

Actualmente el programa que se creó dibuja las líneas en una sola cara, es decir un triángulo, obteniendo así 4 triángulos dentro de la cara para el teselado.

Para esto, se utiliza la topología y geometría de la cara en la función del teselado, sin embargo esta función solo funciona para una única cara, mientras que la intención es permitir que funcione para todas las caras de un dodecaedro.

Para que esta función cree el teselado correctamente, se utilizan las tres caras para obtener los puntos A, B y C, por lo que es requerido realizar los cálculos de o, p y q con tres puntos únicamente. Tomando esto en cuenta, la solución que considero podría ser de utilidad para obtener estos tres puntos para cada cara es que la geometría se vuelva un array de arrays de objetos tipo `Vector3`, de esta forma la geometría quedaría como un array de 8 arrays de `Vector3`, representando cada uno los vectores de cada cara.

Utilizando esta implementación, la idea es que al momento de realizar los cálculos para A, B, C, o, p, q, se hagan por cada uno de esos arrays, obteniendo así 8 valores para cada uno de ellos, los cuales con cada iteración se agregaran a una lista de geometría con la función `list.add()`. Esta lista contendrá un total de 48 vectores representando los valores de A, B, C, o, p y q, para cada cara del dodecaedro. Posteriormente los datos de esta lista se agregan al array de `resultG`, almacenando la geometría de las 8 caras y de su teselado.

De esta manera se podría generar el teselado para cada cara del dodecaedro solo mediante la división de sus 8 caras en arrays dentro de un mismo array.