**Visual Studio**

**How to Set up VS:**

In terminal run the following commands

1. py -m venv venv
2. .\venv\Scripts\Activate.ps1
3. py -m pip install neo4j pandas
4. python decode\_furniture.py (run this or play button on top right)

**Neo4j**

**1. Clear Old Graph**

MATCH (n)

DETACH DELETE n

**2. Import Nodes from CSV**

LOAD CSV WITH HEADERS FROM 'file:///your\_nodes.csv' AS row

MERGE (n {id: row.`id:ID`})

SET n.label = row.label,

n.type = row.type,

n.x = toFloat(row.`x:float`),

n.y = toFloat(row.`y:float`),

n.z = toFloat(row.`z:float`),

n.closest\_furniture = row.closest\_furniture,

n.apartment = toInteger(row.`apartment:int`)

WITH n, row

CALL apoc.create.addLabels(n, [row[":LABEL"]]) YIELD node

RETURN count(node)

**3. Import Edges from CSV**

LOAD CSV WITH HEADERS FROM 'file:///your\_edges.csv' AS row

MATCH (a {id: row.`source:START\_ID`})

MATCH (b {id: row.`target:END\_ID`})

MERGE (a)-[r:CONNECTED\_TO]->(b)

SET r.type = row.type

RETURN count(r)

**4. Tag Building Elements**

MATCH (n:Reference)

SET n:BuildingElement

**5. Create CLOSEST\_TO Relationships**

MATCH (f:Furniture)

WITH f

MATCH (b:BuildingElement)

WITH f, b,

point({x: toFloat(f.x), y: toFloat(f.y)}) AS p1,

point({x: toFloat(b.x), y: toFloat(b.y)}) AS p2

WITH f, b, point.distance(p1, p2) AS dist

ORDER BY f, dist

WITH f, collect(b)[0] AS closest

MERGE (f)-[:CLOSEST\_TO]->(closest)

**6. Project GDS Graph**

1. CALL gds.graph.drop('furnitureGraph', false)
2. CALL gds.graph.project(

'furnitureGraph',

['Furniture', 'BuildingElement'],

{

CONNECTED\_TO: {type: 'CONNECTED\_TO'},

CLOSEST\_TO: {type: 'CLOSEST\_TO'}

}

)

**7. Write Component IDs**

CALL gds.wcc.write('furnitureGraph', {

writeProperty: 'componentId'

})

**8. Tag Anchors**

MATCH (f:Furniture)-[:CLOSEST\_TO]->(b)

SET f.anchor = b.id,

f.anchor\_type = head(labels(b))

**9. Assign Ranks**

MATCH (f:Furniture)

OPTIONAL MATCH (f)-[:CONNECTED\_TO]->(n)

WITH f, COUNT(n) AS connections

SET f.rank = CASE

WHEN connections = 0 THEN 1

WHEN connections = 1 THEN 2

ELSE 3

END

**Go back to VS Code and run:**

You should have csv and JSON in repo