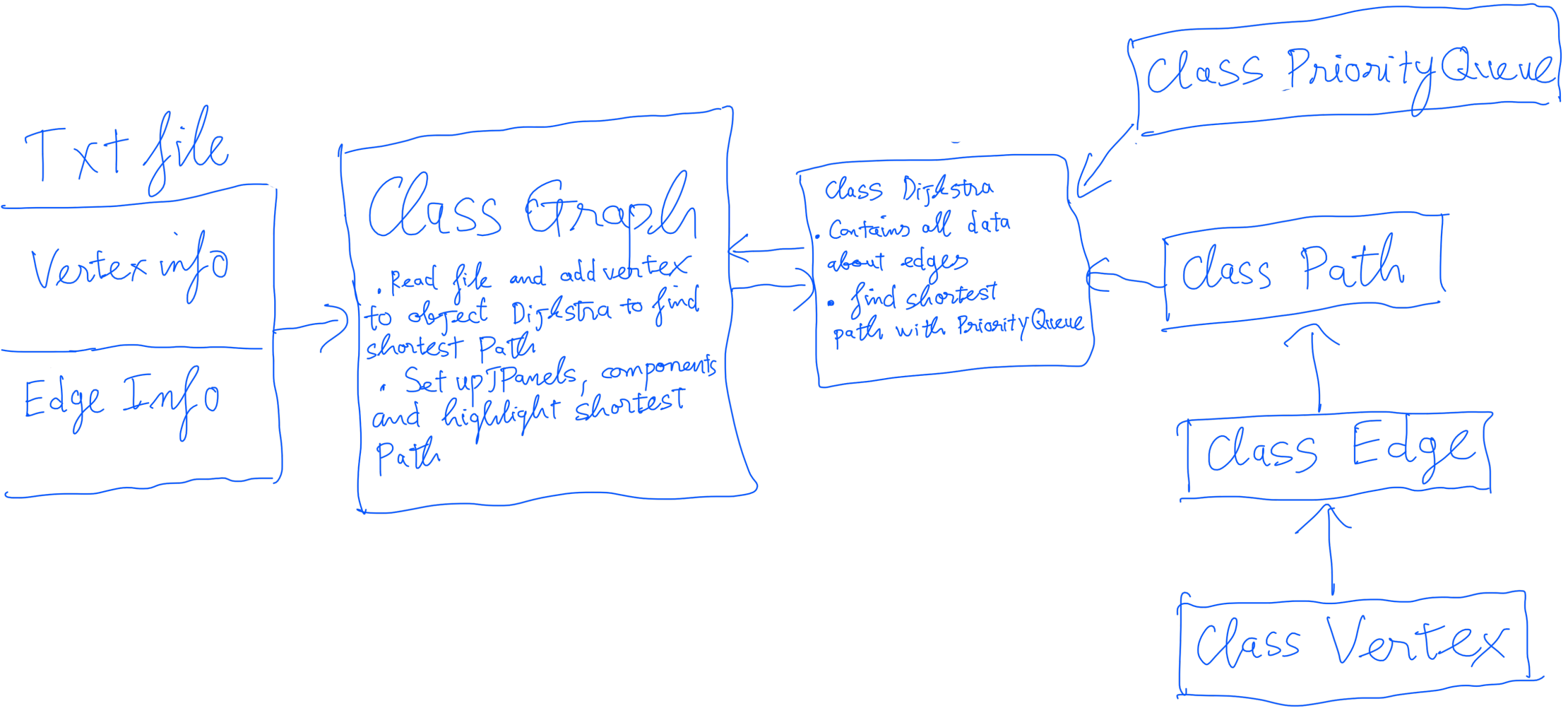
Txt file
Vertex info
Edge Info
Class Graph
• Read file and add vertex to object Dijkstra to find shortest Path
• Set up JPanels, components and highlight shortest Path
Class Dijkstra
• Contains all data about edges
• find shortest path with PriorityQueue
Class PriorityQueue
Class Path
Class Edge
Class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

class Vertex

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

class Edge

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

class Path implements Comparable

* Properties:
  - LinkedList<Vertex> path
  - int totalDistCost, totalTimeCost
  - Vertex startPoint
  - Vertex endPoint

  ex: Path: AC→C→T has startpoint A and end point T
  - int cost: will be either timeCost or distCost
* Method:
  - Constructor (boolean useDistCost)

    True: cost = distCost
    False: cost = timeCost
  - Getter for startPoint, endPoint, totalCost
  - void addPath(Edge newEdge)

    add newEdge to path (make sure endpoint of path is the same as the startpoint of newEdge, update endpoint, update cost, totalDistCost, totalTimeCost
  - @Override compare(): Compare total cost between Path.
  - String toString():

class PriorityQueue<Path>

Compare based on cost of the given 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 and end of the edge. Ex: "AC"
- Vertex start
- Vertex goal
- boolean useDistCost : True → use distCost; False → use timeCost

Methods:

- Constructor(Vertex start, Vertex goal)
- addEdge(Edge edge): add new edges to edgeList
- static shortestPath(bool useDistCost)
  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 TimeCost or DistCost 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 (FinalProjectGraph_Basics_400x400.png)
  - Call Dijkstra.shortestPath() and highlight the path based on the result

* Properties:

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