**Практическая работа №1**

**Цель практической работы:**

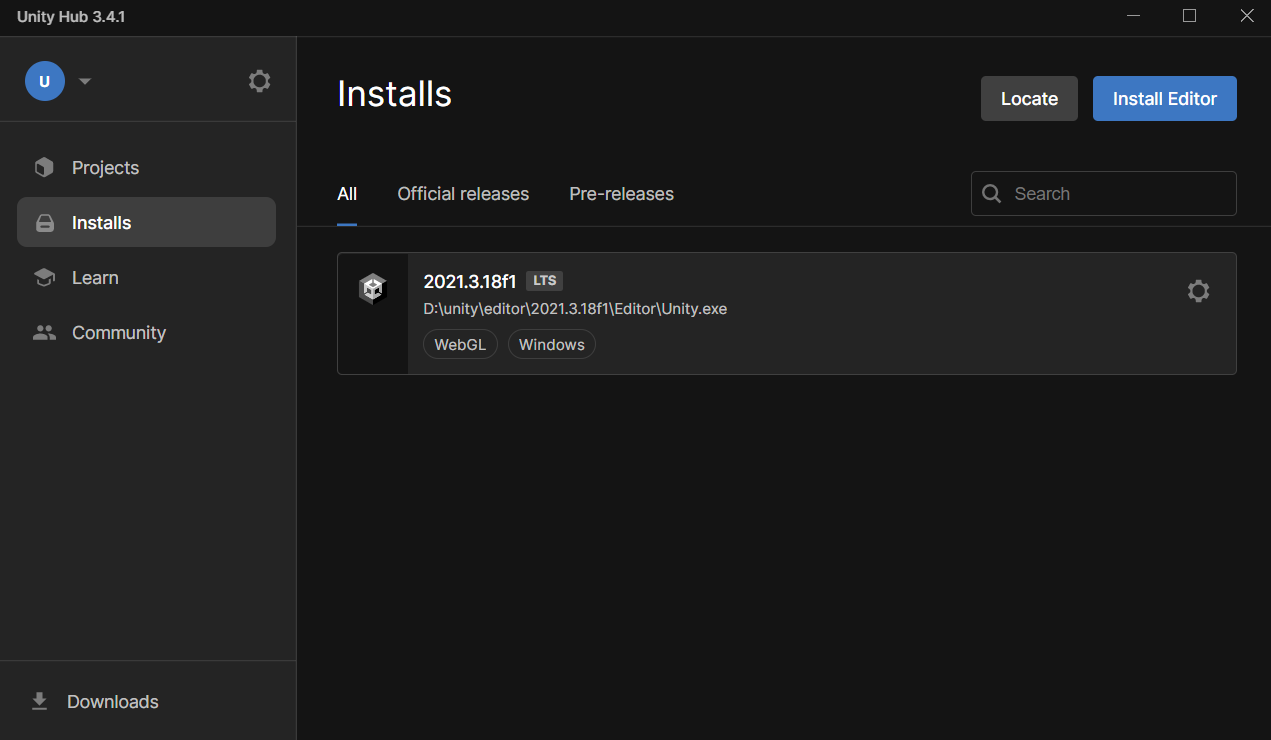
Изучить и настроить интерфейс движка Unity.

**Задачи:**

* Установить Unity и Unity Hub;
* Изучить основные элементы интерфейса Unity;
* Настроить пользовательский интерфейс;
* Разобрать основной функционал.

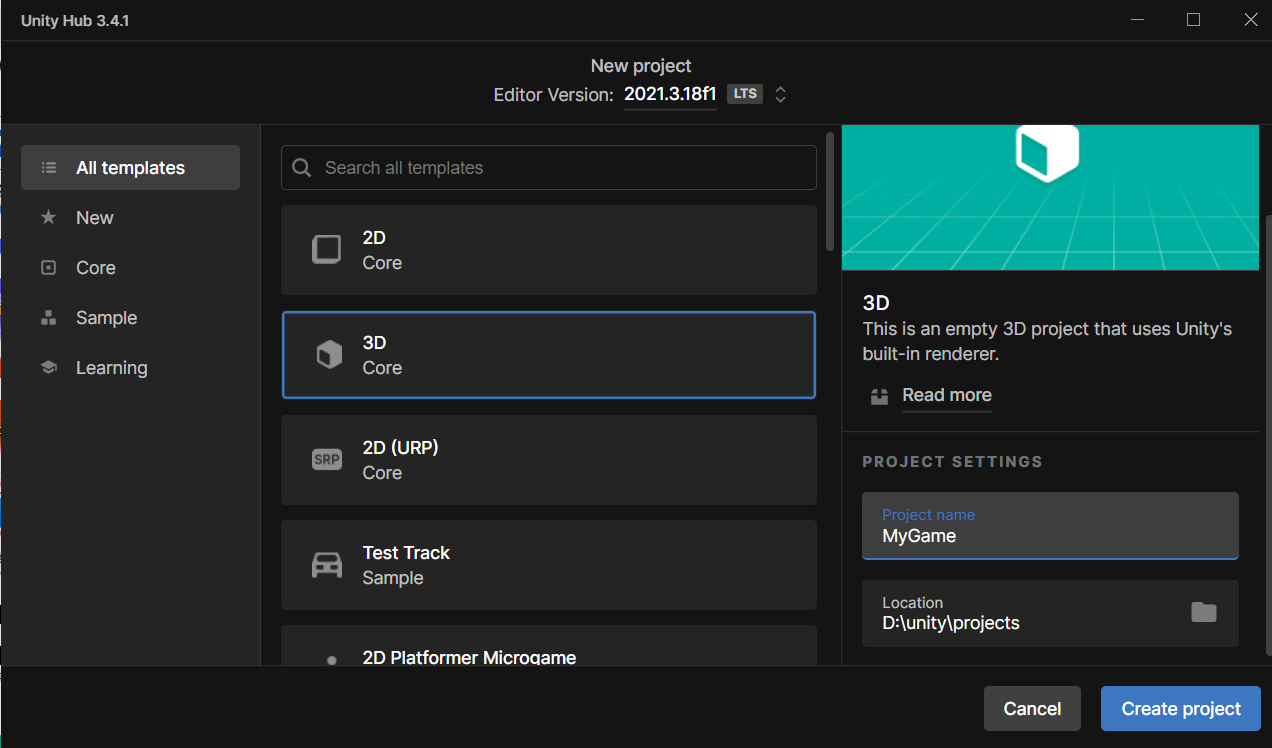
**Ход выполнения:**

Перед началом работы скачаем и установим Unity и Unity Hub с официального сайта: <https://unity.com/ru> (см. рисунок 1)



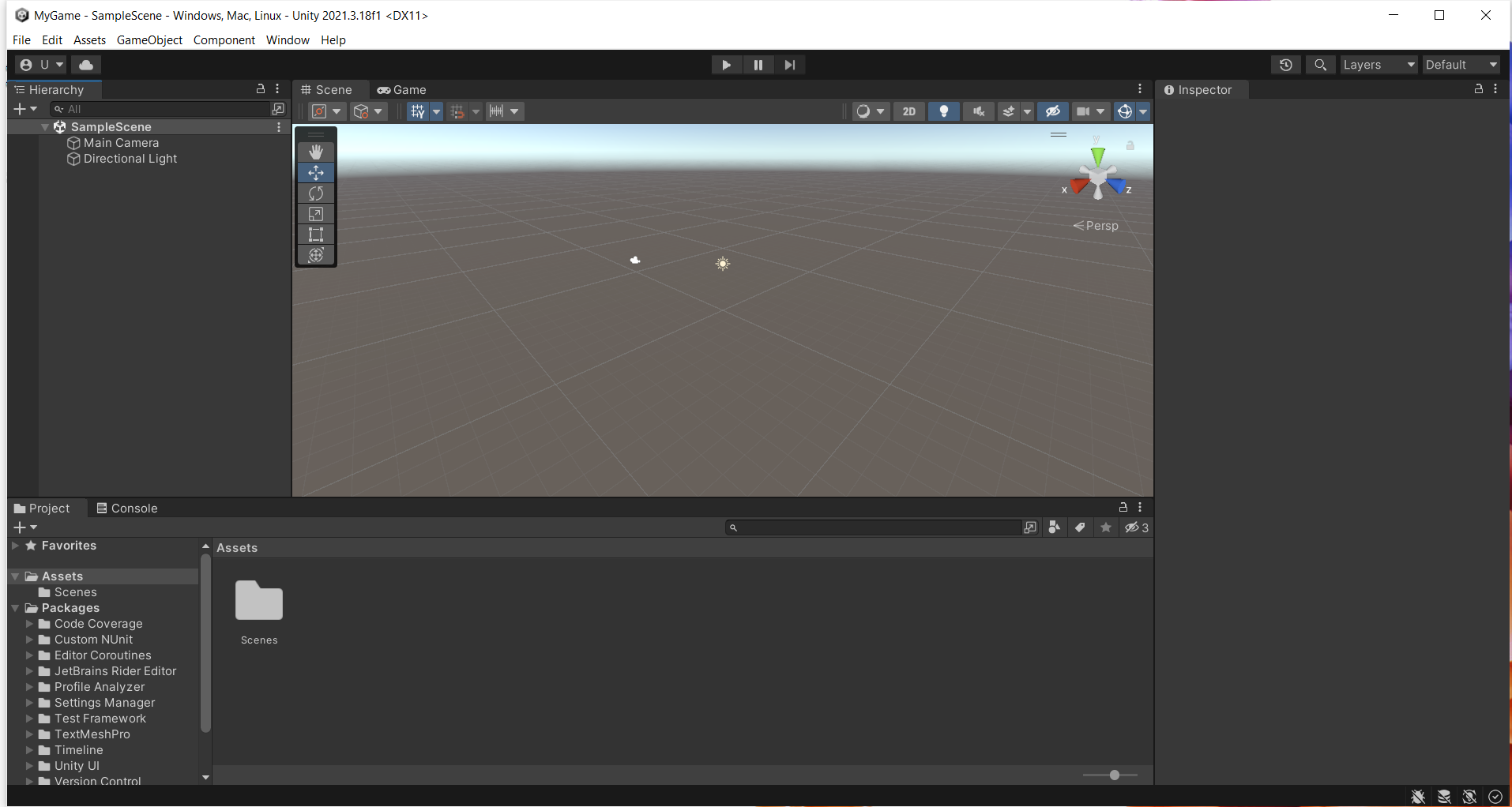
*Рисунок 1 – Установленная программа Unity*

Во вкладке Projects создадим новый проект через кнопку NEW, Template выберем 3D, дадим произвольное наименование проекта и нажмём кнопку Create (см. рисунок 2).



*Рисунок 2 – Создание нового проекта*

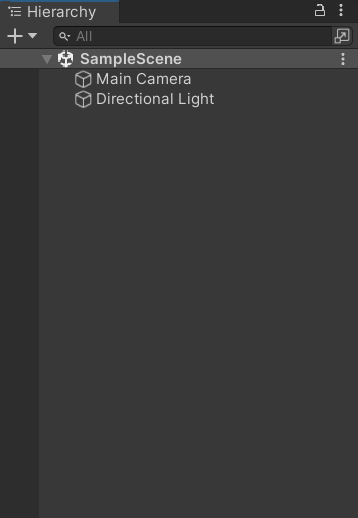
После создания проекта откроется редактор Unity (см. рисунок 3).



*Рисунок 3 – Редактор Unity*

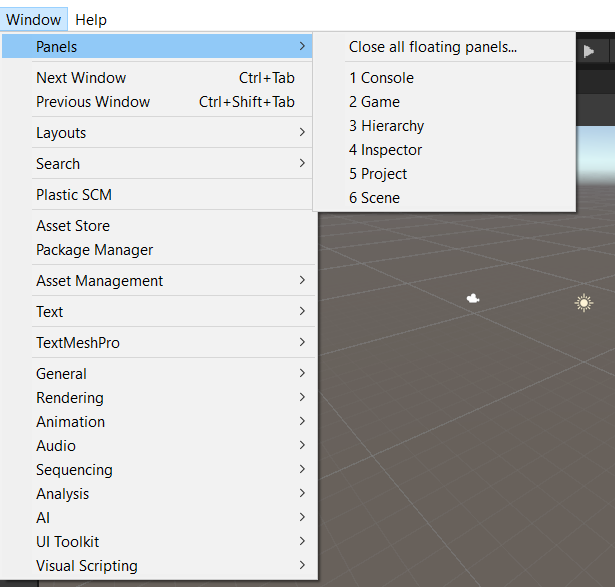
Изучим основные элементы интерфейса.

Hierarchy (Иерархия) находится в левом углу экрана (см. рисунок 4).



*Рисунок 4 – Иерархия*

Так как в проекте открыты не все окна нажмём кнопку: «Window», чтобы посмотреть их доступный список (см. рисунок 5).



*Рисунок 5 – Доступный список окон*

Рядом с окном иерархии находится панель с инструментами для работы с объектами: перемещение объекта, поворот объекта, масштабирования объекта и т.д. (см. рисунок 6).



*Рисунок 6 – Панель с инструментами*

Рядом с панелью находятся две вкладки со сценами, которые можно переключать (см. рисунок 7):

* Scene – отображаемая сцена;
* Game – здесь будет отображаться сцена при ее запуске;



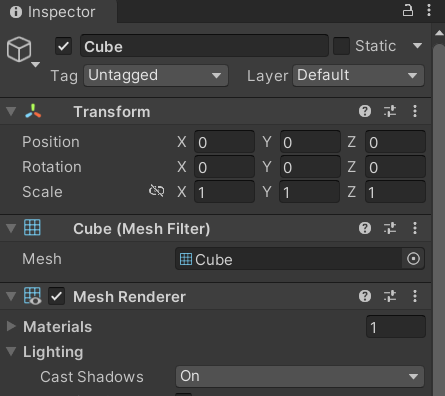
*Рисунок 7 – Вкладки со сценами*

Над окном со сценой находятся кнопки, с помощью которых можно запустить или остановить игру (см. рисунок 8).



*Рисунок 8 – Кнопки запуска и остановки игры*

В правом углу экрана находится Inspector, здесь отображаются все компоненты и их свойства, прикрепленные к объекту (см. рисунок 9).



*Рисунок 9 – Свойства объекта*

В нижней части экрана расположены окна Project и Console (см. рисунок 10).



*Рисунок 10 – Окна Project и Console*

Окно Project предназначено для эффективной работы с файлами проекта, именно там будут отображаться все файлы, которые разработчик добавил в папку Assets. Файлы из Asset Store автоматически попадают в папку Assets.

В окне Console будут отображаться все системные сообщения: предупреждения, ошибки, дебаги.

**Практическая работа №2**

**Цель практической работы:**

Изучить объекты и их настройки на сцене.

**Задачи:**

• Создать различные 3D объекты;

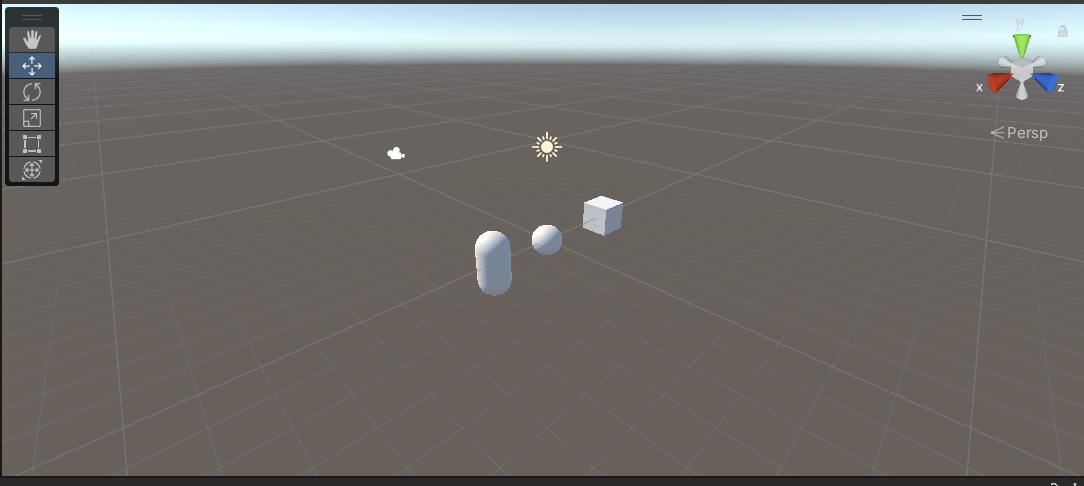
• Изучить их параметры;

• Импортировать 3D модель из Blender;

• Создать и наложить материал.

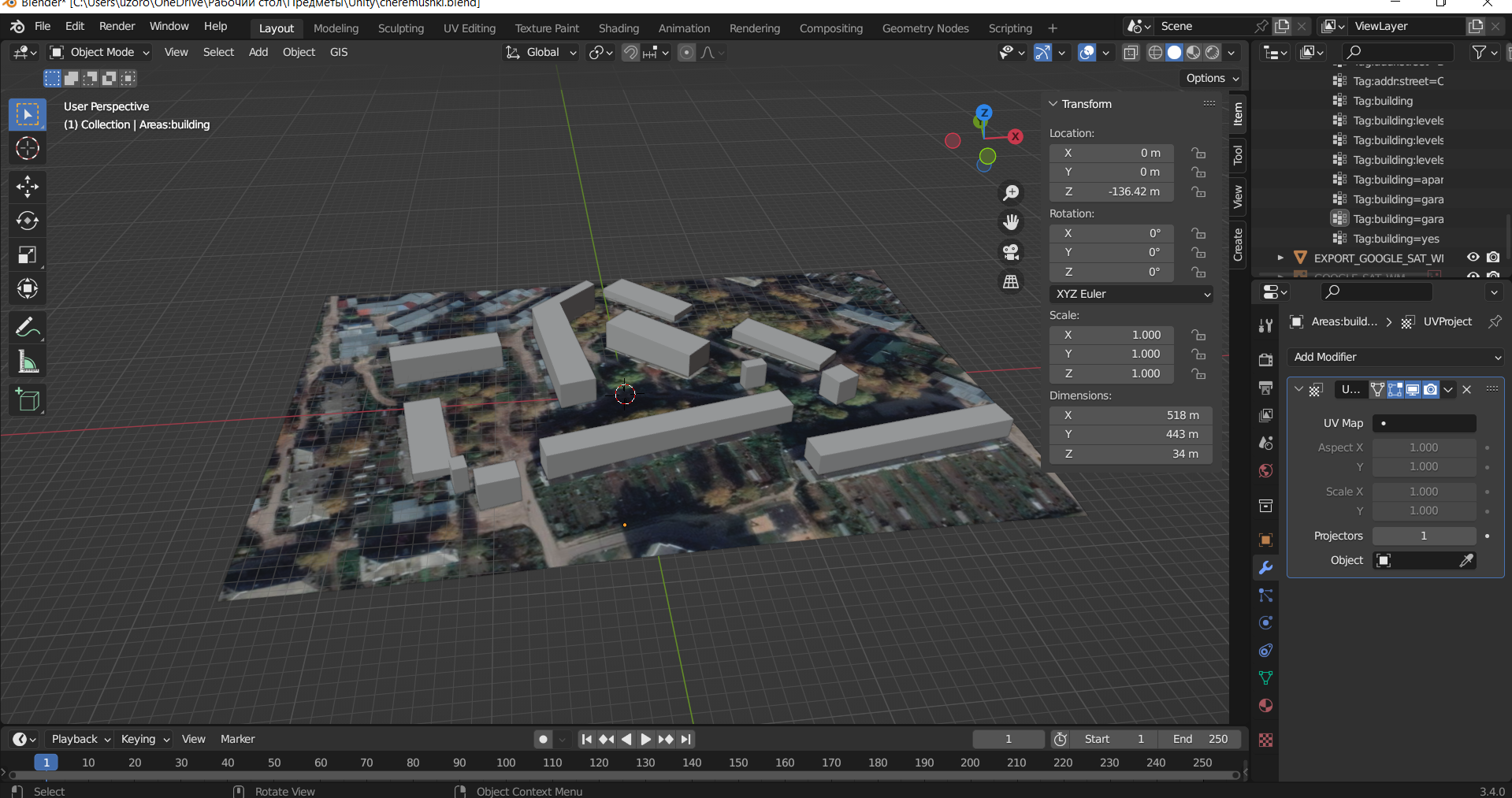
**Ход работы:**

Создадим новые объекты. Для этого в окне иерархии с помощью правой кнопки мыши вызовем контекстное меню, выберем из него 3D Object и нажмём на Cube, потом на Sphere и на Capsule, после чего передвинем их (см. рисунок 1).



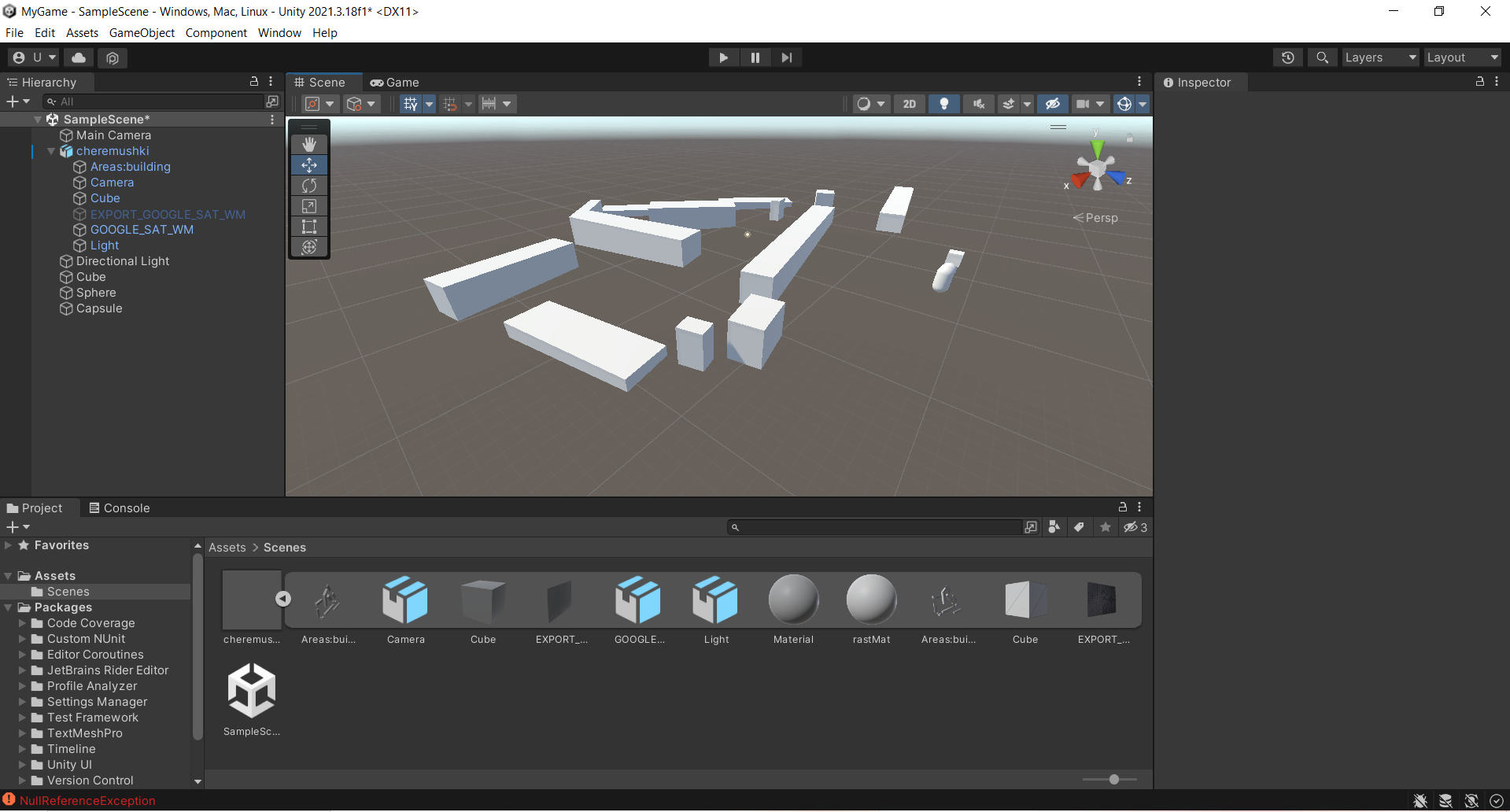
*Рисунок 1 – Созданные объекты*

Теперь создадим 3D объект в Blender, так как темой нашего проекта является создание района города, создадим его макет (см. рисунок 2).



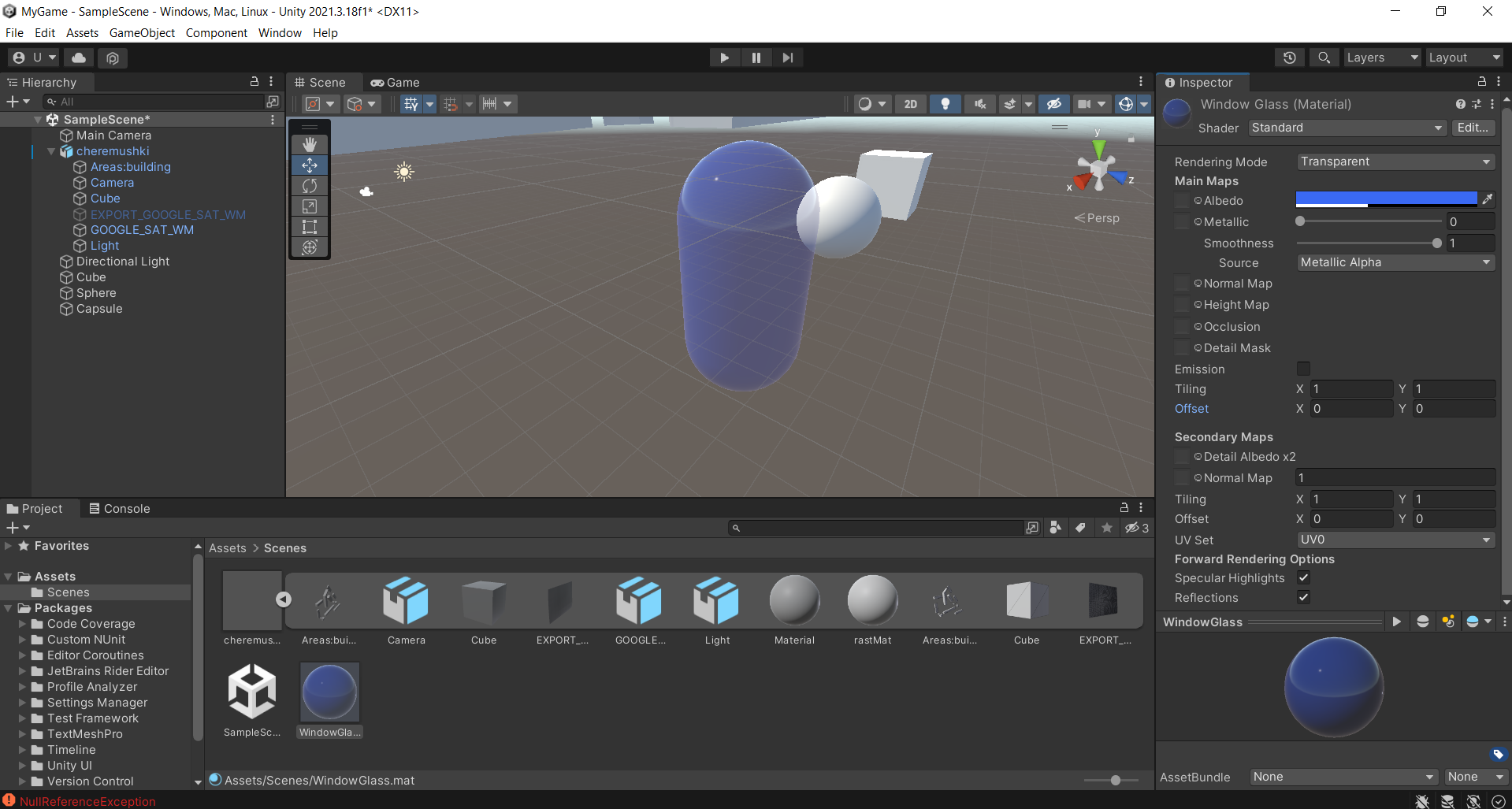
*Рисунок 2 –Макет района в Blender*

После применения модификатора UVW Project выполним экспорт модели в формате fbx. Сохранённый файл переносим в окно Assets Unity. Автоматически создается Prefab. С зажатой мышкой вынесем prefab на сцену, после чего объект появится на сцене (см. рисунок 3).



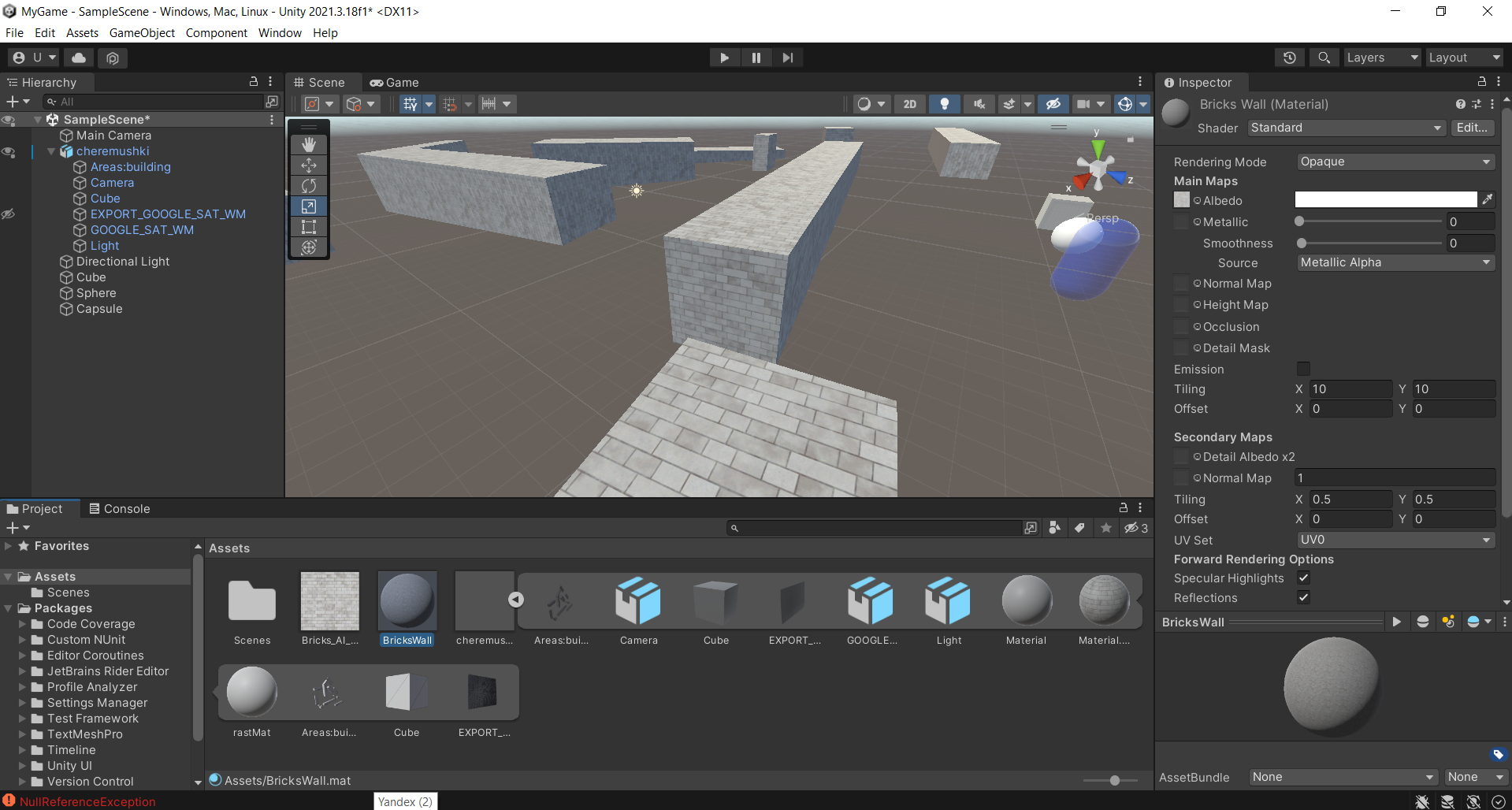
*Рисунок 3 – Добавление модели на сцену*

Теперь создадим материалы для объектов. Нажмём правую клавишу мыши и добавим материал с названием WindowGlass. Определим цвет материала, а также настроим прочие параметры, чтобы в результате получить стекло (см. рисунок 4).



*Рисунок 4 – Созданный материал*

После создания стекла добавим текстуру кирпичной стены. Для этого создадим новый материал под названием BrickWall. В карту Albedo поставим карту текстуры кирпичной стены, взятой из открытых источников. Отрегулируем параметры, чтобы получить наилучшее отображение имеющейся текстуры (см. рисунок 5).



*Рисунок 5 – Импортированный материал*

**Практическая работа №3**

**Цель практической работы:**

Изучить базовые команды на языке C#, принцип работы и применение скриптов в Unity.

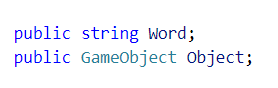
**Задачи:**

* Изучить переменные, что из себя представляют и что они хранят;
* Изучить методы, Start(), Update();
* Написать скрипт для объекта;
* Применить скрипт к объекту;
* Показать преподавателю.

**Ход работы:**

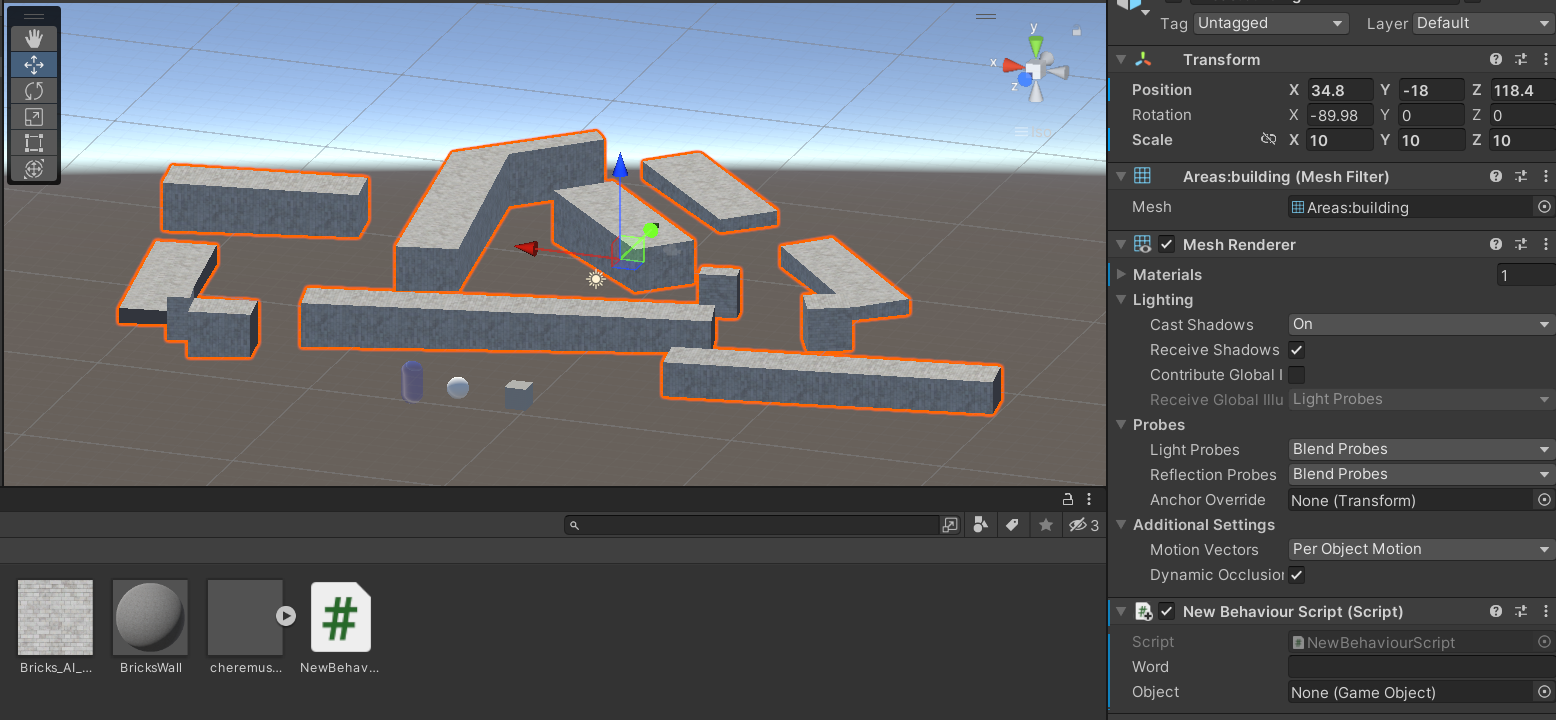
Создадим C# скрипт (ПКМ в окне Assets, Create => C# Script) и откроем его в Microsoft VisualStudio, для этого нажимаем правой кнопкой мыши по скрипту в папке Assets и выбираем в выпадающем меню кнопку Open.

Создадим несколько переменных с уровнем доступа Public, одна будет типа string, другая типа GameObject (см. рисунок 1).



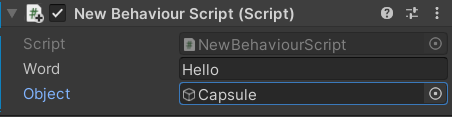
*Рисунок 1 – Созданные переменные*

Сохраним изменения (Ctrl+S), вернемся в окно Unity и перетащим полученный скрипт в окно инспектора объекта, созданного в предыдущей практической работе. В результате в инспекторы будут отражаться переменные, объявленные в скрипте (см. рисунок 2).



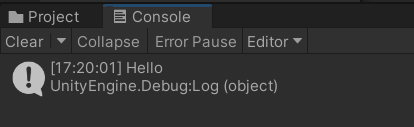
*Рисунок 2 – Отображение переменных*

Теперь зададим значения переменных. Первая переменная будет хранить строку, а вторая объект со сцены (см. рисунок 3).



*Рисунок 3 – Переменные с заданным значением*

Поменяем модификатор доступа строковой переменной на private и выведем её значение в лог при помощи команды: Debug.Log(Word) (см. рисунок 4).



*Рисунок 4 – Вывод значения переменной в лог*

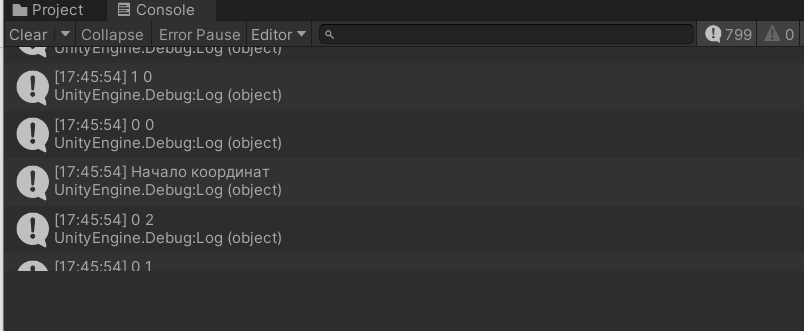
Создадим метод RandomGeoData (см. рисунок 5).



*Рисунок 5 – Метод RandomGeoData()*

Данный метод генерирует случайные координаты X и Y, а также выполняет проверку на заданное условие – если X и Y равны нулю, в консоль будет выводиться: «Начало координат».

Результат работы метода показан на рисунке 6.



*Рисунок 6 – Результат работы метода RandomGeoData()*