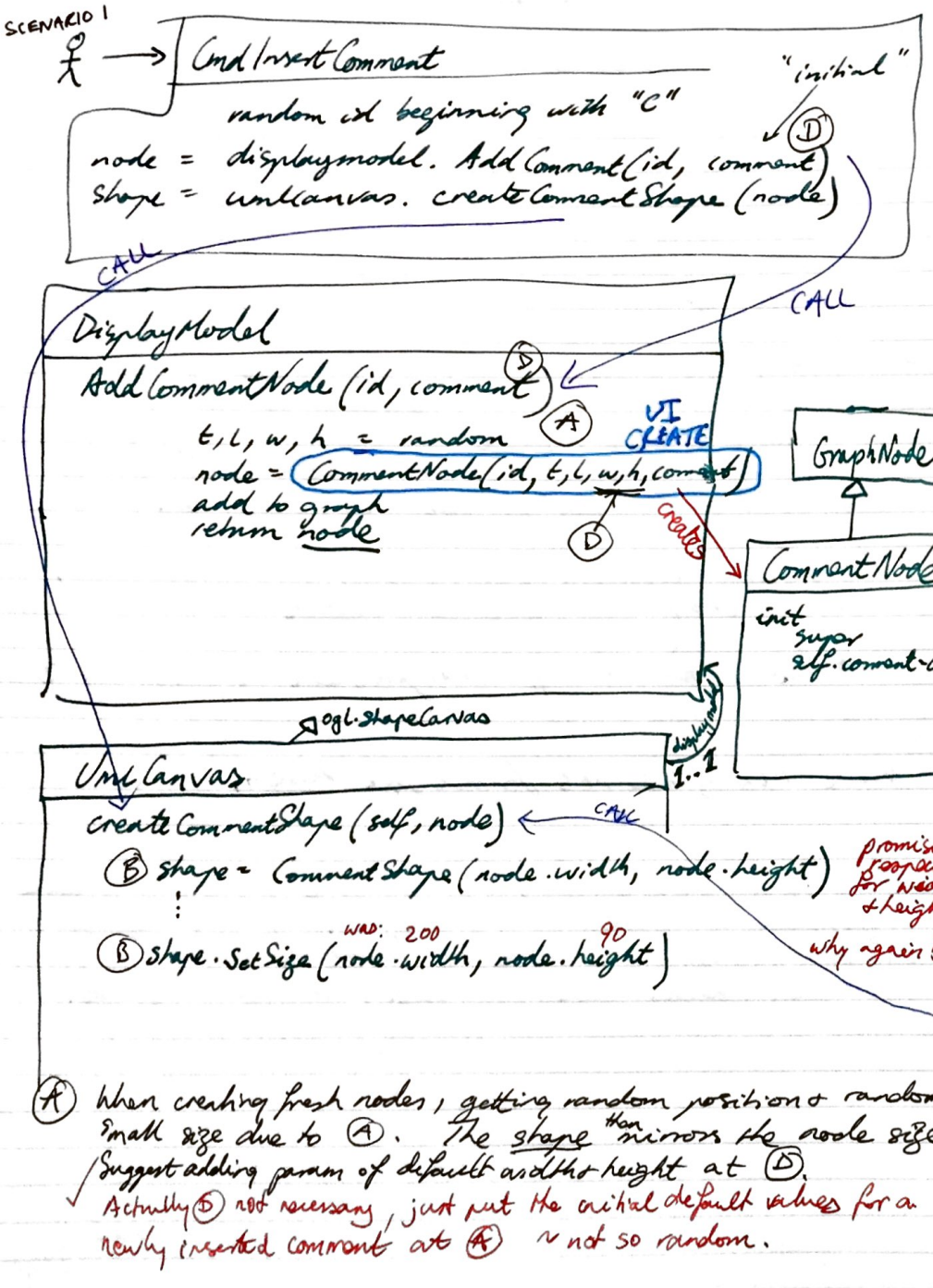
PyrSource 10 / 1 / 2019

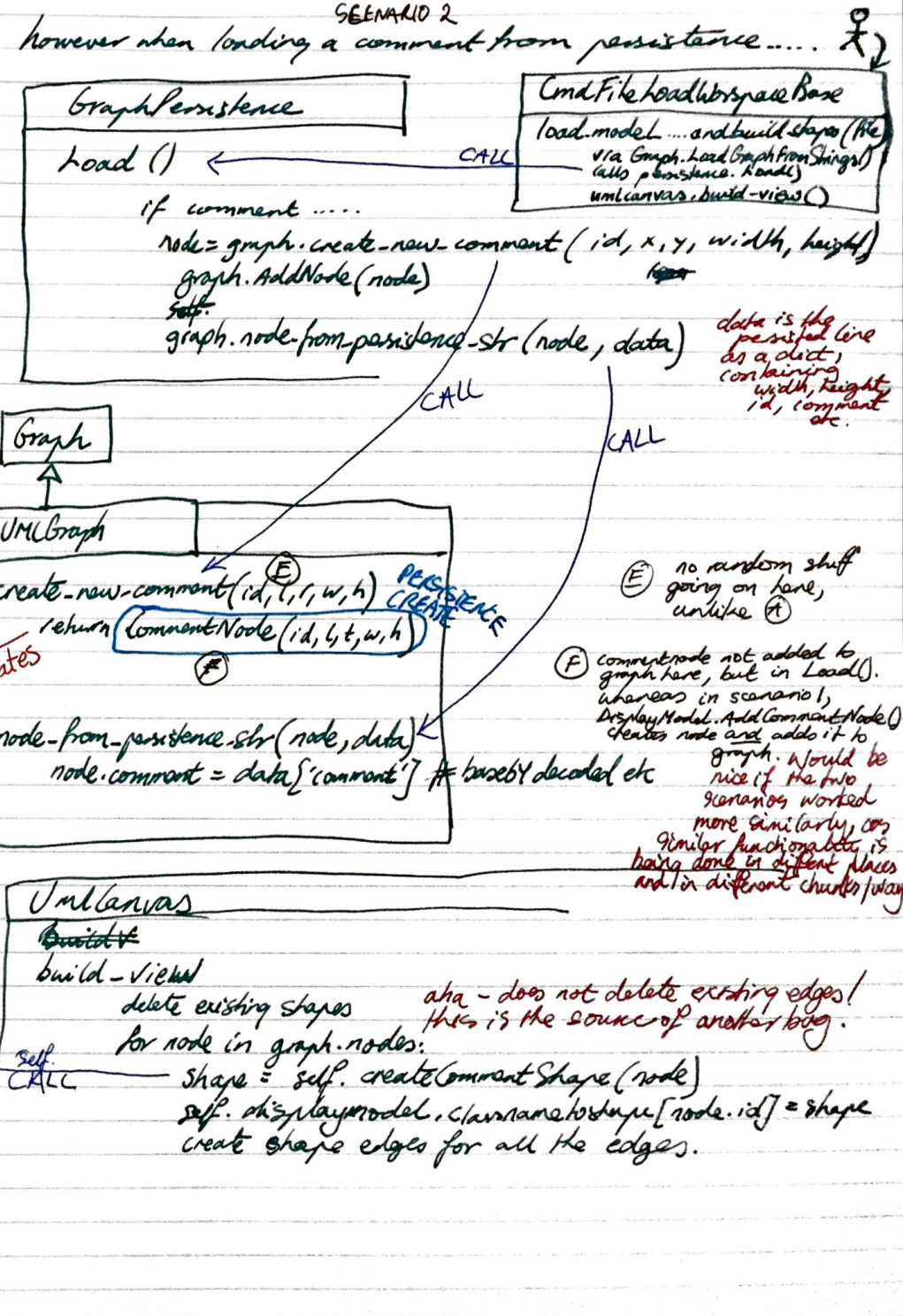
Setting size of comment &

SCENARIO 1



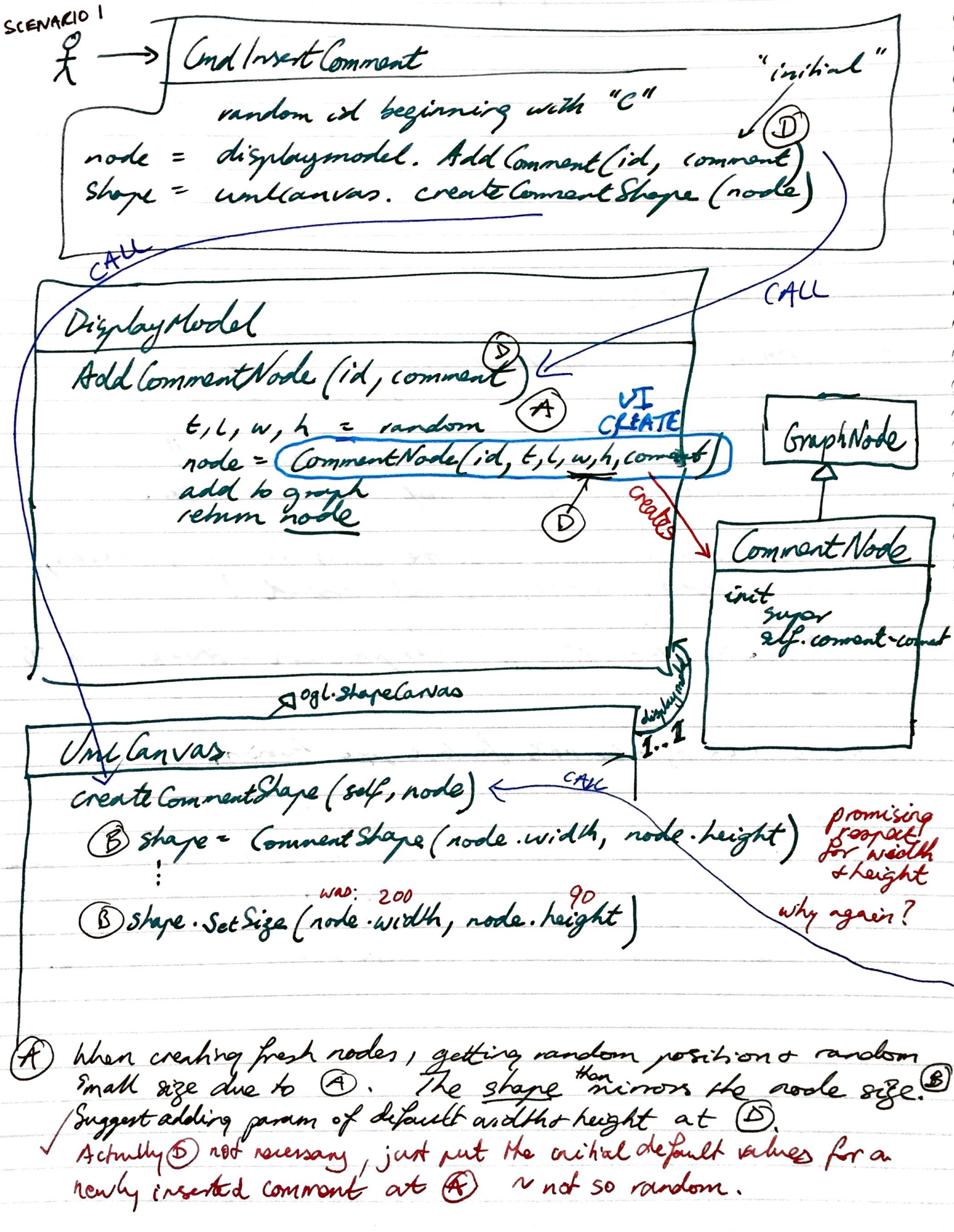
SCENARIO 2

however when loading a comment from persistence.....



Pynsource 10 / 1 / 2019

Setting size of comment &



(A) When creating fresh nodes, getting random positions + random small size due to (A). The shape than mirrors the node size. (B)

✓ Suggest adding param of default width+height at (D).

Actually (D) not necessary, just put the actual default values for a newly inserted comment at (A) ~ not so random.

SCENARIO 2

however when loading a comment from persistence.....

GraphPersistence

load()

if comment .....

node = graph.create_new_comment(id, x, y, width, height)

graph.AddNode(node)

~~self.~~ graph.node_from_persistence_str(node, data)

CALL

CmdFileLoadWorkspaceBase

load_model ....and build shapes (file via Graph.LoadGraphFromStrings() (calls persistence.load()) umlcanvas.build_view()

data is the persisted line as a dict, containing width, height, id, comment etc.

Graph

UMLGraph

create_new_comment(id, l, t, w, h) (E)

return CommentNode(id, l, t, w, h) (F)

PERSISTENCE CREATE

creates

node_from_persistence_str(node, data)

node.comment = data['comment'] # base64 decoded etc

CALL

CALL

(E) no random shuff going on here, unlike (A)

(F) commentnode not added to graph here, but in Load(). whereas in scenario 1, DisplayModel.AddCommentNode() creates node and adds it to graph. Would be nice if the two scenarios worked more similarly, as similar functionality is being done in different places and in different classes/ways

UmlCanvas

~~build()~~

build_view

delete existing shapes

for node in graph.nodes:

aha - does not delete existing edges! this is the source of another bug.

self. CALL

shape = self.createCommentShape(node)

self.displaymodel.classnametoshape[node.id] = shape

create shape edges for all the edges.