В качестве входных данных предоставляется:

1. Модель ноги в формате obj
2. Файл в формате txt с 4 координатами углов листа бумаги

Примечание 1: нет необходимости реализовывать загрузчик obj файла или текстового файла – достаточно включить obj в asset-ы, а координаты точек перенести на сцену вручную.

Задача.

Необходимо разработать программу, которая на предоставленном объекте построит сечение стопы на высоте 10 мм от поверхности листа бумаги. Для расчета масштаба (высоты) следует использовать расстояние между углами листа, зная его размер (210 х 297).

Под сечением понимается последовательный набор точек/отрезков, проходящих через все полигоны на указанной высоте и образующих замкнутую ломаную линию.

Примечание 2: mesh состоит исключительно из треугольников. Закладываться на обработку полигонов, состоящих из большего количества точек, нет необходимости.

\* Дополнительное задание: построить сечение на высоте 10 мм не от плоскости, проходящей через точки углов листа бумаги, а от плоскости пола (края листа бумаги могут быть приподняты над полом). Также следует учитывать, что на высоте 10 мм могут быть другие полигоны, не относящиеся к ноге (не связанные/не примыкающие к ней), например, тот самый приподнятый край листа бумаги. Они не должны входить в сечение стопы.

Результат.

В качестве результата необходимо отобразить на сцене саму 3D модель и объект Line Renderer, содержащий весь последовательный набор точек сечения (отражающий обхват модели на указанной высоте).

Ниже приведен простейший пример перебора всех координат точек треугольных полигонов.

Mesh mesh = model.GetComponent<MeshFilter>().mesh;

Vector3[] verts = mesh.vertices;

int[] triangles = mesh.triangles;

for (int i = 0; i < triangles.Length; i += 3)

{

int i1 = (i + 0);

int i2 = (i + 1);

int i3 = (i + 2);

Vector3 v1 = verts[triangles[i1]];

Vector3 v2 = verts[triangles[i2]];

Vector3 v3 = verts[triangles[i3]];

}