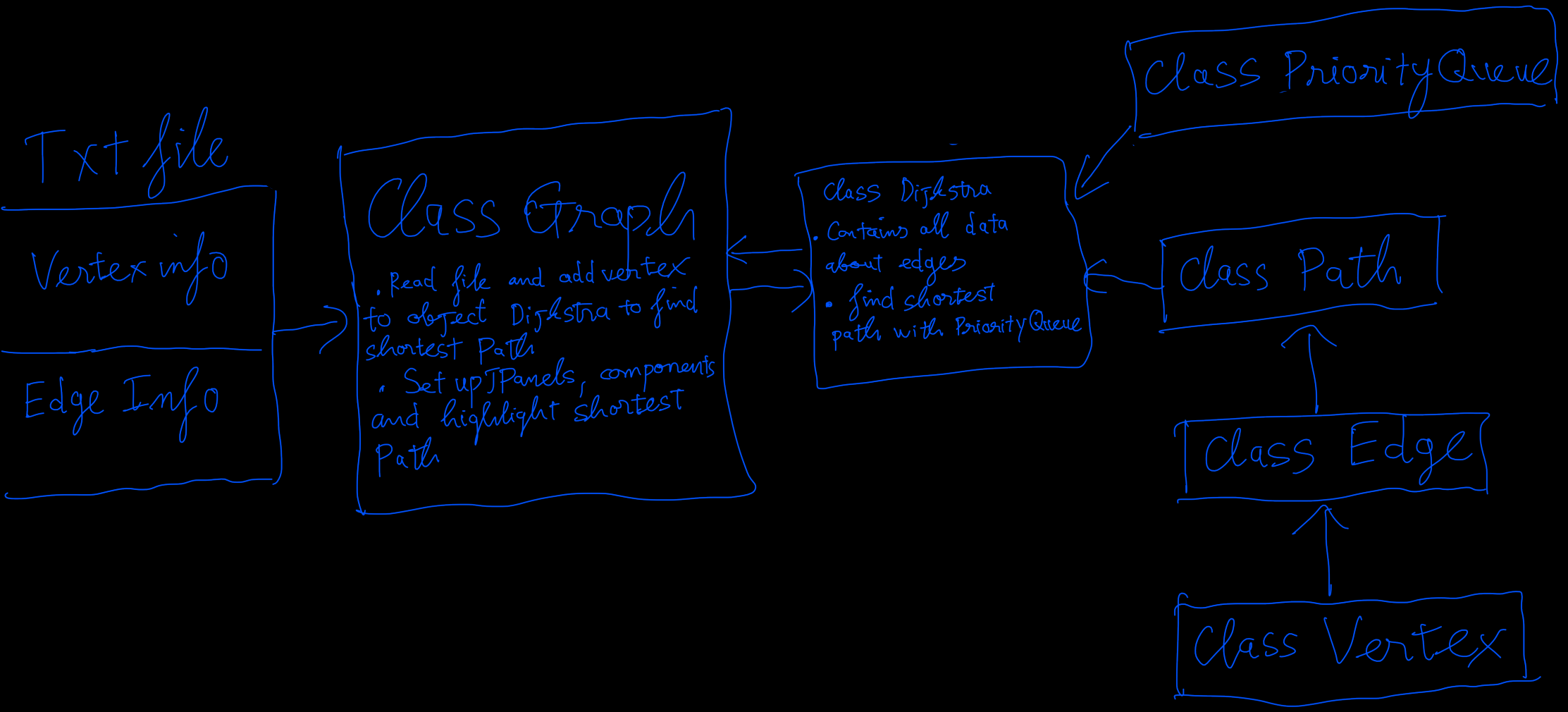
Txt file
Vertex info
Edge Info
Class Graph
• Read file and add vertex to object Dijkstra to find shortest Path
• Set up JPanels, components and highlight shortest Path
Class Dijkstra
• Contains all data about edges
• find shortest path with PriorityQueue
Class PriorityQueue
Class Path
Class Edge
Class Vertex

Class Vertex

* Properties:
  - String name
  - String address
  - int x, y
* Method:
  - Constructor (name, address, x, Y)
  - toString()

Class Edge

* Properties
  - String start
  - String end
  - int timeCost
  - int distCost
* Methods:
  - Constructor (String start, String end, int timeCost, distCost)
  - String getStart():
  - String getEnd():
  - int getTimeCost():
  - int getDistCost():

Class Path implements Comparable

* Properties:
  - LinkedList<Vertex> path
  - int totalCost:
  - String startPoint
  - String endPoint

  ex: Path: AC → C → T has startpoint A and end point T
* Method:
  - Constructor(): default
  - Getter for startPoint, endPoint, totalCost
  - void addPath(Edge newEdge)
    add newEdge to path (make sure endpoint of path is same as the startpoint of newEdge, update totalCost, update endPoint)
  - @override compare(): Compare total cost between Path.
  - String toString():

Class PriorityQueue<Path>

Compare based on totalCost of Path

Class Dijkstra

+ Properties:

- PriorityQueue<Path>
- int totalCost : total cost of the shortest path
- Map<String, Edge> edgeList : : this way, we can find the desired edge faster.
  - String start
  - String goal

(String: start point of the edge)

Methods:

- Constructor(String start, goal)
- addEdge(Edge edge): add new edges to edgeList
- static shortestPath()
  return Path shortestPath using Dijkstra

Class Graph extends JPanel (Reference: GameStarterTemplate.txt)

* Methods:

- Constructor (String fileName): set fileName. Set up Jpanel, JFrame
- String toString(): return all vertices and costs
- Methods that create the panels for user input start, end point and choose between use Time Cost or Dist Cost and save into this class properties.
- void readFile(String fileName) (Ref: on Canvas Announcement)
  - Read file and save vertex and edges info into class properties. And add those info to object Dijkstra
- void paintComponent(Graphics g)
  - Import graph image (FinalProject Graph_Basics_400x400.png)
  - Call Dijkstra.shortestPath() and highlight the path based on the result

* Properties:

- Map<String, Vertex> vertexList
  - String: point Name (ex: A, B, ...)
  - Vertex: object corresponding to name
- Map<String, Edge> edgeList: ; this way, we can find the desired edge faster.
  - String: start Point of the edge