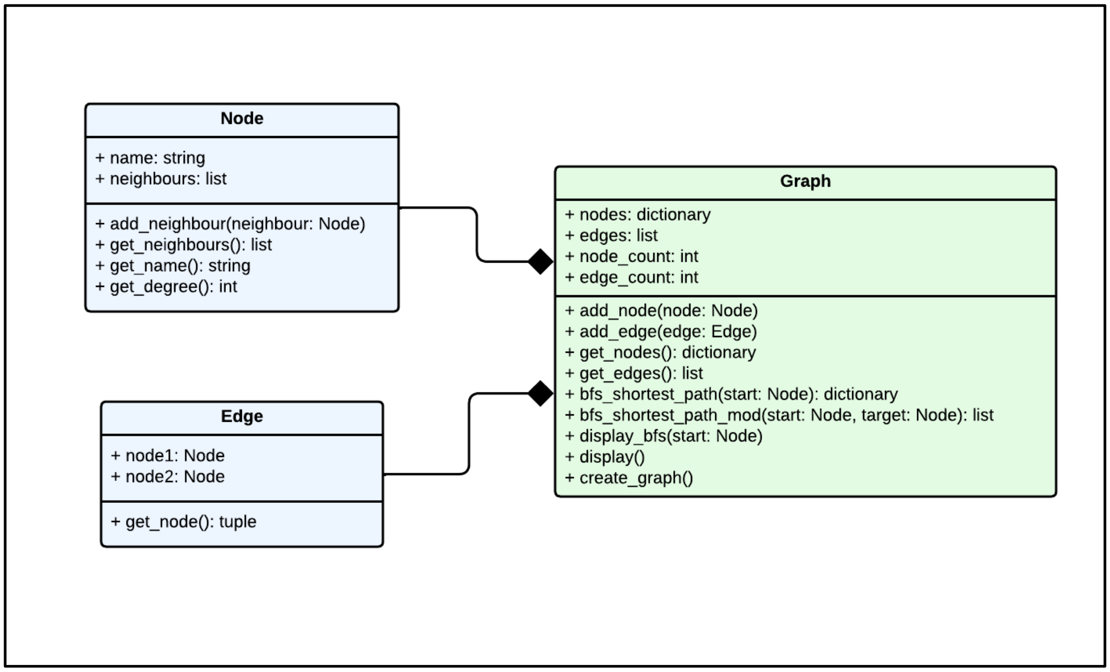
World Design

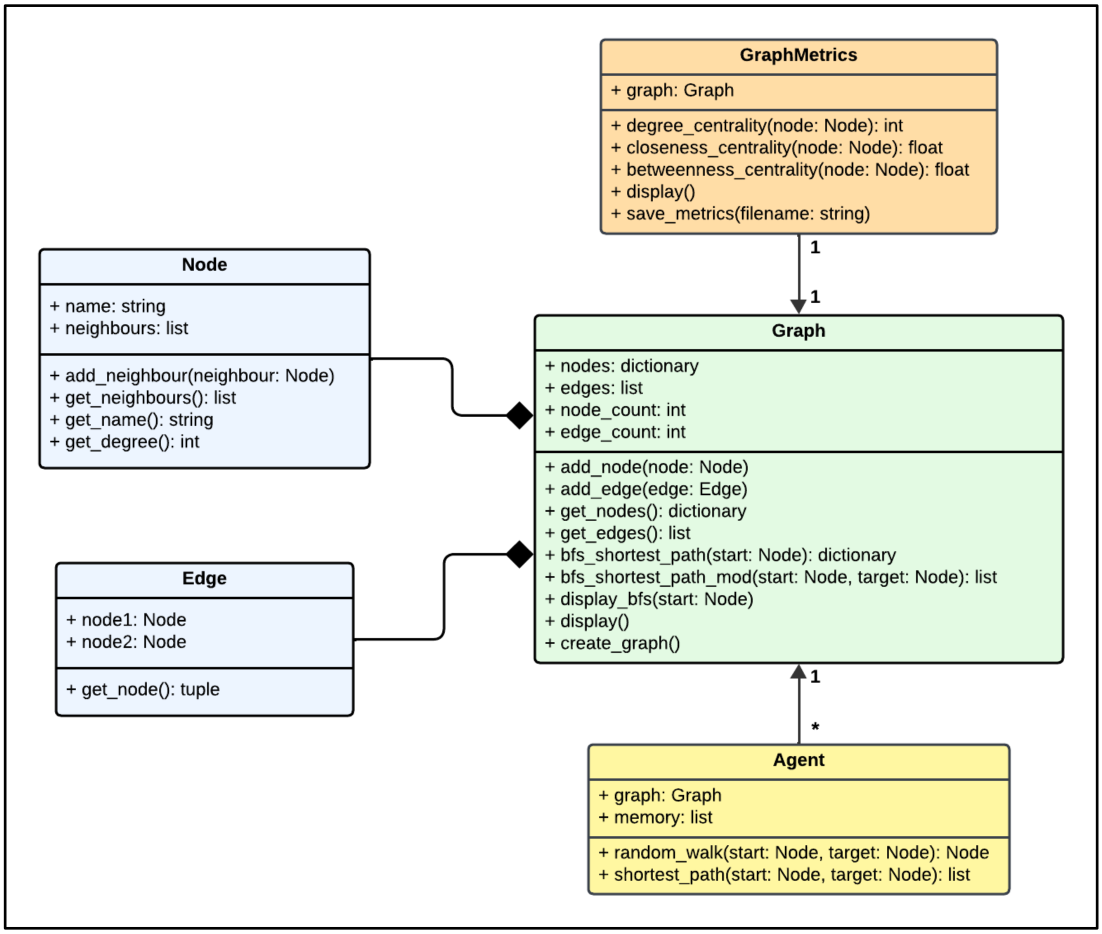


In order to design the undirected graph given Clements chapter 4.1, I chose to use a *dictionary* to store its nodes for efficient access to their names and a *list* for its edges to easily preserve the graph's connections. The graph class is composed of the node and edge classes as seen in the diagram above. The functions to compute the shortest paths and distances within the graph use the backbone of the breath-first search algorithm. These are later used to calculate the graph's metrics defined in Clements chapter 2.5.

World Metrics

In designing my GraphMetrics class, I ensured that my three centrality measures respected the equations defined in Clements chapter 2.5. For Betweenness Centrality specifically, I leveraged the BFS algorithm and slightly modified it to include a path recording that stores the current shortest path found – and the shortest path only – to a *list* of path(s) that exist from the start to the target node (see graph.py lines 78-96).

Agent Design



Evaluation

After saving the simulation results in a .csv file in the root directory of my python solution, here are the significant differences that I found from analysing the data from Random Walk (RW) and Shortest Path (SP) movement:

- Number of visited nodes: The RW function more often than not results in more nodes visited while navigating from the randomly selected start and target nodes. A simple manual percent difference computation results in the RW having approximately 120% longer paths than the SP approach. (The average RW visited approximately 12 nodes, while the SP visited about 3 nodes).
- Reason for this: randomness is inherently inefficient for an agent to move around a world. This approach has a high potential of revisiting nodes and taking overly complicated routes to the target node. On the other hand, the BFS nature of the SP method allows it compute the shortest path directly to the target node.
- Am I surprised? No. A lack of directionality – which usually means taking longer paths to a goal – is one of the fundamental properties (or problems) of the Random Walk in graph theory.

After carefully comparing the simulation results with the metrics results (which are automatically saved in two separate .csv files), I found some pleasing correlations with the theoretical centrality measures discussed in Clements chapter 2.5. As expected, simple analysis suggests that nodes with higher centrality metrics tend to be visited more often during RWs. As for the SP walks, we see quite the opposite. Indeed, we notice that the SP walks often omit visiting nodes with higher centrality metrics. This is a clear illustration of why “more connected” nodes in a graph usually act as bridges in longer paths.

References

Clements, M. (2019). Empowerment and Relevant Goal Information as Alternatives to Graph-Theoretic Centrality for Navigational Decision Making.