Как пользоваться приложением:

**Установка**

* Скачать Unity Hub
* Зайти в Installs и установить версию Unity 2020.3.3f
* В Projects нажать Add и добавить папку с проектом (PhotoRenderer)
* Открыть проект

**Запуск**

* Нажать кнопку Play

**Сохранение положений модели**

* Нажать на компонент PositionSaver в Иерархии. В Испекторе появится отображение компонента. Нужно указать имя файла с которым тот сохранится (на картинке имя «1») и Model мышкой перенести нужную модель
* Установить модель нужных местах:



Это можно сделать выбрав необходимый компонент в иерархии (Model). Также нужно выбрать компонент Move Tool и двигать элемент за стрелочки на сцене. Для поворота моделей/камеры использовать компонент Rotate Tool рядом с Move Tool.

* Нажать клавишу W для сохранения позиции.
* Переместить модель на следующую позицию.
* Повторить пункты 1-3 нужное количество раз.
* Нажать клавишу A для сохранения файла с позициями.

После этого в папке Assets/Resources/ModelPositions создастся файл с указанным именем.

**Сохранение параметров камеры**

В проекте реализовано 3 варианта стратегии камеры:

* Rotate360
* WallMove
* SecurityCamera
* Для всех стратегий нужно установить начальное положение камеры. Для этого нужно выбрать компонент СameraStrategy в иерархии. Также нужно выбрать компонент Move Tool и двигать элемент за стрелочки на сцене. Для поворота камеры в иерархии есть удобный компонент - Target. Камера настроена так, что постоянно смотрит на этот объект. Поэтому двигая его за стрелочки можно настроить поворот камеры. Все это будет стартовая позиция камеры.
* Выбрав компонент СameraStrategy, в Испекторе появится отображение компонента. В параметрах файла нужно выбрать стратегию и указать имя файла, под которым он сохраниться.
* В инспекторе есть поля для параметров каждой стратегии.
* Rotate360: Нужно указать количество фотографий (если например 4, то камера сфотографирует одно положение с 4-х точек вращаясь по 360/4 = 90 градусов)
* WallMove: Нужно указать величину шага и количество шагов. Величину шага можно определить двигая камеру вверх-вниз и смотря на изменения в инспекторе.
* SecurityCamera: Нужно указать угол поворота и сколько раз на него поворачивать (во все четыре стороны.
* Нажать клавишу E для сохранения данных
* Нажать клавишу S для сохранения файла со стратегией.

После этого в папке Assets/Resources/CameraParameters создастся файл с указанным именем.

**Настройка файла конфигурации**

В папке Assets/Resources/ConfigurationFile лежит файл с настройками



В нем нужно указать имена файлов, которые нужно подгрузить. Там же лежит файл с примером для нескольких сцен-моделей.

**Запуск рендеринга**

* Нажать пробел

После этого в папке проекта OutPut будет созданы (если уже не создан) датасет с именем Dataset. Если папка с таким именем уже существует, ничего не произойдет

**Завершение:**

* Нажать кнопку Play еще раз

**Создание новых префабов**

* Скачать модель в папку models/environment
* Перетащить мышкой модель на сцену
* Нужную текстуру перетащить мышкой на объект на сцене
* Выбрать модель, развернуть компонент material в инспекторе, в поле normal map добавит карту нормалей (если есть)
* Можно подвигать ползунки metallic и smoothness для создания большей реалистичности
* Если это модель человека, то в инспекторе выбрать Layer «Model»



* Выбрать объект в иерархии и перетащить его в папку Resourсes/… Во всплывающем окне выбрать создать новый префаб

Архитектура и компоненты кода:

* **RenderLogic**

Компонент, отвечающий за всю логику процесса. В нем находятся 2 ссылки на другие объекты сцены (Cameras и DataSaver), а также поля для подгружаемых данных: 5 полей для данных (модели, комнати, свет, файы позиций, файлы параметров камеры) и поле для файла конфигурации.

Основные функции:

* BuildAllPrefabs() – Подгружает все данные и модели. Вызывается клавишей «B».
* RenderDataset() – Запускает рендер. При нажатии на пробел последовательно вызываются функции BuildAllPrefabs() и RenderDataset(). Логика функции следующая: циклом проходим по всем кофигурациям, в каждой конфигурации проходим циклом по всем позициям модели, и для каждой позиции делаем заданное количество рендеров. Итоговое количество рендеров равно:

Кофигурации \* позиции модели \* рендеры в стратегии камеры

* SetParameters() – принимает на вход переменные циклов и задает положение текущей модели и камеры
* SaveLogFile() – сохраняет лог файл
* **CamerasController**

Компонент находится на объекте Cameras и отвечает за рендер всех разметок. В нем находится массив всех камер, которые участвуют в процессе. Если нужно отключить рендер какой-либо разметки – удалите эту камеру из списка.



Основные функции:

* MakeRender() - запускает рендер камер. Вызывется из RenderDataset(). На вход принимет индекс текущего рендера
* LimitationCheck() - Проверяет на ограничения видимости модели и пересечений.

Каждая камера имеет свой индивидуальный скрипт и шейдер для рендера нужной разметки, однако все они наследуются от общего класса PhotoRender. В нем есть функция ChangeBool() - которая дает другой функции GetTexture() сигнал сохранить текущий рендер. Для всех камер есть общее поле, в которое прописано название для каждой разметки.