

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«Уральский федеральный университет  
имени первого Президента России Б.Н. Ельцина»  
Институт радиоэлектроники и информационных технологий – РТФ

**Отчет №5 по дисциплине «Цифровые двойники в электронике,  
радиотехнике и системах связи»**

Преподаватель \_\_\_\_\_ Денисов Дмитрий Вадимович  
Студент \_\_\_\_\_ Хадж Махаммед Яхья  
Группа \_\_\_\_\_ РИ-231103

Екатеринбург  
2025

## Конспект лекции

### 1. Что такое Unity?

- Кроссплатформенный движок для создание игр и приложений.
- Поддерживает два вида программирования:

- **C#**
- **Bolt** визуальные блоки.

### 2. Возможности Unity:

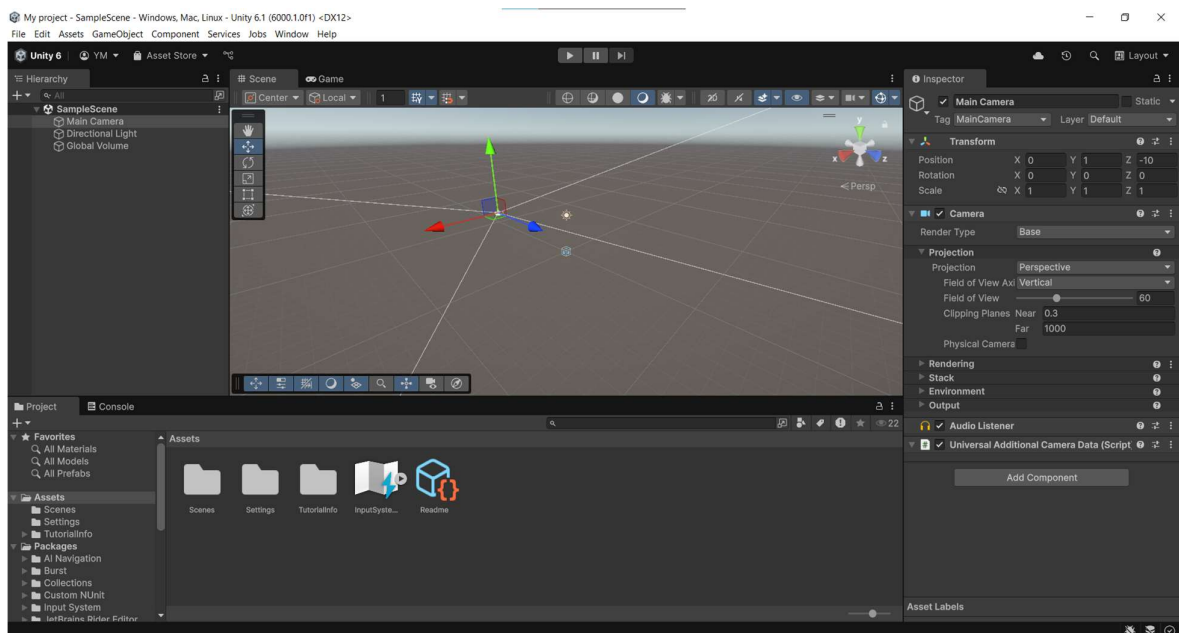
- Создание игр.
- VR-симуляторы .
- Обучающие проектов: виртуальные лаборатории, эксперименты.

### 3. Структура проекта:

- Папки: **Assets**, **Project Settings**, **Library**.
- Основные окна: **Hierarchy**, **Scene**, **Inspector**.

### 4. Советы для разработки:

- Использовать **Unity Hub** для управлениях проекты.
- Сочетать **C#** и **Bolt** для упрощения кода.



## **Рефлексия:**

### **Что я понял:**

- Unity — это мощный инструмент не только для игр, но и для наук и образование. Мне понравилось, что можно создавать VR-проекты для медицин.

### **Что было сложно:**

- Трудно разобраться в терминах: Assets, Prefabs, Rigidbody. Нужно больше практики с этими элементами.
- Непонятно, как правильно настроить Bolt для визуального программирования.

### **Как применю знания:**

- Попробую сделать простую игру с падающими объектом.
- Научусь добавлять физику к объекты, например, чтобы кубик падал на плоскость.

### **Возникелые вопросы:**

- Как работать с анимацией в Unity?
- Можно ли использовать Python вместо C#?

# Динамическая симуляция волн на воде в Unity

## Краткое описание проекта:

Проект демонстрирует реалистичное моделирование волны на поверхности воды с использованием шейдеров и физики Unity.

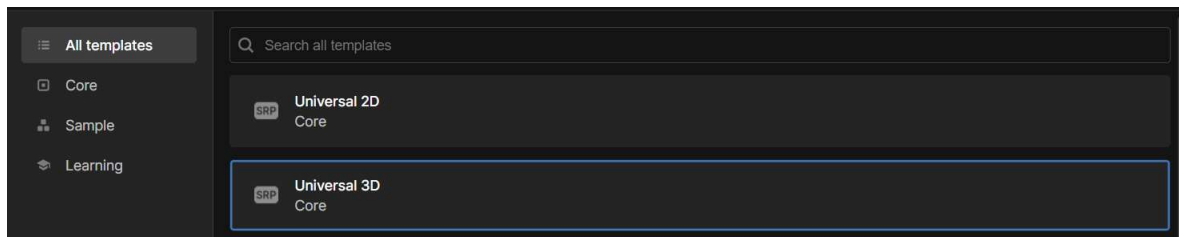
- **Основные функции:**

- Генерация волн при взаимодействии с объектами
- Настройка параметров волн: скорость, амплитуда, плотность.
- Визуализация отражения света и деформации водной поверхности.

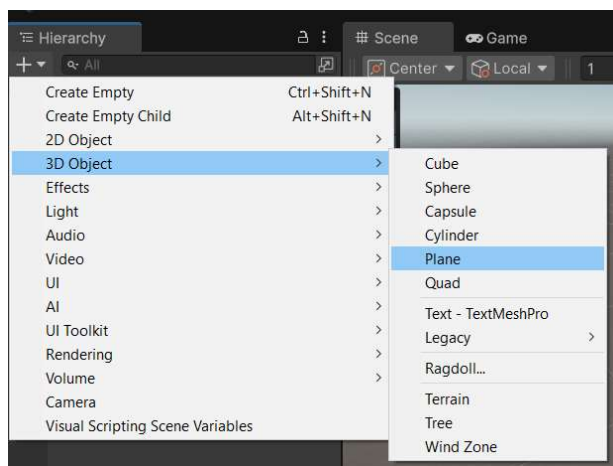
- **Технологии:**

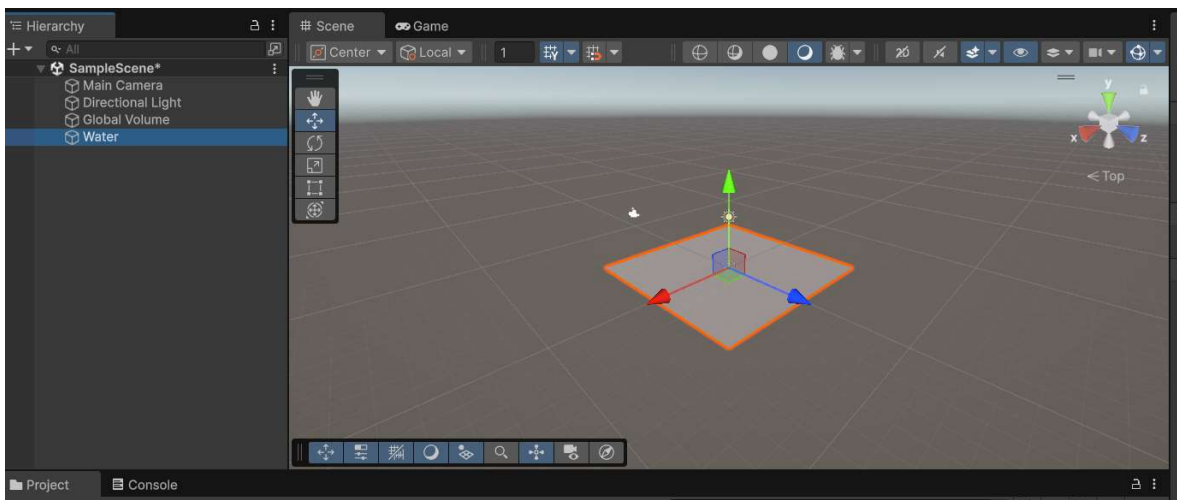
- Шейдеры на основе узлов **Shader Graph**.
- Уравнения Герстнера для моделирования волн.
- Интерактивный интерфейс для управления параметрами.

### 1. Создать новый 3D-проект в Unity:

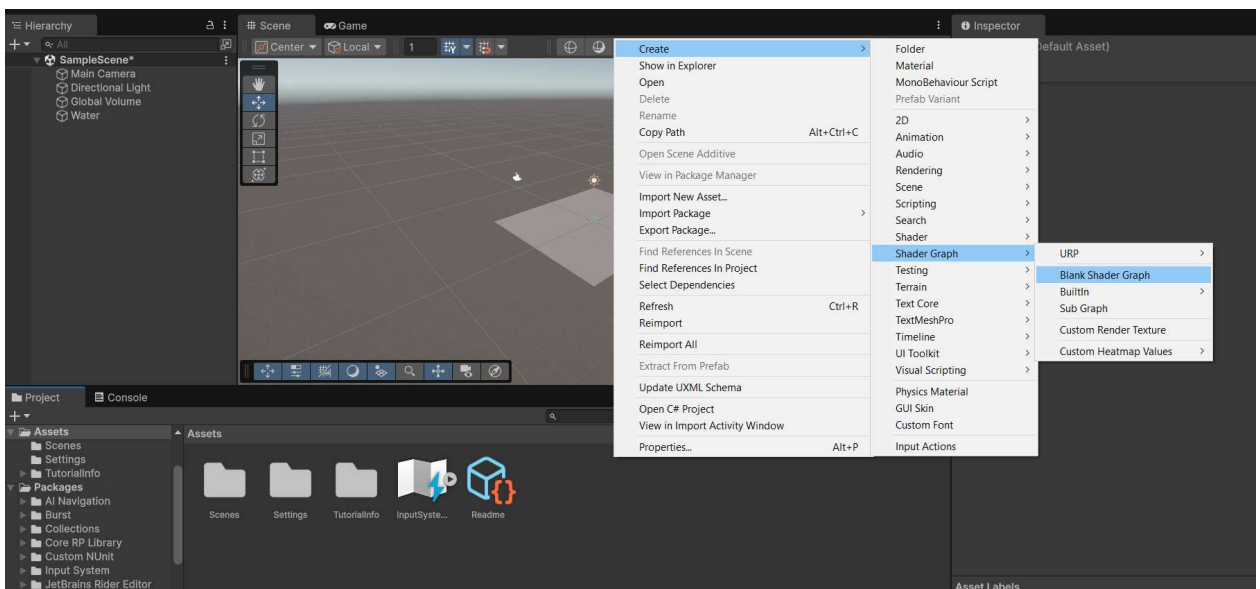


### 2. Создать объект plane и назвать water

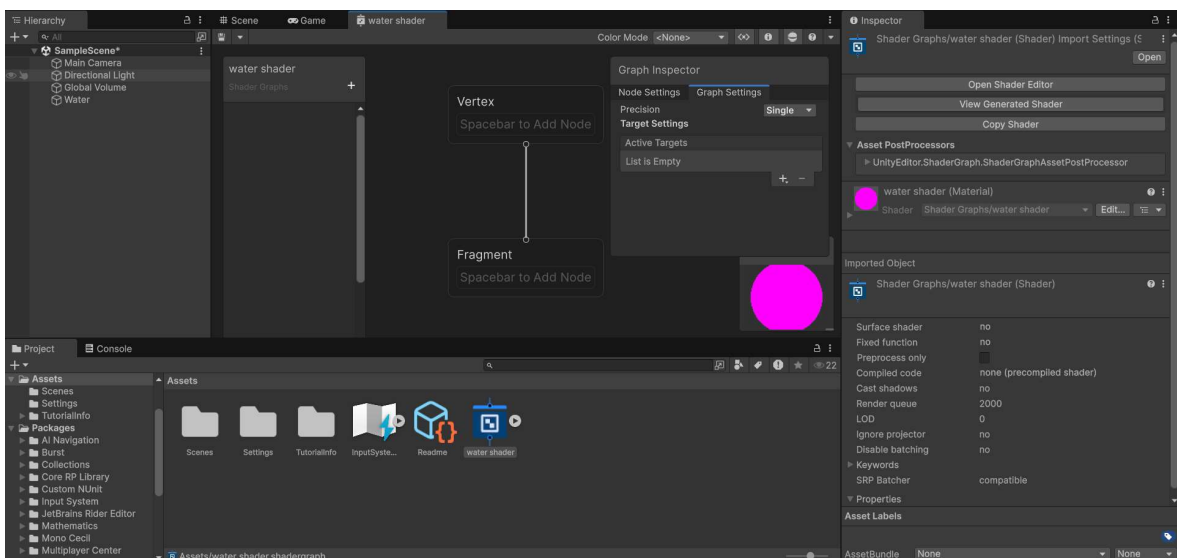




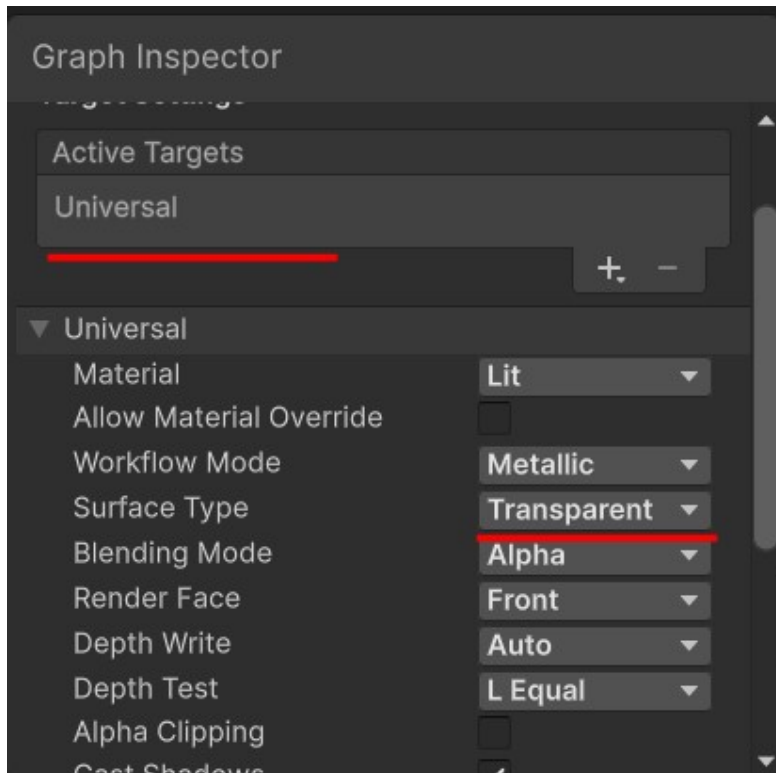
3. Нажать ПКМ по полю assets и найти blank shader graph



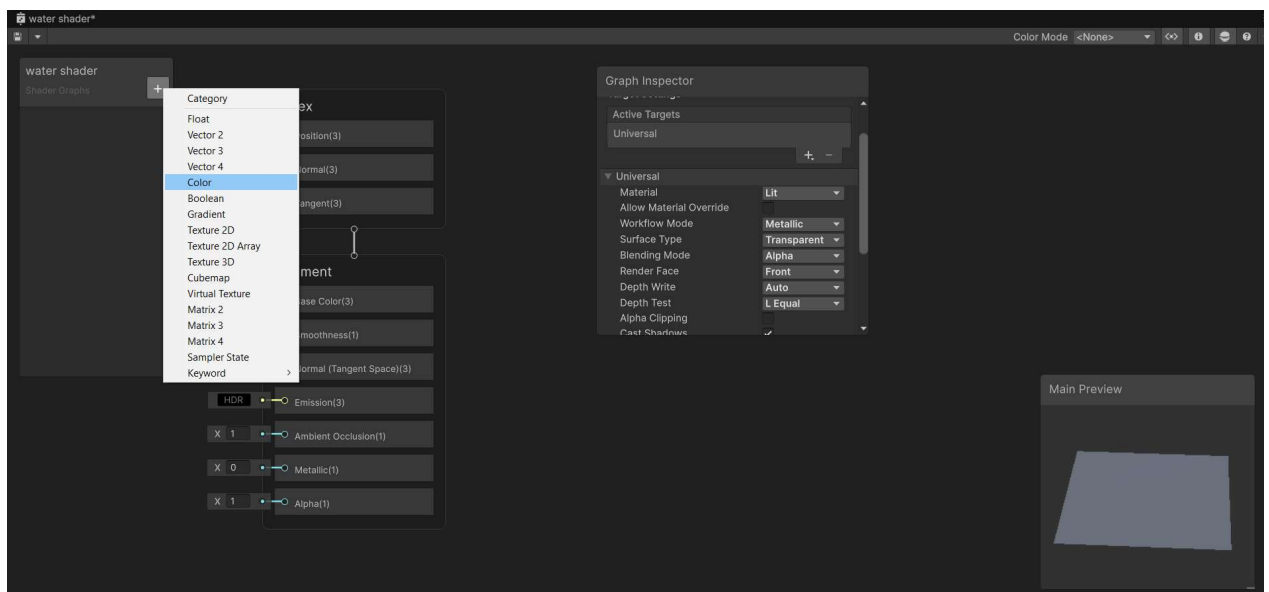
4. Назвать его water shader и двойным кликом по этому объекту открыть инспектор

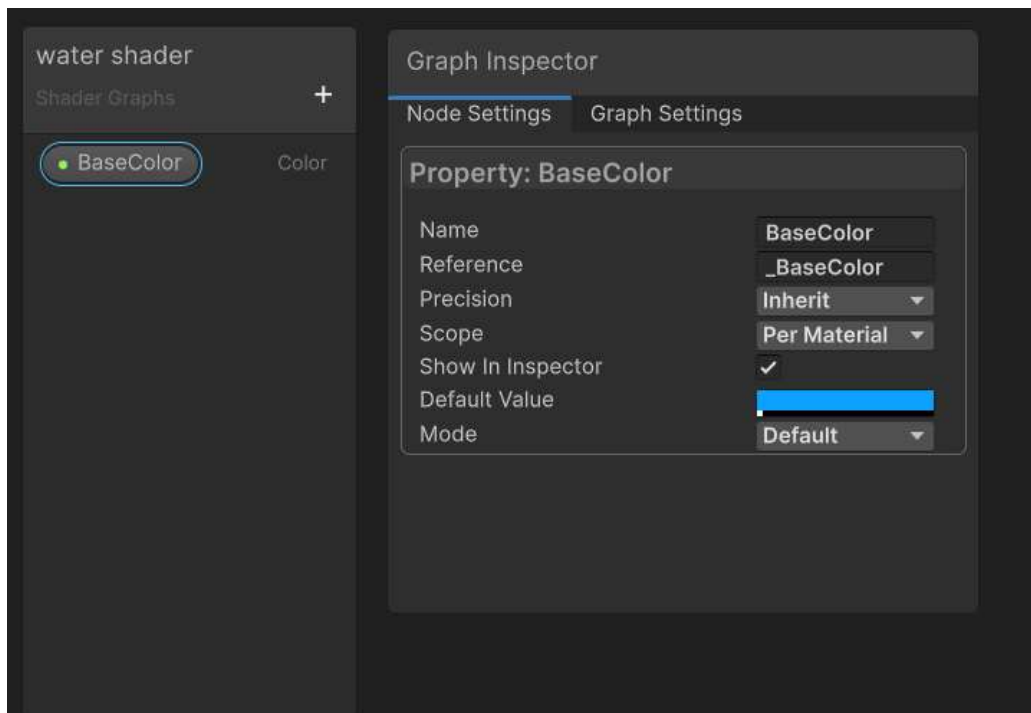


5. Установить active targets – universal и surface – transparent

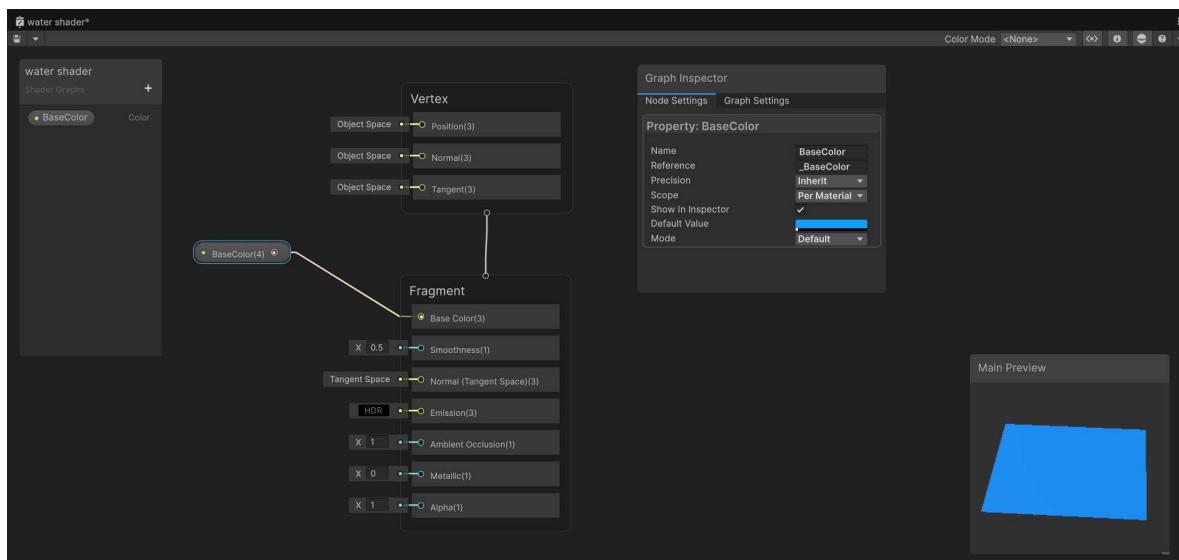


6. Перейти на вид water shader – maximize для дальнейшего удобства и изменить превью объекта на plane.
7. Установить цвет water shade на ярко-голубой и назвать BaseColor

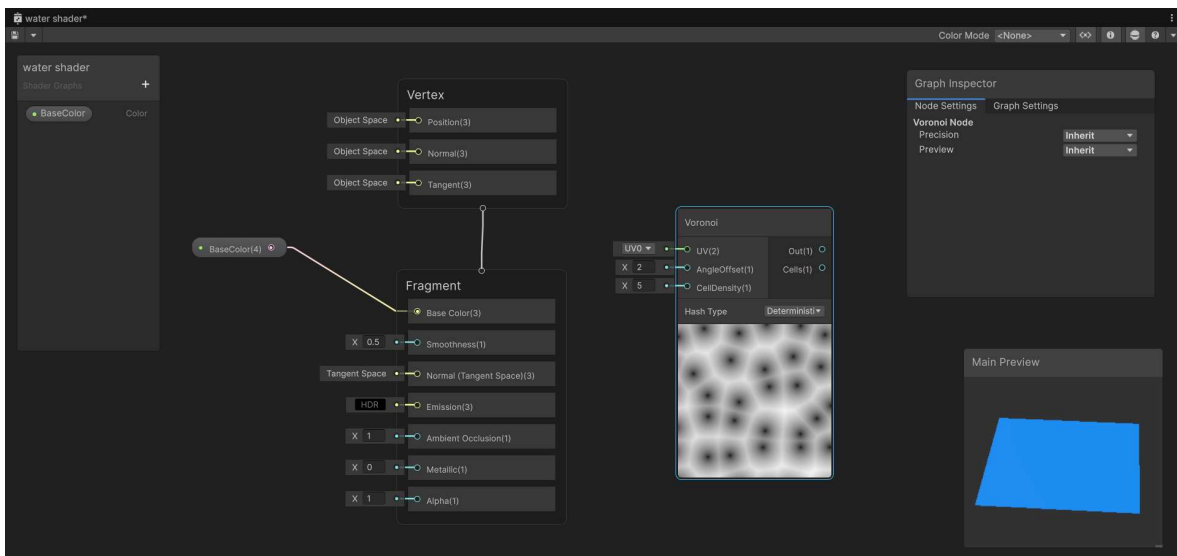




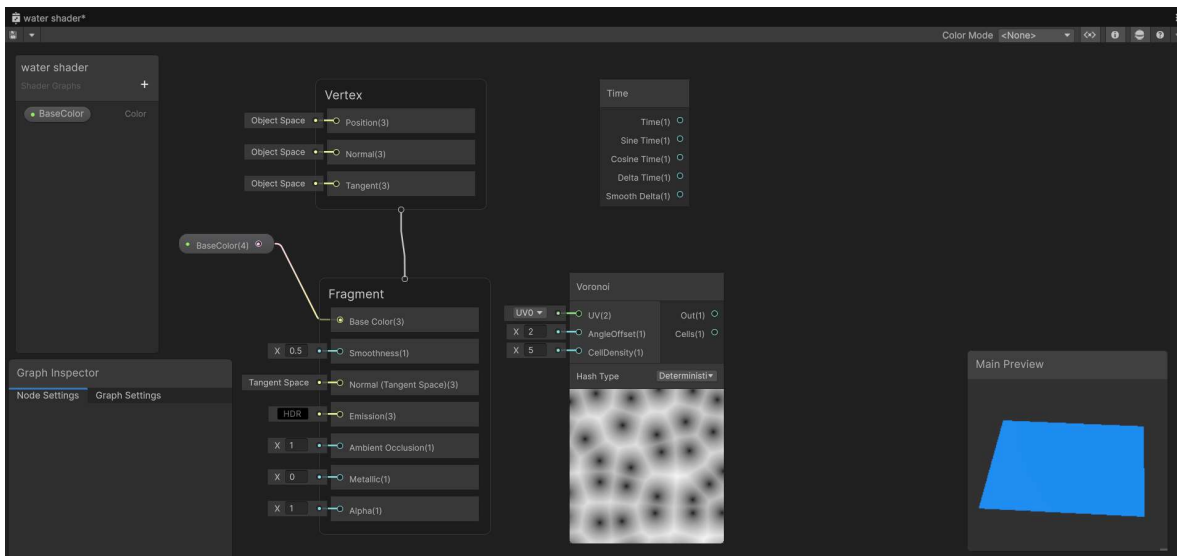
8. Перетащить BaseColor на поле и протянуть связь от него к свойству color фрагмента



9. Нажимаем ПКМ по полю и выбирает create node. Ищем voronoi

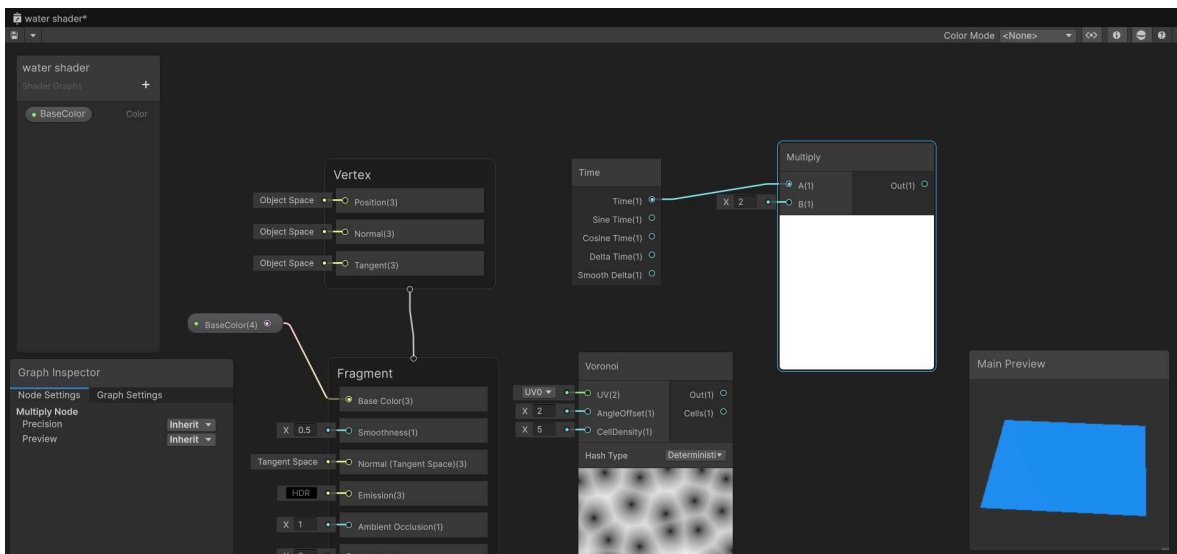


1. Нажимаем ПКМ по полю и выбирает create node. Ищем time

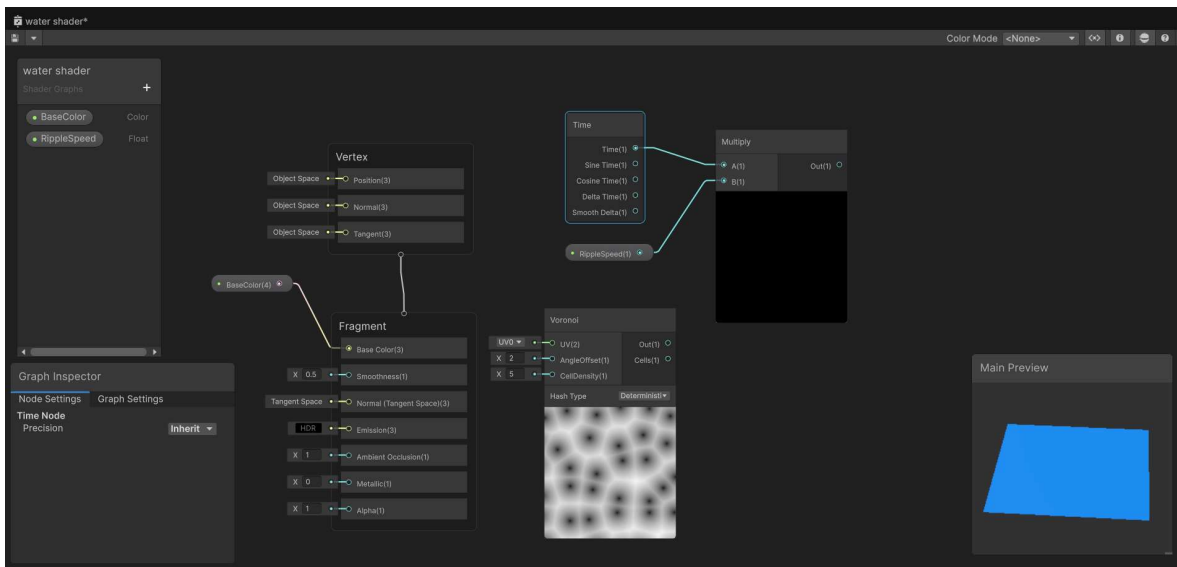


1. Нажимаем ПКМ по полю и выбирает create node. Ищем multiply. Устанавливаем связь между свойством time в time и свойством A в multiply.

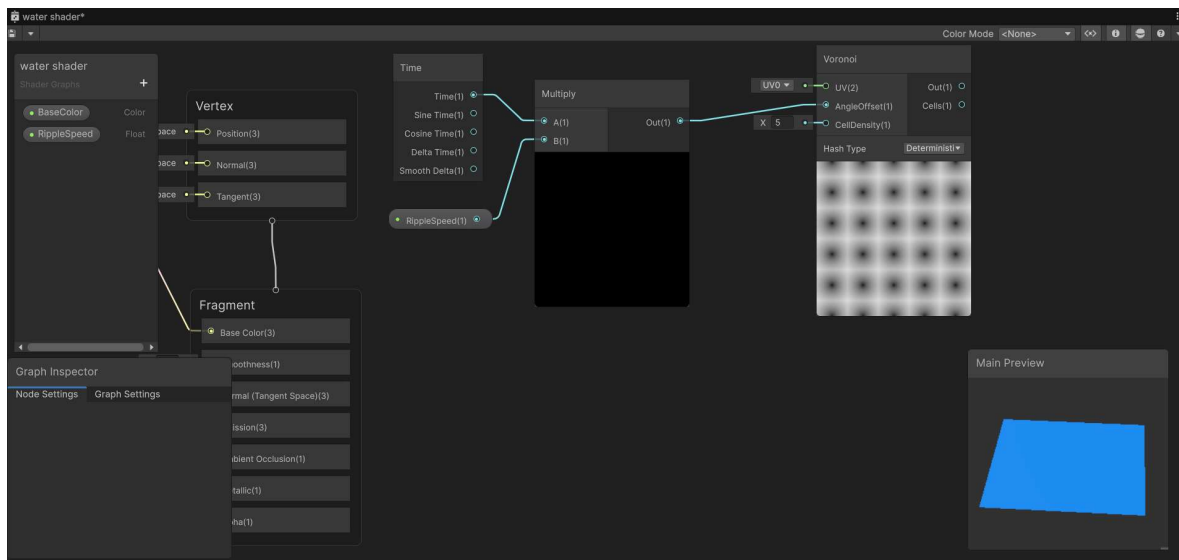




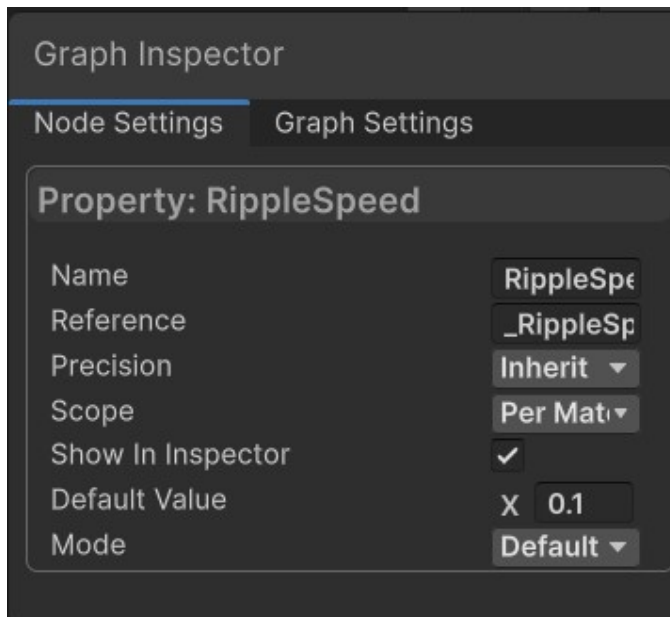
2. Создаем свойство Float в water shader и называем RippleSpeed. Выносим его на поле, устанавливаем ему связь свойством B в multiply



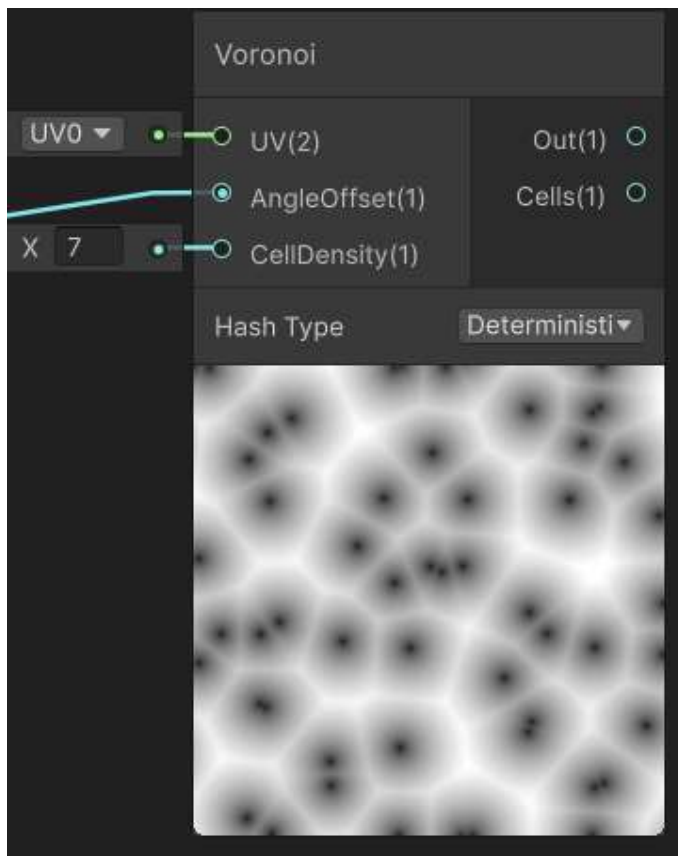
3. Устанавливаем связь между свойством out в multiply и свойством AngleOffset в Voronoi



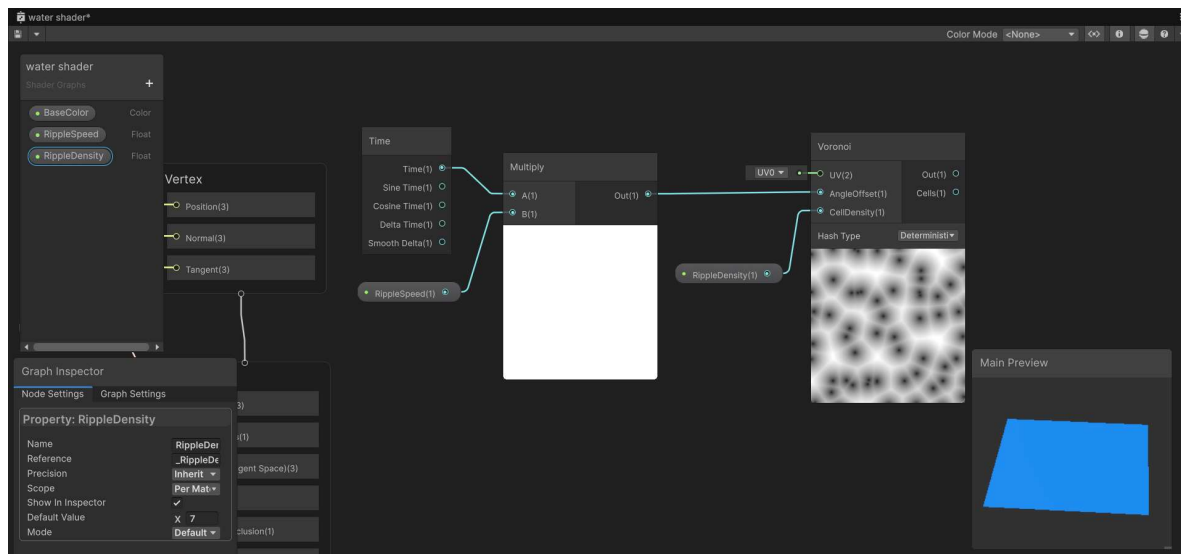
4. Выделяем свойство RippleSpeed и устанавливаем ему default value - 1



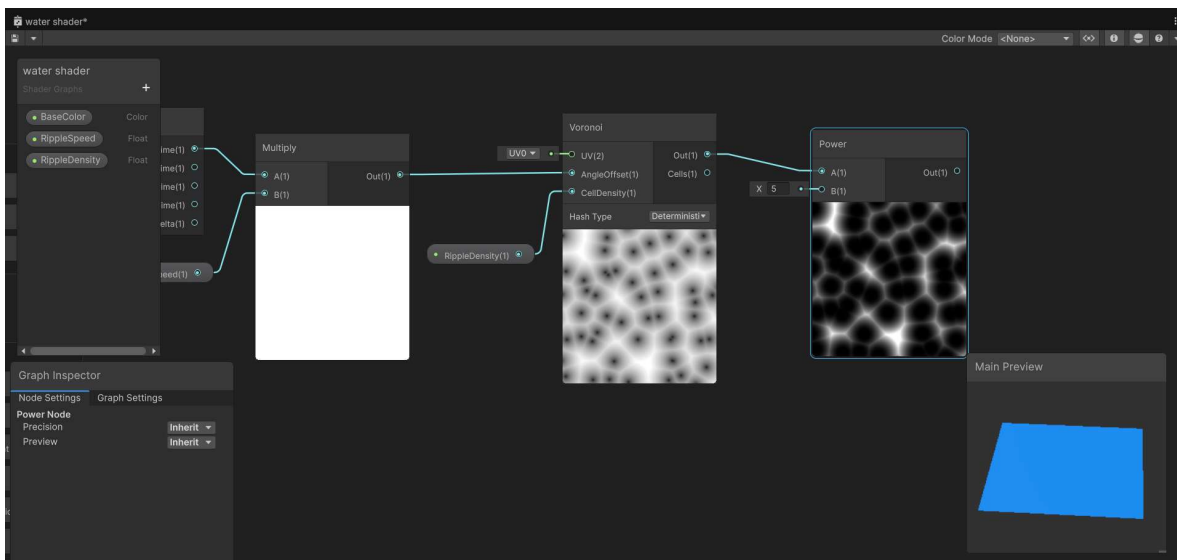
5. Установим x в объекте Voronoi на 7



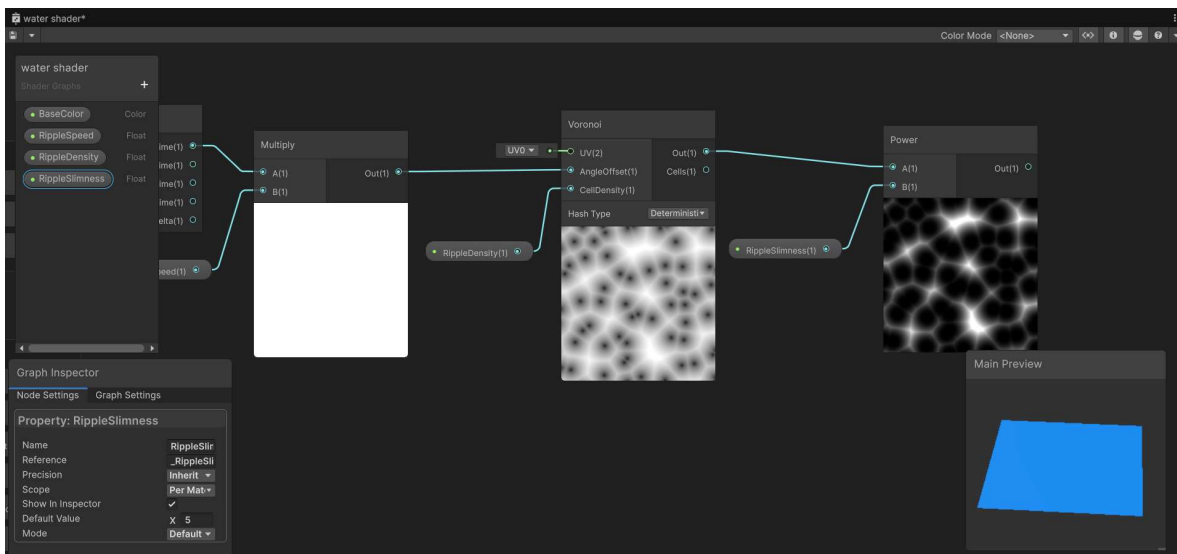
6. Создаем еще одно свойство Float в water shader и называем RippleDensity. Устанавливаем ему default value – 7. Выносим его на поле и устанавливаем ему связь со свойством CellDensity в Voronoi.



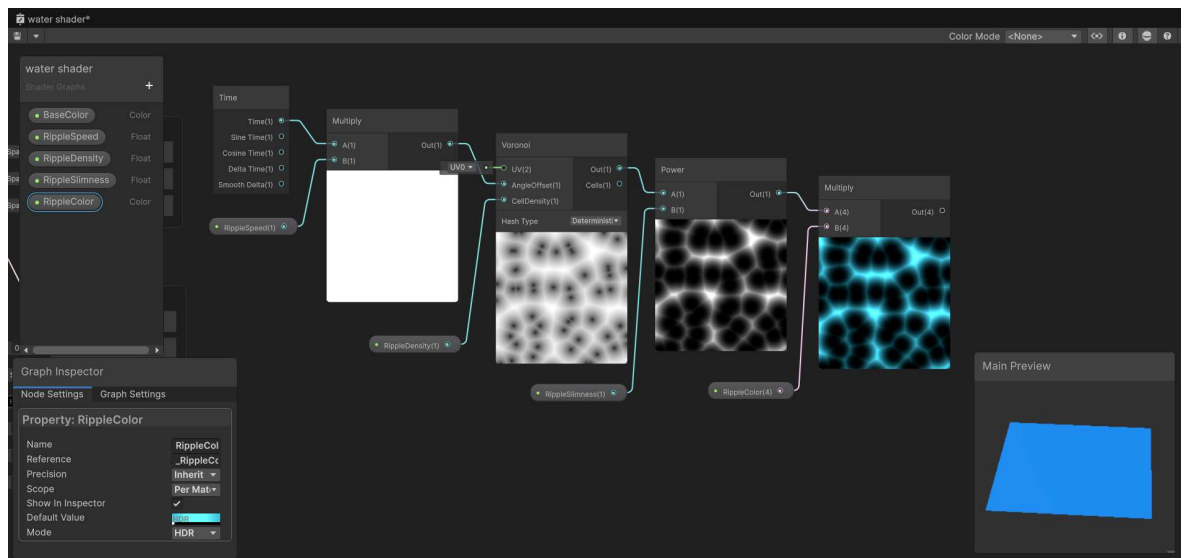
7. Нажимаем ПКМ по полю и выбирает create node. Ищем power. Устанавливаем связь между свойством out в voronoi и свойством A в power. Устанавливаем x в power на 5.



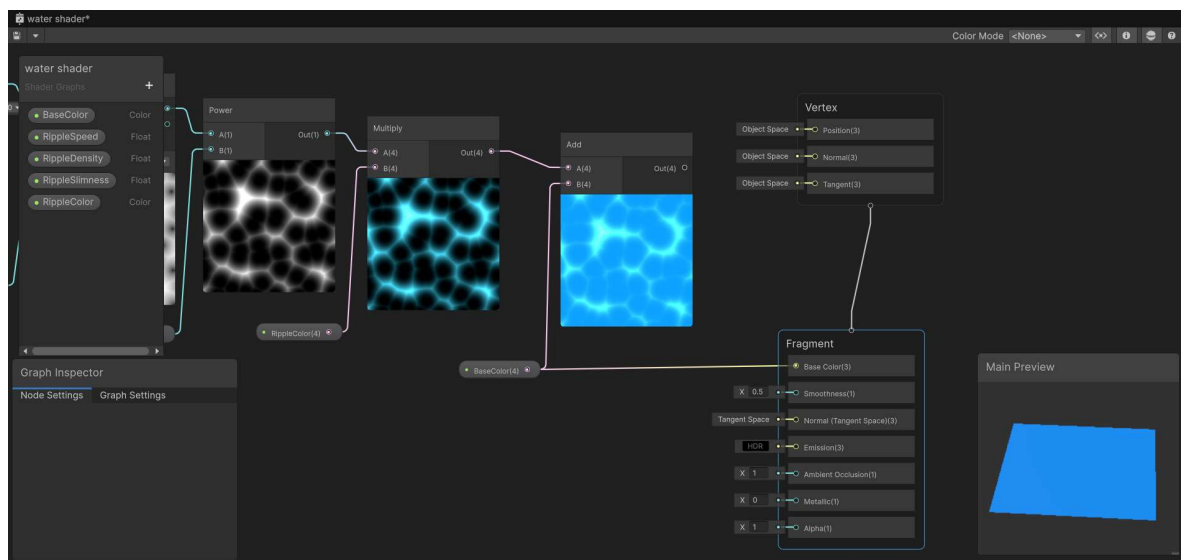
8. Создаем еще одно свойство Float в water shader и называем RippleSlimness. Устанавливаем ему default value – 5. Выносим его на поле и устанавливаем ему связь со свойством b в power.



1. Нажимаем ПКМ по полю и выбирает create node. Ищем multiply. Устанавливаем ему связь по свойству A со свойством out в power.
2. Создаем еще одно свойство Color в water shader и называем RippleColor. Устанавливаем ему mode – HDR и цвет светло-голубой +1. Выносим его на поле и устанавливаем ему связь со свойством b в multiply.

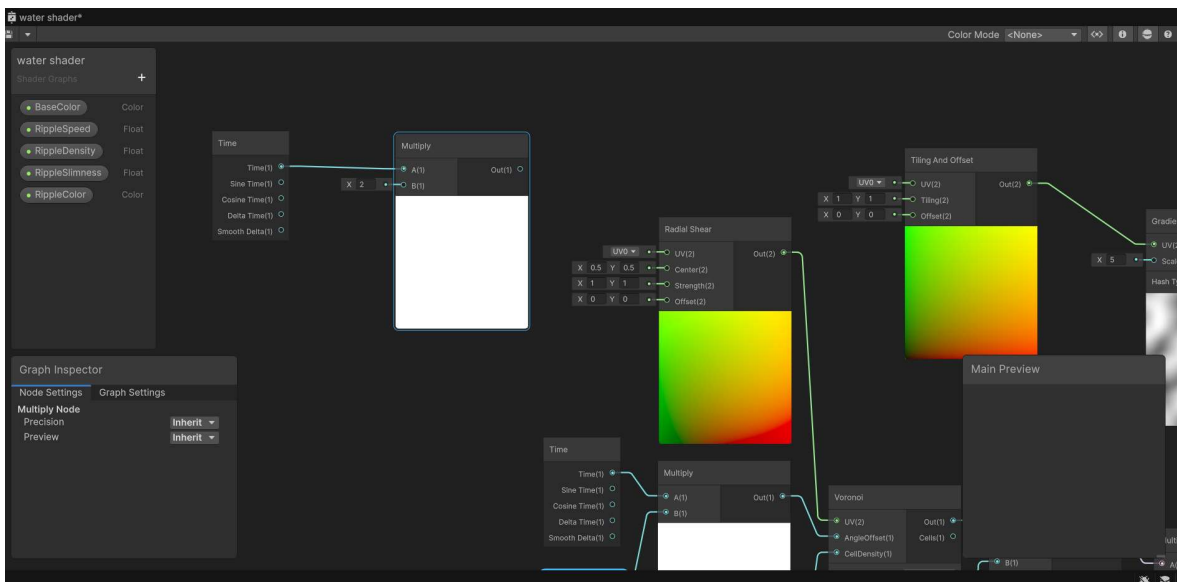


3. Нажимаем ПКМ по полю и выбирает create node. Ищем add. Устанавливаем ему связь по свойству A со свойством out в multiply и по свойству B со свойством BaseColor.

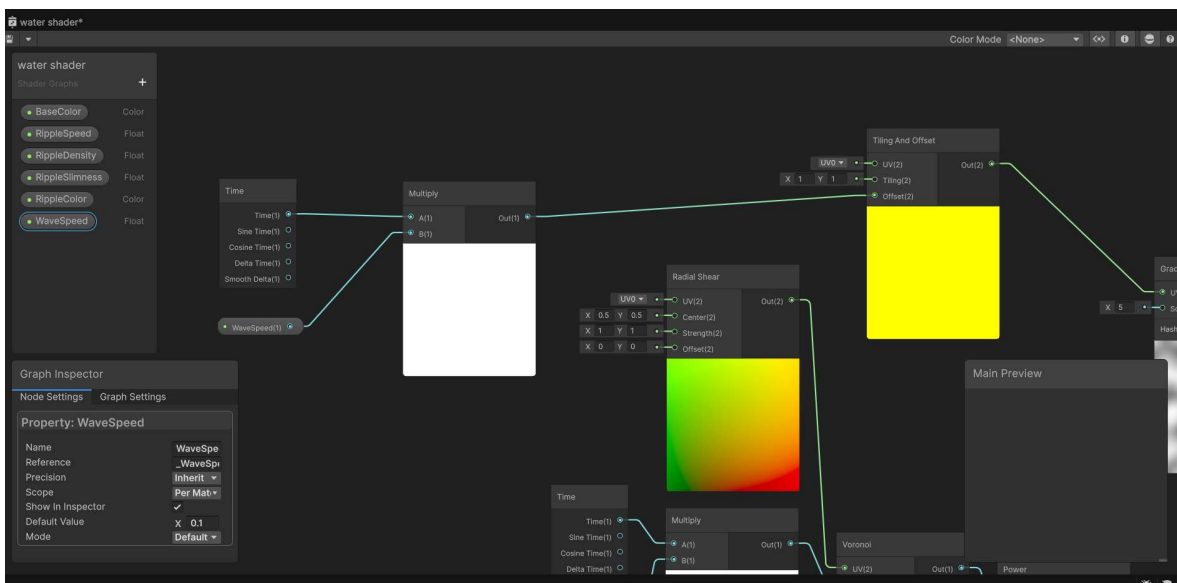


4. Отменить связь BaseColor из shader с фрагментом и установить связь BaseColor из фрагмента со свойством out из add
5. Нажать save
6. Возвращаемся на сцену и переносим water shader material из water shader на объект water.
7. Возвращаемся в вид maximize на water shader. Нажимаем ПКМ по полю и выбирает create node. Ищем radial shear. Ставим ему на x – 1 и на y – 1 в свойстве strength. Устанавливаем ему связь по свойству out со свойством UV в voronoi.

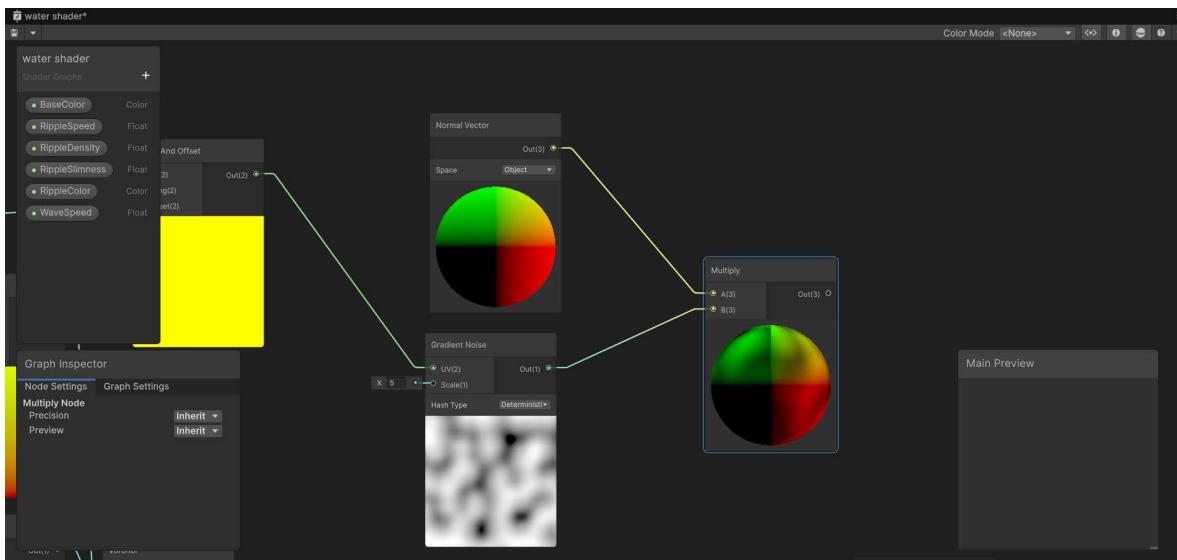




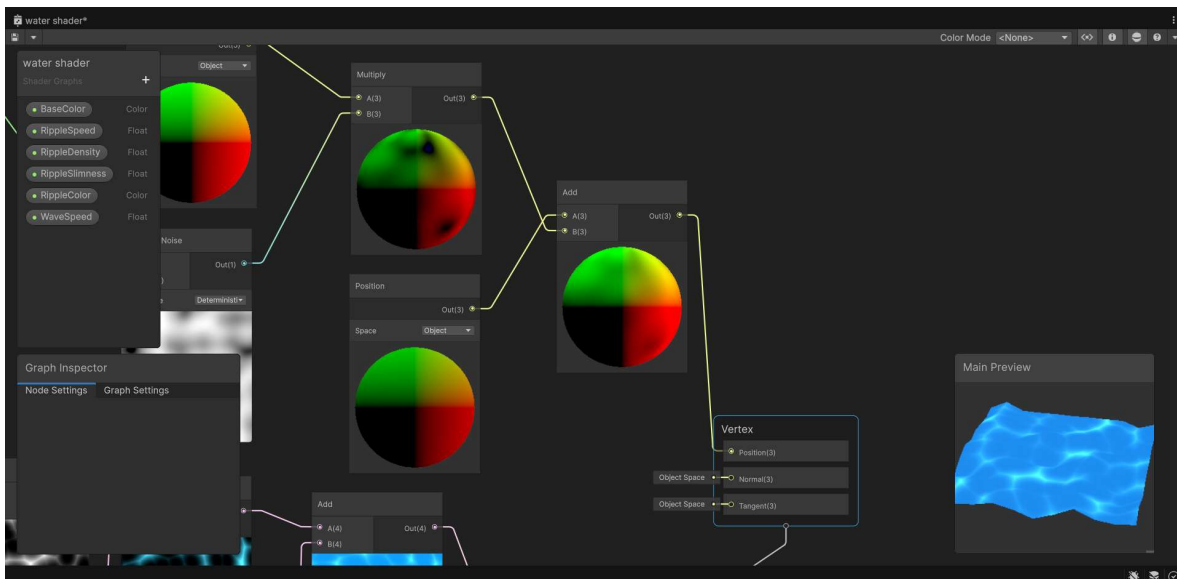
11. Создаем свойство Float, называем WaveSpeed, устанавливаем default value – 0.1. Выносим на поле и соединяем с Multiply. Соединяем out Multiply с offset в Tiling and offset.



12. Создаем узел Normal vector и узел Multiply, соединяем. Соединяем тот же Multiply с out в Gradient noise.

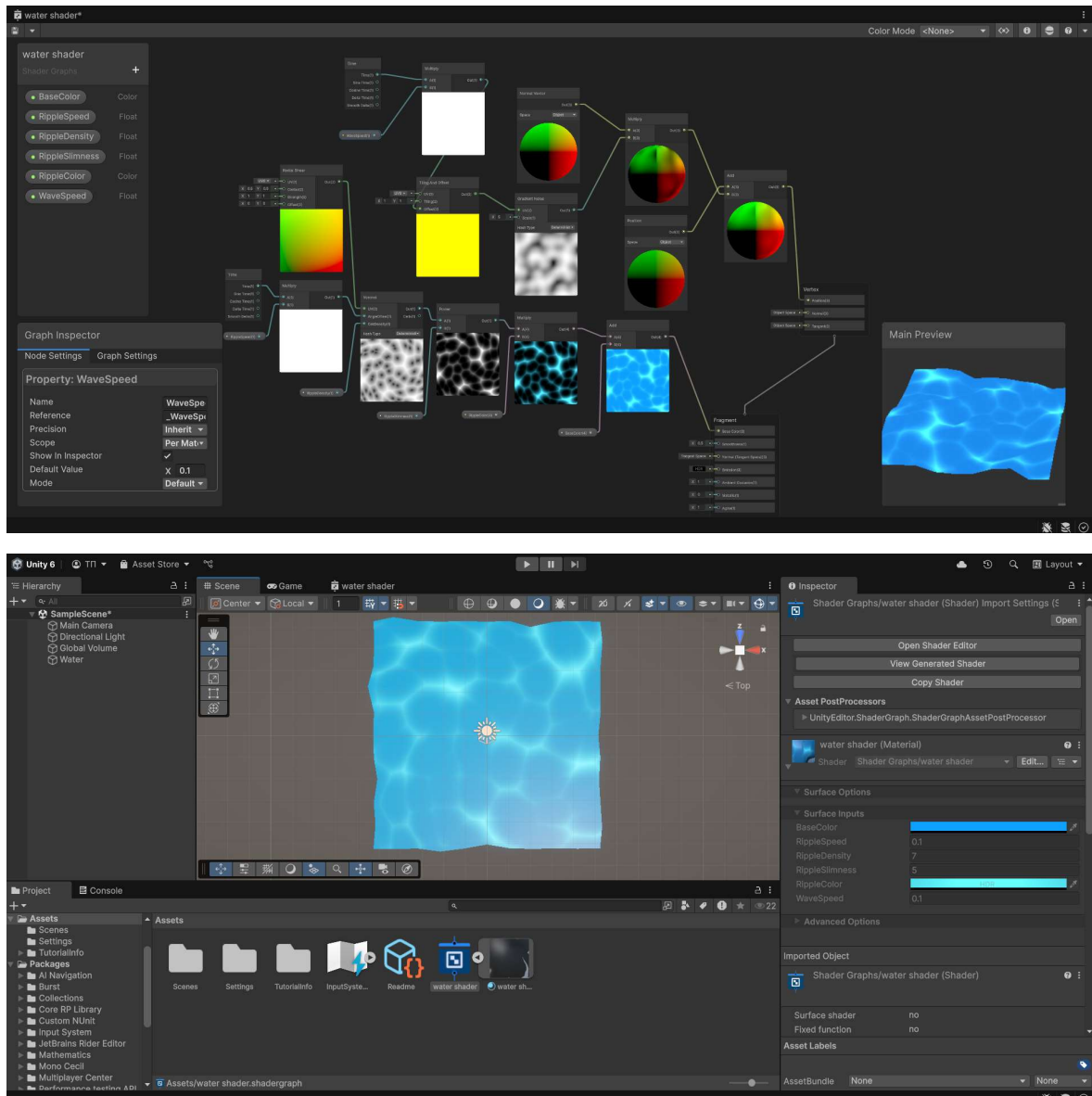


1. Создаем узел Position и узел Add, соединяем. Соединяем тот же Add с out в Multiply. Соединяем Add с Position в Vertex.





## 2. Итог:



Ссылка на GitHub: <https://github.com/yahia-2002/WaterWaveSimulation.git>