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 шейдеров.

Листов: 15

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

Москва 2023

ПРАКТИЧЕСКАЯ РАБОТА №10.

Цель работы: при помощи базовых знаний интерфейса Unity и языка программирования С#, создание шейдеров.

1. Создаем папку (в будущем будем хранить в ней шейдеры)

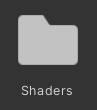


Рисунок 1. Папка Shaders

1. Создаем шейдер и называем его LavalampShader

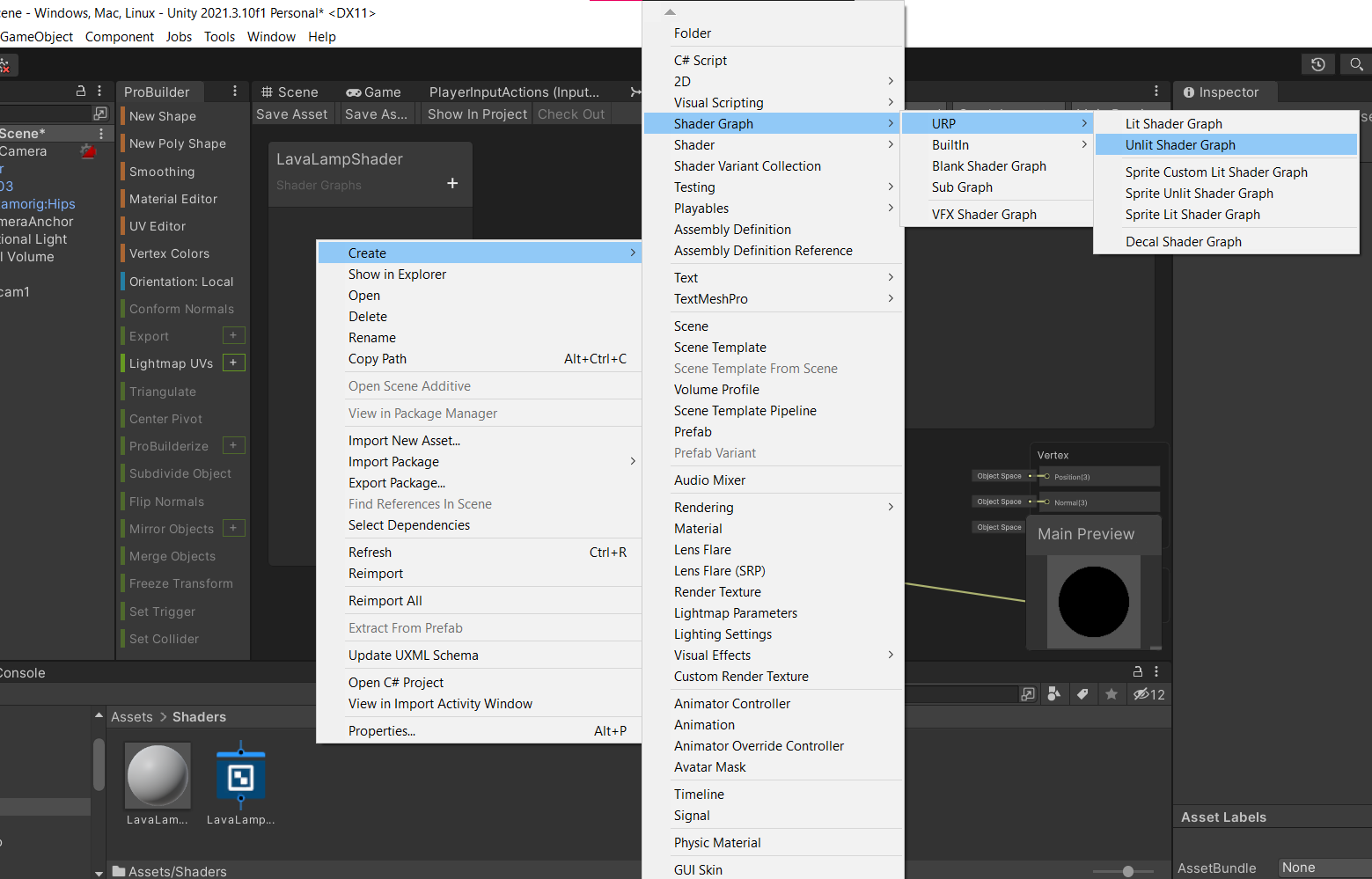


Рисунок 2. Создание шейдера

1. Дальше создаем материал

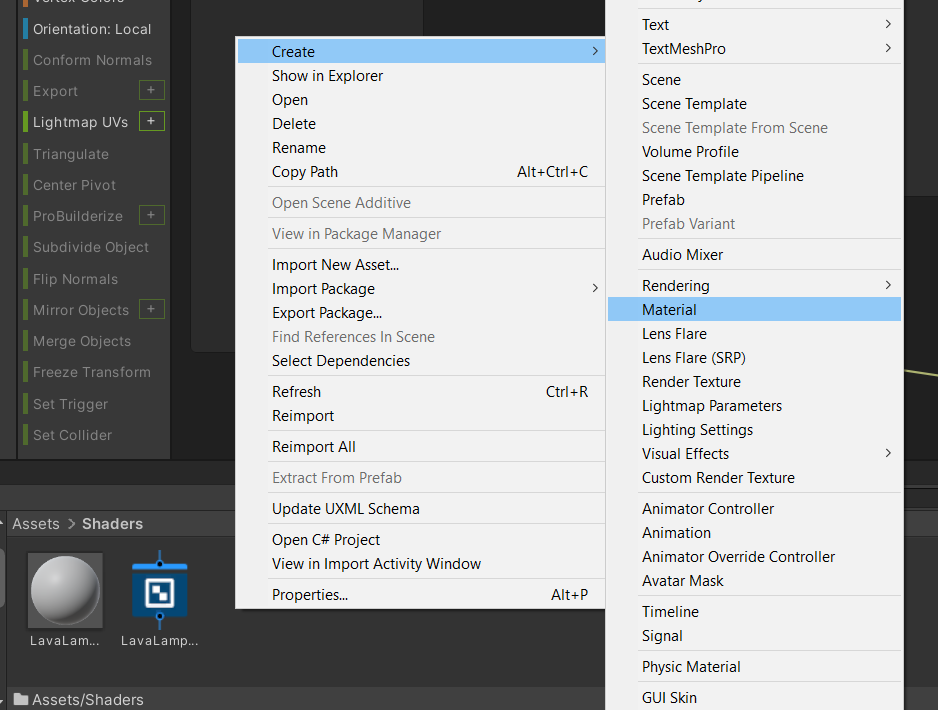


Рисунок 3. Создание материала LavaLampMaterial

1. Запускаем LavaLampShader

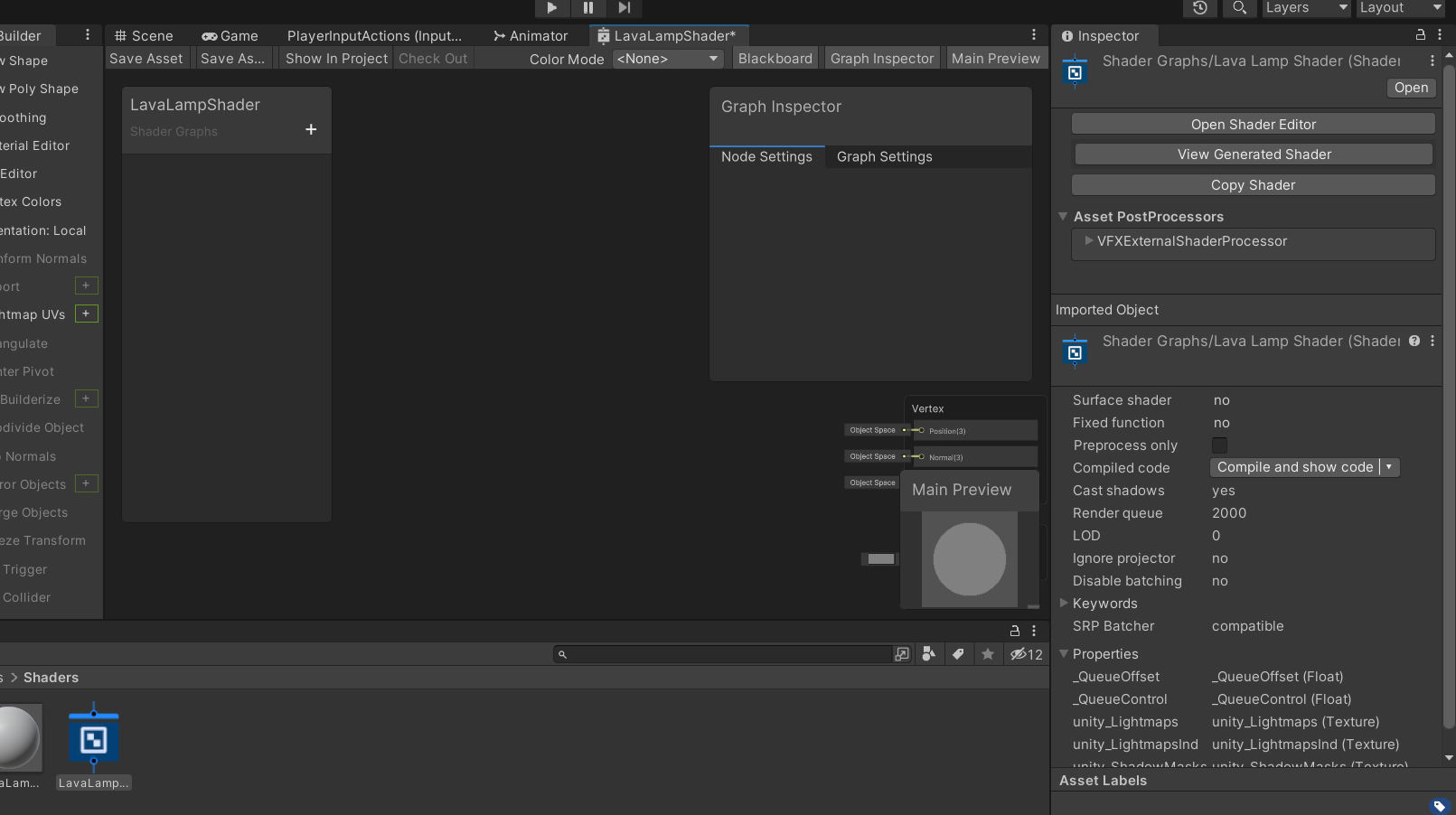


Рисунок 4. Открывается окно настройки щейдера

1. Создаем вектор

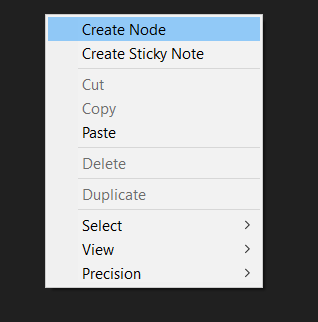


Рисунок 5. Нажимаем ПКМ и выбираем Create Node

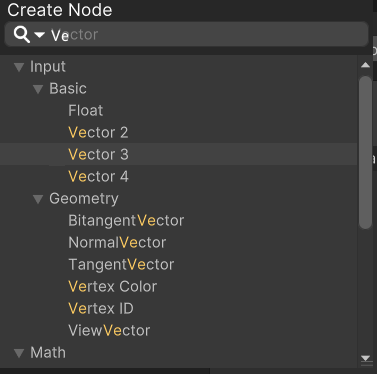


Рисунок 6. Ищем Vector 3 и кликаем два раза

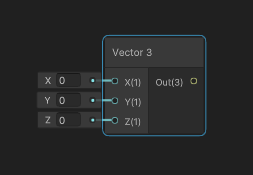


Рисунок 7. Созданный вектор

1. Строим связь с Base Color



Рисунок 8. Связь

1. Как и ранее с помощью ПКМ создаем новый элемент UV

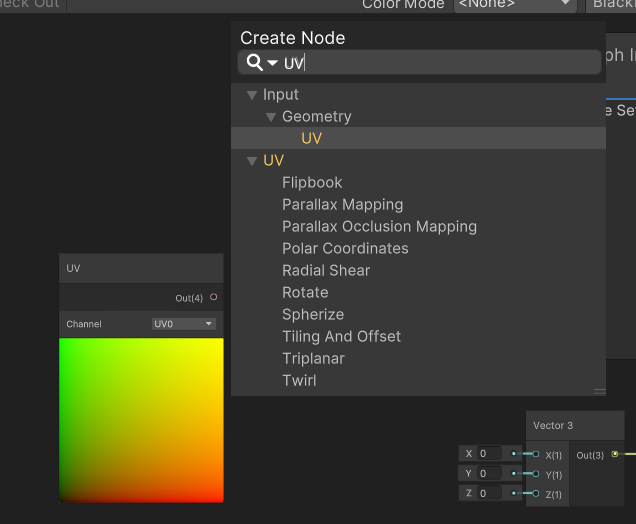


Рисунок 9. Создание элемента

1. По аналогии создаем Tiling And Offset

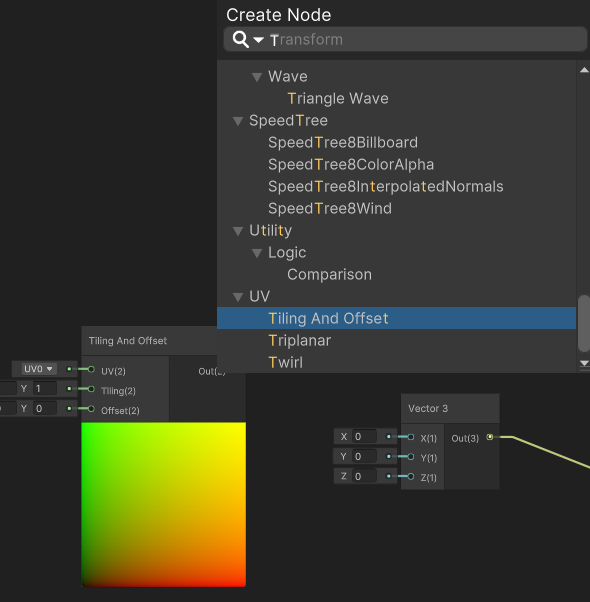


Рисунок 10. Создание элемента

1. Проводим связь

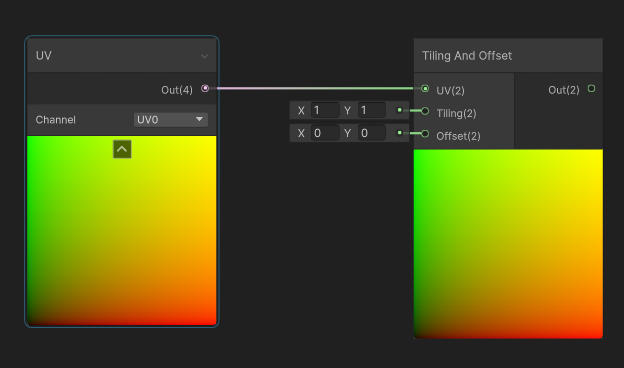


Рисунок 11. Связь

1. Создаем ещё элемент Simple Noise

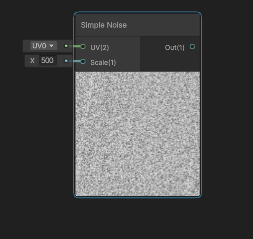


Рисунок 12. Добавление элемента

1. Проводим связь

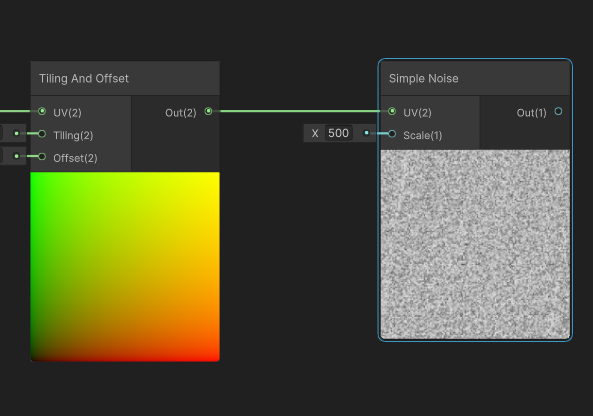


Рисунок 13. Связь

1. Меняем значение x

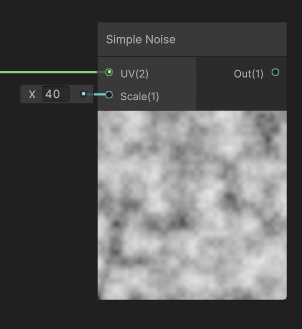


Рисунок 14. Измена значения

1. Создаем ещё два элемента Time и Multiply и строим связь

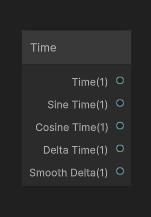


Рисунок 15. Элемент Time

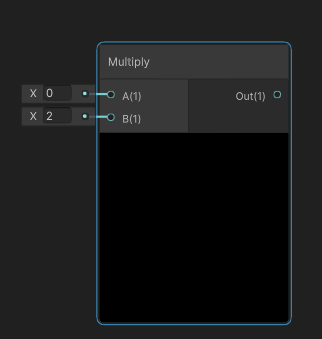


Рисунок 16. Элемент Multiply

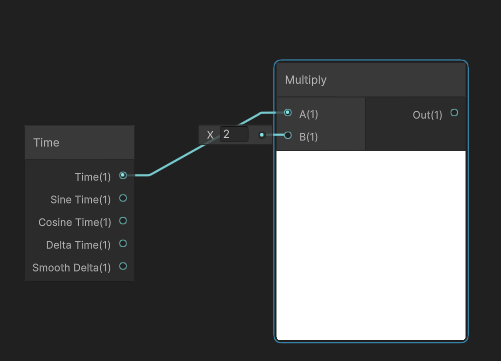


Рисунок 17. Связь

1. Создаем Vector 2 и связываем

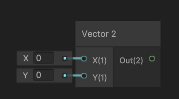


Рисунок 18. Элемент Vector 2

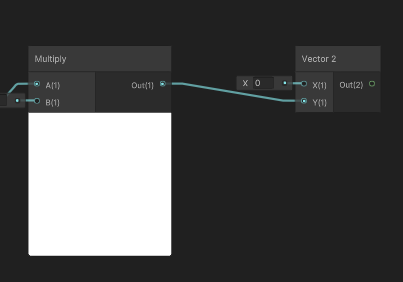


Рисунок 19. Связь

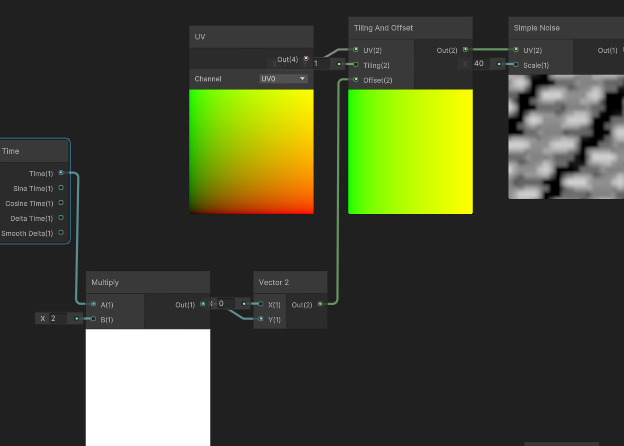


Рисунок 20. Связь

1. Добавляем еще элементы, связи и в итоге получается такая цепь

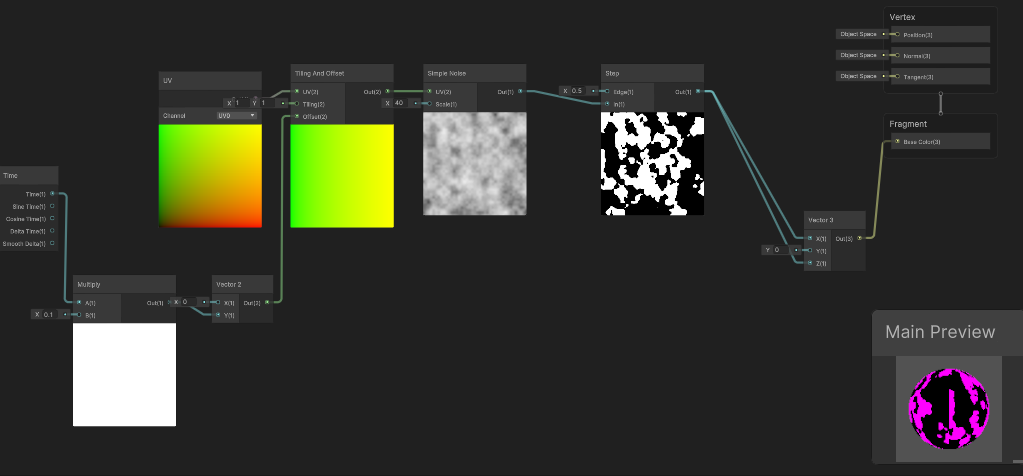


Рисунок 21. Готовая цепь шейдеров



Рисунок 22. Так выглядят шейдеры в игре

1. Добавляем фотографию в папку с шейдерами

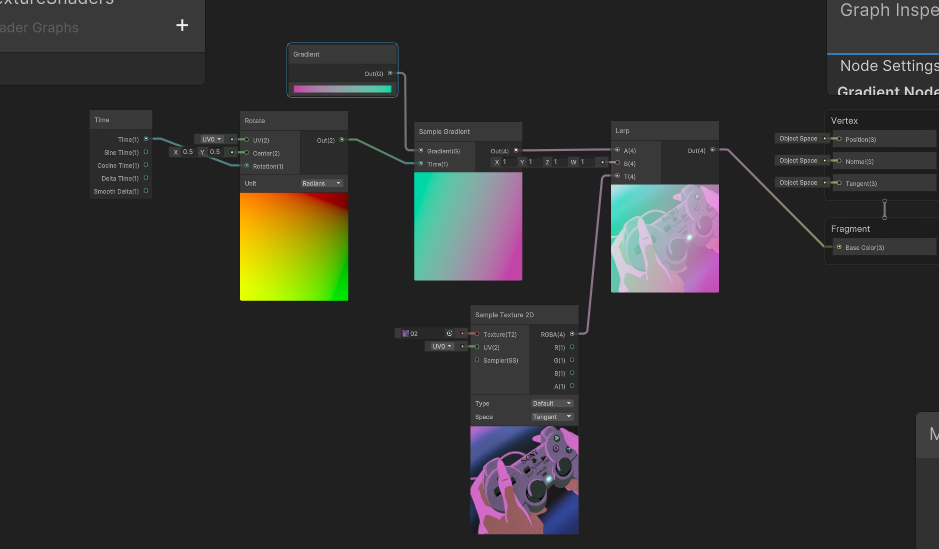


Рисунок 23. Фото в папке

1. Знакомимся с новыми элементами и по классике строим связи и настраиваем элементы



Рисунок 24. Готовый результат для второго шейдера

1. Делаем третий шейдер.

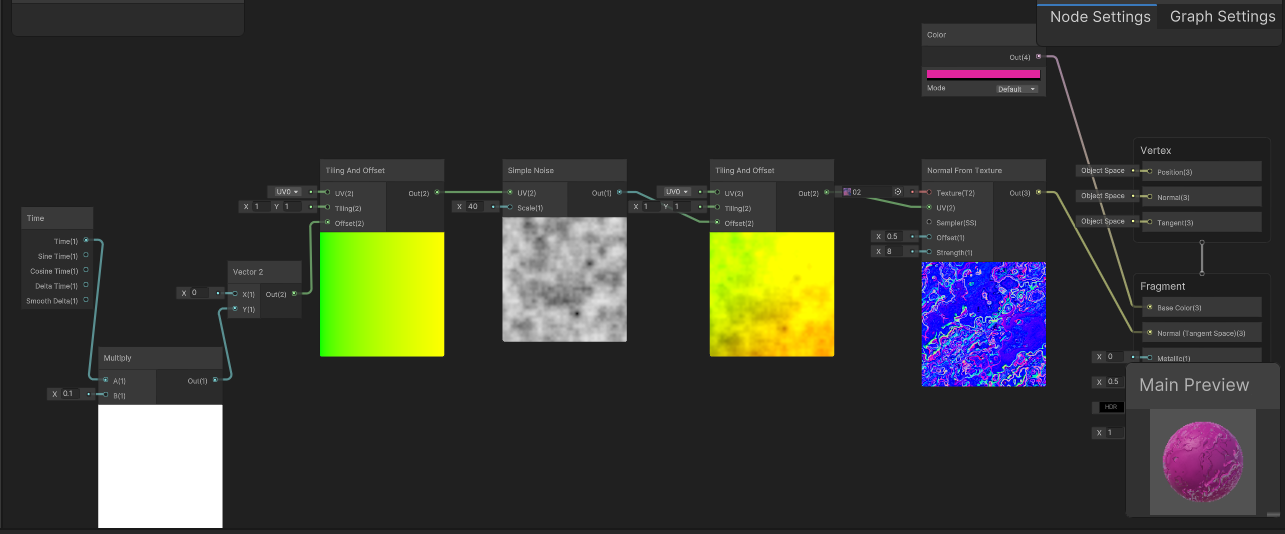


Рисунок 25. Цепь третьего шейдера

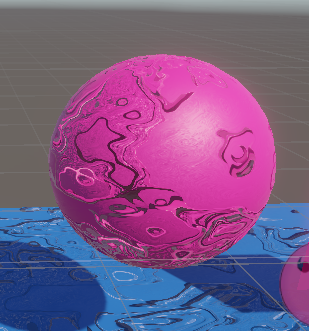


Рисунок 26. Готовый результат для третьего шейдера

1. Далее по примерам (или не совсем) добавляем ещё два шейдера

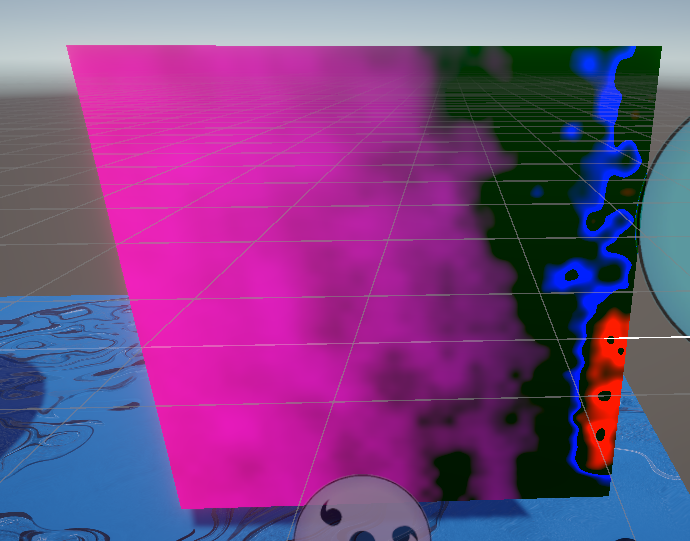


Рисунок 26. Готовый результат для четвертого шейдера

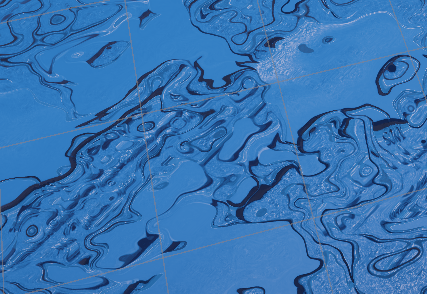


Рисунок 27. Последним шейдером будет реализация воды на платформе, по ней будет ходить персонаж

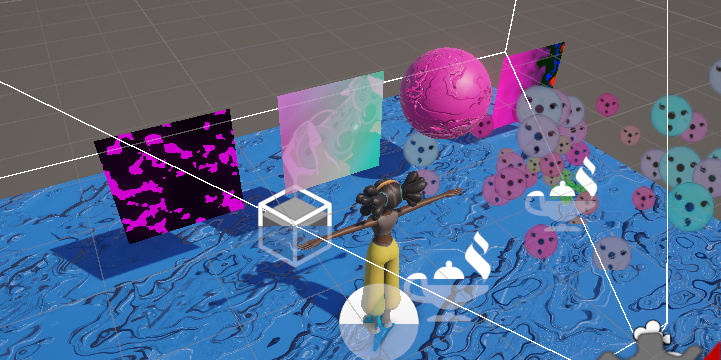


Рисунок 28. Готовые шейдеры в игре

Вывод: в данной практической работе при помощи базовых знаний интерфейса Unity и языка программирования С#, были добавлены в проект шейдеры.