Taller

Guía

- La idea de este taller es trabajar sobre los conceptos desarrollados en clase de manera práctica.
- Pueden usar su lenguaje de programación favorito, se dan ejemplos en Python.

En este taller vamos a generar un Doubly Connected Edge List (DCEL) a partir de una una lista de segmentos. La estructura DCEL debe tener vertices, half-edges, y faces.

Paso 1: Definir las clases

Los objetos necesarios dentro de la estructura DCEL son `Vertex`, `HalfEdge`, y `Face`. El siguiente es un ejemplo en Python de la clase `Vertex`

```
class Vertex:
    def __init__(self, vertex_id, coordinates):
        self.coordinates=coordinates
        self.id = vertex_id
```

En los objetos `HalfEdge`, y `Face` qué información debemos tener? Cómo creamos estas estructuras?

Paso 2: Estructuras adicionales

Una manera de llevar registro de los `Vertex` es crear un diccionario con llave el respectivo identificador y sus correspondientes coordenadas, con esta estructura cómo se puede crear un identificador único para cada half edge sin crear más datos?

Paso 3: Funciones necesarias

Dado un `HalfEdge`, e,

Problema 1: Es necesario poder determinar cuál sería *Next(e)*, un criterio es el half-edge que empieze en su `Vertex` destino con el que forme un menor ángulo, cómo lo implementamos?

Problema 2: Cómo podríamos identificar la componente conexa que genera el camino de un `HalfEdge` y cómo determinamos si hace parte de un camino de una frontera interior (`InnerComponet`) o de una frontera exterior (`ExteriorComponent`)?

Problema 3: Con las componentes conexas identificadas en la solución al problema anterior, cómo identificamos la correspondiente `Face`, f, y asignamos `InnerComponet` y `ExteriorComponet` para f.

Paso 4: Juntando toda la estructura

Run the DCEL initialization function with a set of line segments.

```
segments = [((0, 0), (-4, -1)), ((-4, -1), (-2, 4)),((0,0),(-2,4)), ((0,
    0), (3, 4)), ((3, 4), (1, 6)),((7, 6), (1, 6)),((7, 6), (5, 0)),((0,
    0), (5, 0))((3, -2), (5, 0)),((3, -2), (0, 0))]
dcel_result = initialize_dcel(segments)
```