Экзаменационная работа по дисциплине

«Высокоуровневые методы программирования»

Студент: Моисеев Владислав Артёмович ИД 23.1\Б-21

Научный руководитель: Преображенский Максим Владимирович

2023г.

Введение

**Цели проекта**: Разработка видео-игры, где моя непосредственная роль – графика. Я создаю модели, анмимрую их, накладываю текстуры и экспортирую. В том числе на мне создание шейдеров для всё тех же моделей.

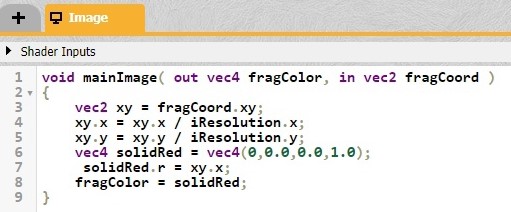
Основная часть

Технологическая задача

1. Список решаемых задач:
   1. Написание собственной текстуры кодом.
      1. Для решения задачи пользовался языком специальным языком для написания шейдеров, основанном на языке программирования c – GLSL, платформой для написания шейдеров - shadertoy.
   2. Написание шейдера анимации воды.
      1. Для решения задачи пользовался специальным языком для написания шейдеров, основанном на языке программирования c – cg, кроссплатформенной средой разработки компьютерных игр – Unity.
   3. Написание шейдера более естественного и более плавного вида погружения объекта в воду.
      1. Для решения задачи пользовался специальным языком для написания шейдеров, основанном на языке программирования c – cg, кроссплатформенной средой разработки компьютерных игр – Unity.
   4. Написание шейдера, обозначающего движение под водой NPC.
      1. Для решения задачи пользовался специальным языком для написания шейдеров, основанном на языке программирования c – cg, кроссплатформенной средой разработки компьютерных игр – Unity.

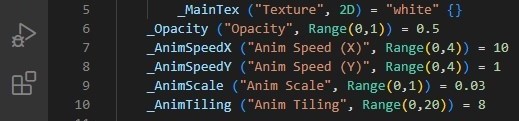
Реализуемая архитектура

1. Реализация функционала.
   1. Написание собственной текстуры кодом. На стартовом этапе я изучал создание визуальных эффектов с цветами в текстурах. И так как, это было первое задание на пути изучения шейдеров, я дошёл до градиента включительно и его записал, как результат своей работы, потому что задание «создать собственную текстуру» уже было выполнено и на этапе заливки всего экрана нужным цветом. Дальнейшее изучение было из собственного интереса, в результате чего я мог предложить уже не просто одноцветную текстуру. Ниже приведён код этого градиента на GLSL:



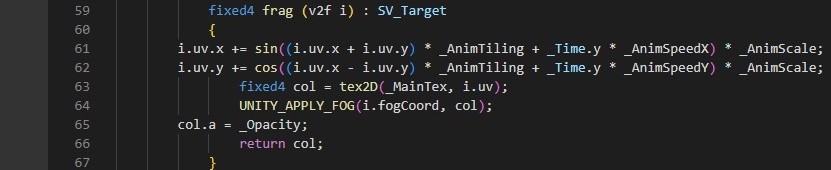
Эти девять строк кода создали градиент. Результат скромный, но показывал в начале изучения, что основы синтаксиса изучены.

* 1. Написание шейдера анимации воды. Здесь уже включается синтаксис cg, которым я писал все дальнейшие решения. В этот раз уже, разумеется, буду делать акцент не всём коде, а на самых важных, определяющих его, участках:



Как и в любом шейдере, начинаем с объявления параметров, которые будут меняться. Особенностью относительно остальных здесь является не изменение цвета, а анимация. Так что большая часть критериев связана с анимацией.

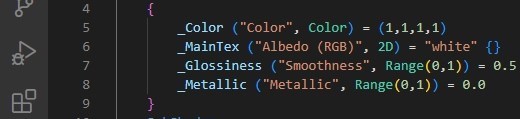
Далее в коде объявляются различные параметры рендеринга, вывода информации и тп. На этом останавливаться не будем. Важнее то, что идёт дальше. Сама функция построения анимации



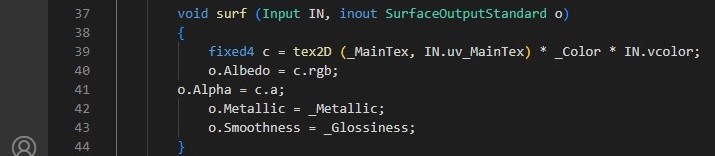
В ней мы позволяем менять положение пикселей по заданным нами функциям в зависимости от заданных в начале параметров, что и даёт эффект анимации.

* 1. Написание шейдера более естественного и более плавного вида погружения объекта в воду.

В этом случае входные параметры самые обычные, скрин нужен лишь для понимания, какие переменные участвуют в дальнейшей функции



Далее, как и в прошлом случае, объявляются различные параметры рендеринга, вывода информации и тп. Так что снова перейдём к главному - к функции:



Функция получает цвет из текстуры и проводит его изменения в соответствии с заданными в начале параметрами, что и даёт искомый визуальный эффект.

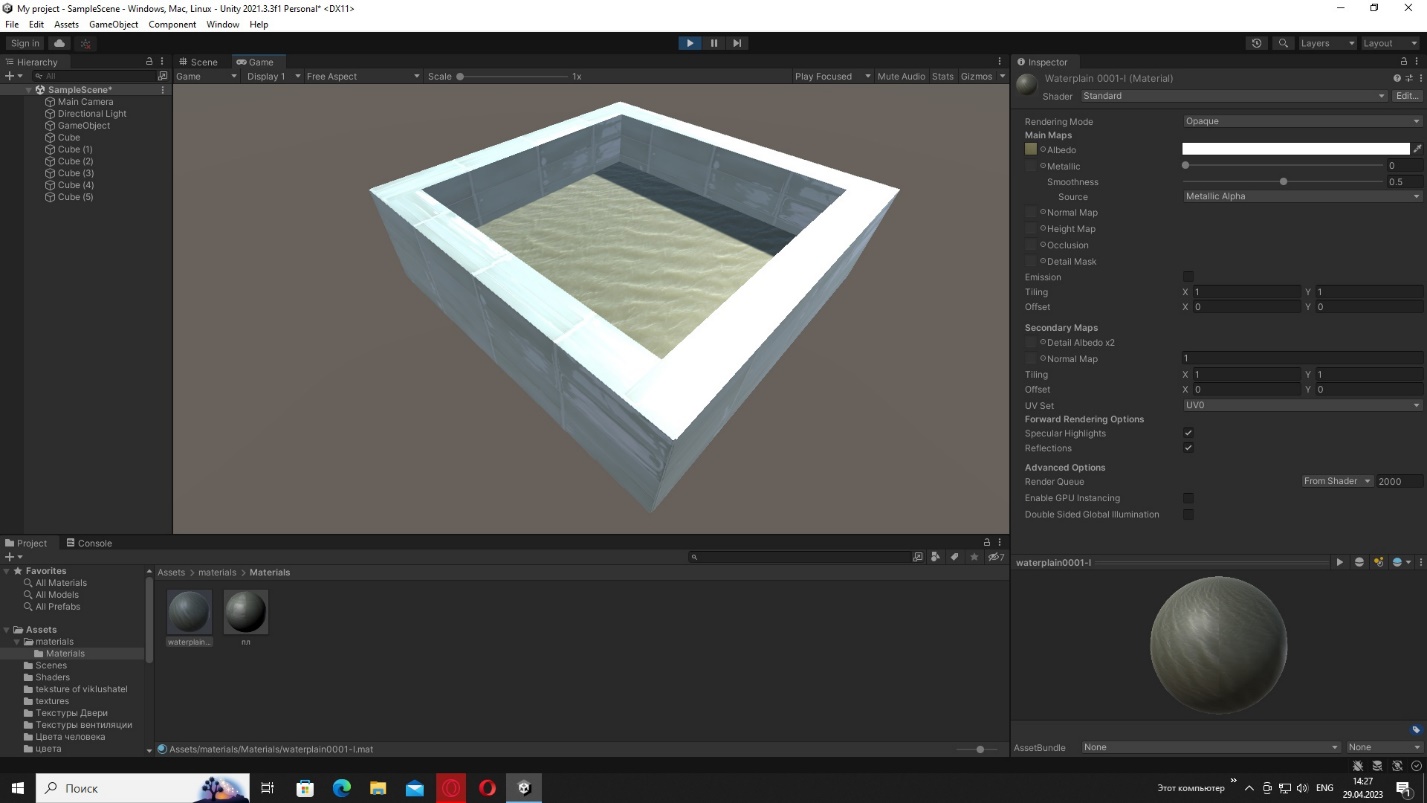
Описание работы

3.1. Написание собственной текстуры кодом.



3.2. Написание шейдера анимации воды.

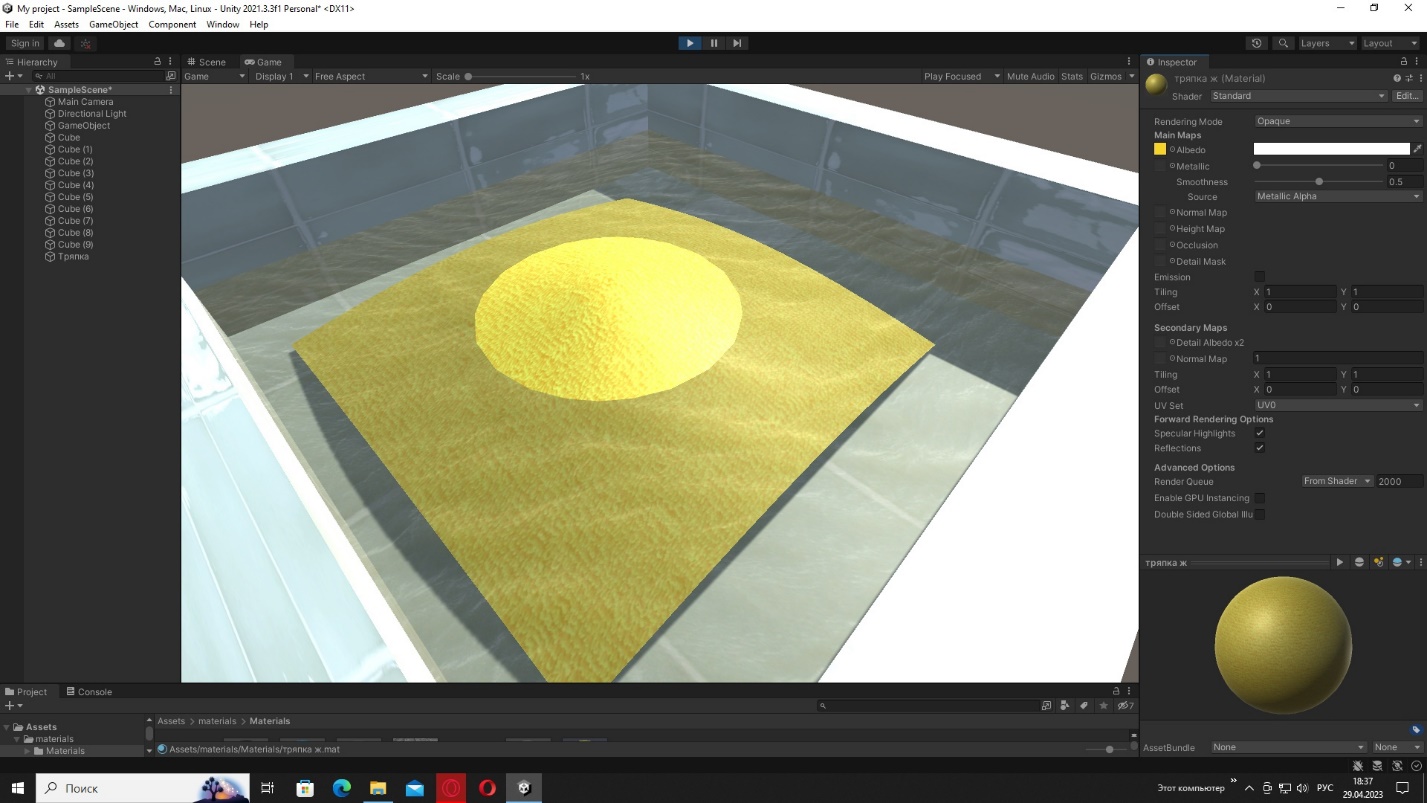
Ниже можно видеть скриншот воды до применения шейдера и видео после. Вода стала прозрачной, анимированной.

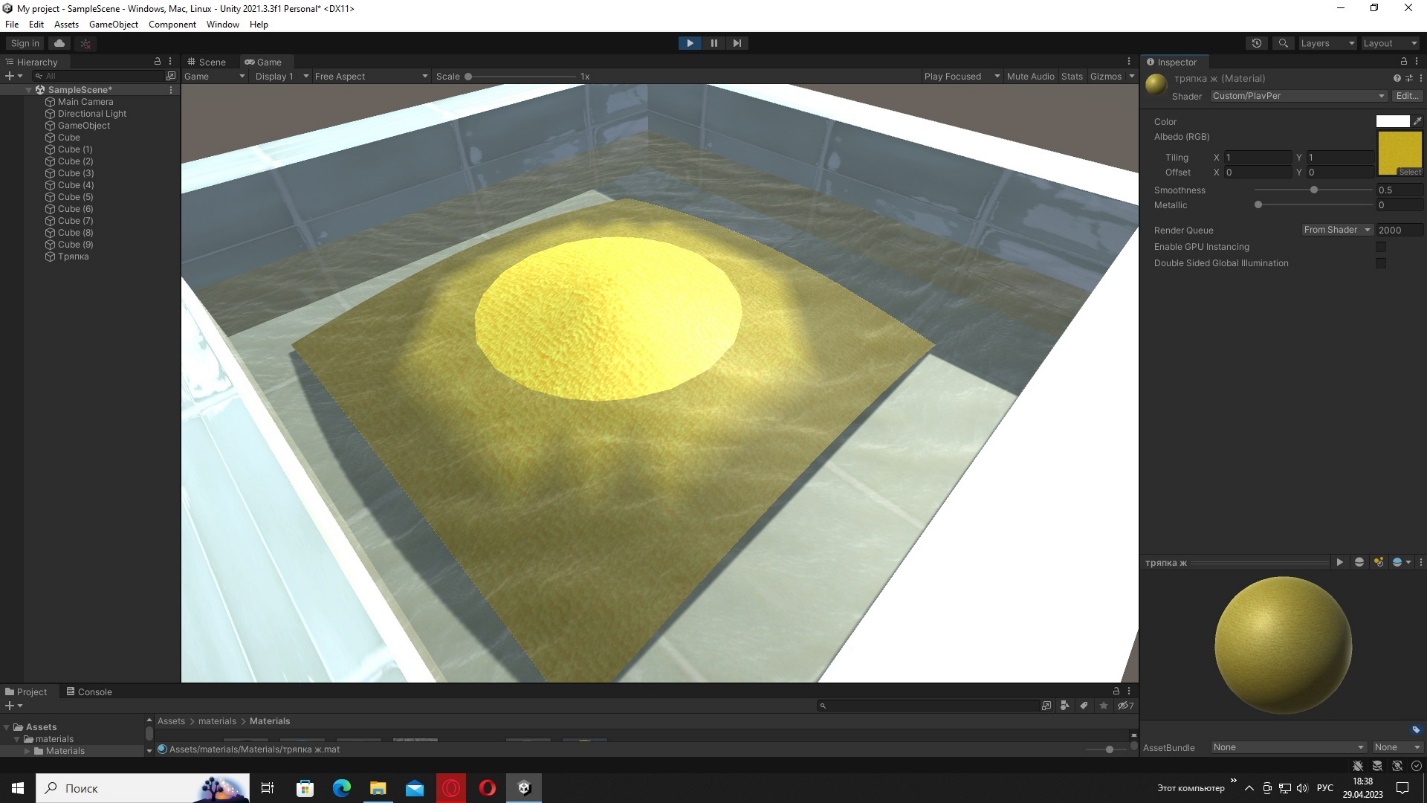




3.3. Написание шейдера более естественного и более плавного вида погружения объекта в воду.

На скриншотах ниже можно видеть тряпку, частично погружённую в воду, до и после наложения шейдера. Тряпка на глубине под водой после наложения шейдера затемнена и не по идеальной границе, что делает погружение более естественным и плавным.





Выводы

По итогу работы над шейдерами на языке программирования cg постановленные задачи были выполнены, но далеко не самым продуктивным образом. В ходе изучения и работы с шейдерами, я узнал, что существуют способы их создания в специальных программах без написания кода, встроенными модулями. Это и быстрее, и требует меньше знаний (не учить новый язык программирования), и нагляднее (мы передаём все параметры не через код, а меняем текстуру и эффекты в реальном времени через огромное множество встроенных модулей).