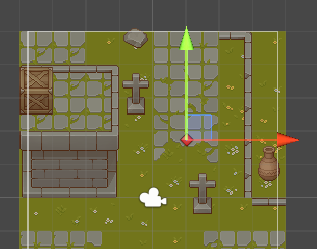


Рисунок 18. Отображение объекта на сцене



Рисунок 19. Перетаскиваем объект наверх сцены

1. Создаем новый тег для ранее созданного объекта

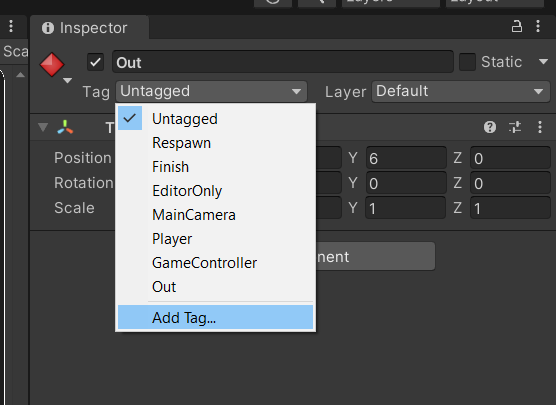


Рисунок 20. В настройках объекта создаем новый тег

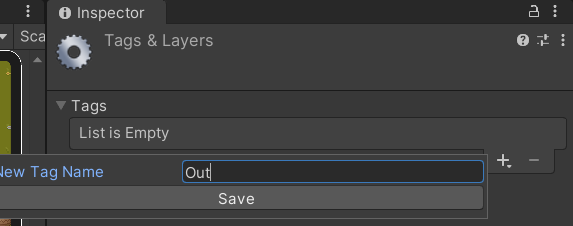


Рисунок 21. Создаем новый тег Out и сохраняем

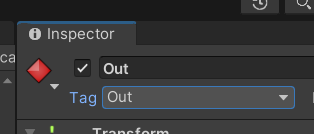


Рисунок 22. Ставим в настройках тег, который только что создали

1. Создаем новый пустой объект со слоями (по-аналогии с Platform1)

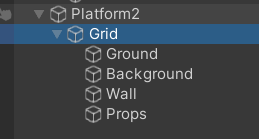


Рисунок 23. Создаем объекты

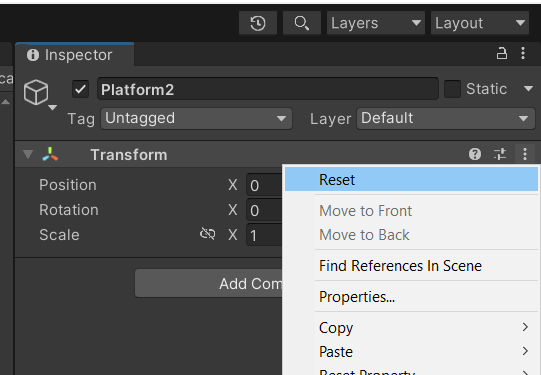


Рисунок 24. Сбрасываем настройки

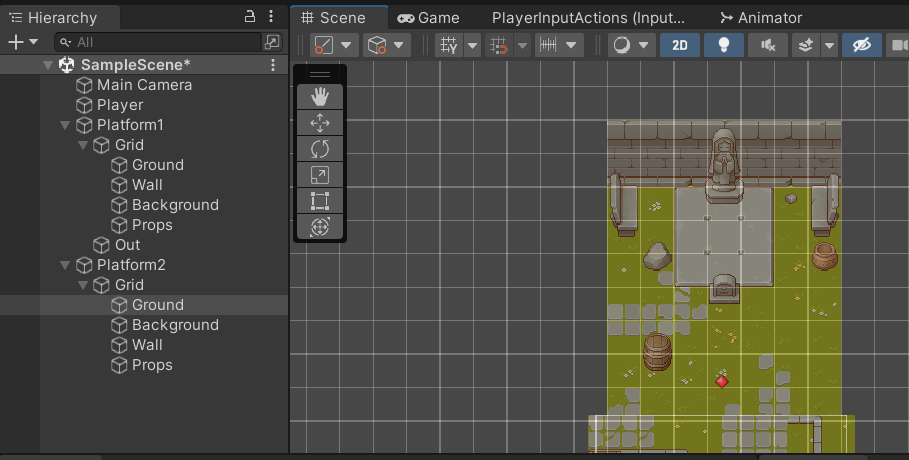


Рисунок 25. Рисуем вторую платформу

1. Для того, чтобы было видно котика, в настройках поставим ему 5 позицию в слоях (так уже делали с другими слоями)

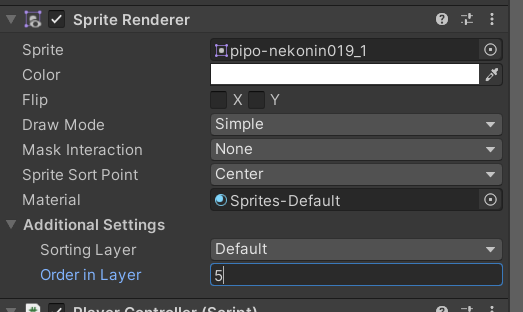


Рисунок 26. Меняем настройки

1. Далее в настройках сбрасываем расположение Platform2, чтобы она наложилась под Platform1, далее просто перетаскиваем Platform2 в Assets (важно! Сначала Reset, а потом префаб и платформы должны находиться четко друг под другом)

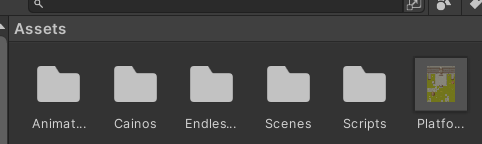


Рисунок 27. Создаем префаб

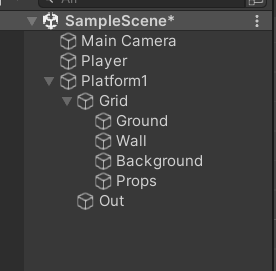


Рисунок 28. После префаба нет отображения второй платформы



Рисунок 29. Прописываем новый скрипт

1. Создаем пустой объект и помещаем на него скрипт



Рисунок 30. Пустой объект

1. Кидаем скрипт на пустой ранее созданный объект. В инспекторе в настройки скрипта кидаем префаб платформы

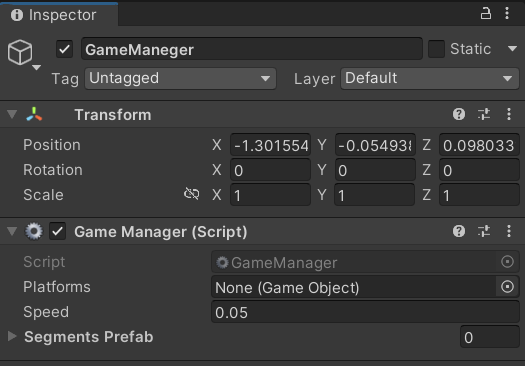


Рисунок 31. Кидаем скрипт на объект

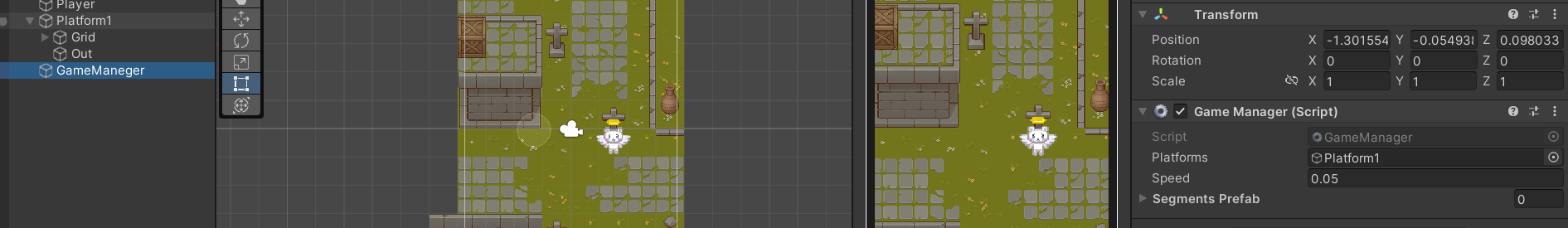


Рисунок 32. Перекидываем объект Platforms в скрипт

1. Открываем доп. настройки и кидаем туда префаб

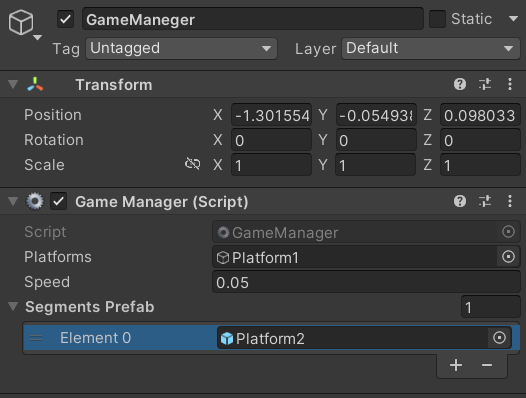


Рисунок 33. Кидаем префаб в Segments Prefab

1. Создаем новый тег Start и вещаем его на platform1

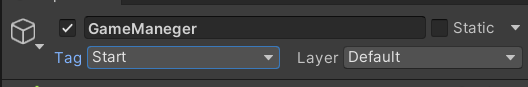


Рисунок 34. Ставим тег

1. Создаем новые платформы, добавляем на них теги, настраиваем и запускаем игру



Рисунок 35. Пример новой территории

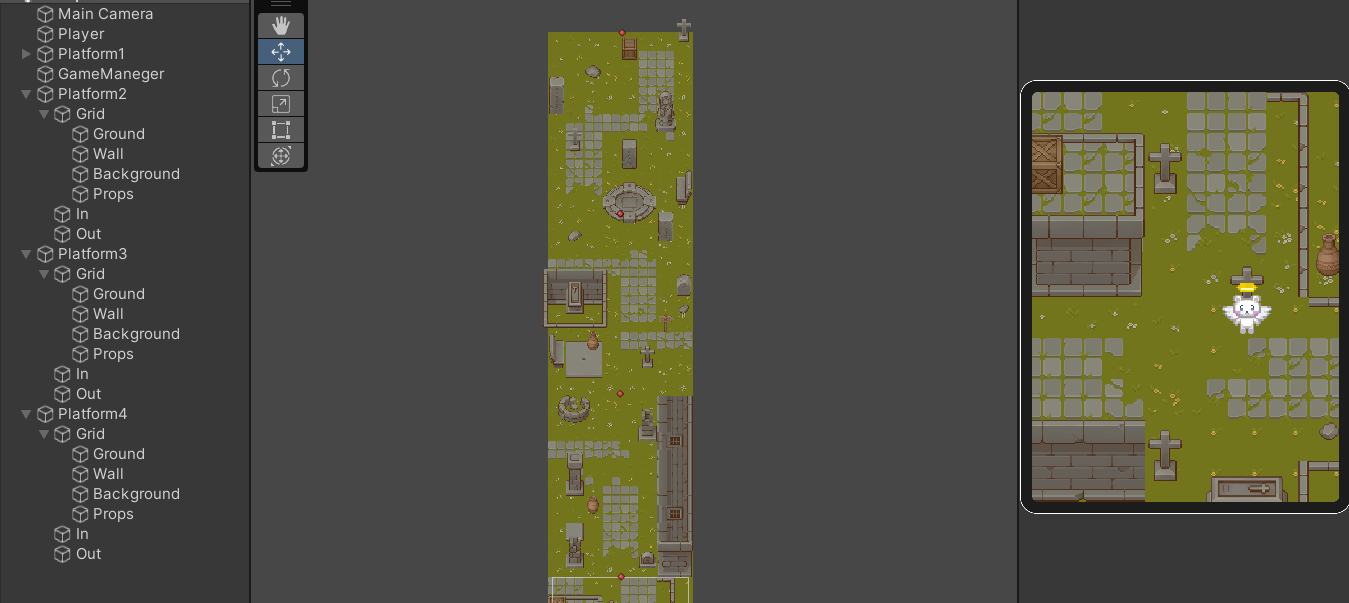


Рисунок 36. Также создаем теги и проделываем те же действия, что и с прошлой платформой

Вывод: при помощи базовых знаний интерфейса Unity и языка программирования С# и инструментов для Android разработки, создали несколько платформ при помощи палитр и написали скрипт для процедурной генерации.