DAVID STEVEN MONTOYA PARRA - A00362450
ALEJANDRO FONSECA FORERO - A00362167

REPORT OF ENGINEERING METHOD

PROBLEM IDENTIFICATION

Problem Definition

Azur Lane, a famous company specialized in general ore mining has been having problems with its outdated operational software for years, but the original team that developed said software was dissolved recently. IT dept. director Hiryuu Smith has decided she had enough, and opened a contract with various software developer teams to evaluate different proposals and pick the one that suits their needs the most.

For the operations regarding ore mining, they need to keep track of their caves, especially what minerals are excavated there, and the tunnels a cave has with another cave. For the tunnels they need to know the travel difficulty they have (1 for an easy travel, 10 for a really difficult one) and the direction of traversing: A tunnel can be one way only or both ways. On the other hand, they also need to calculate the easiest route between a cave and another cave in the system; this to prevent unnecessary resource consumption and prevent accidents.

You're one of the software developer teams that were picked, and before creating the complete software, you were tasked in the creation of a demo that manipulates a small cave system. The software needs to fulfill the CRUD functions regarding caves and tunnels, as well as the easiest route calculating function.

Functional Requirements:

+Give a crude visualization of the cave.
+Show the difficulty of travelling from a cave to another.
+Calculate the easiest route between a given cave and any other.

COMPILATION OF INFORMATION

Definitions:

Vertex: A vertex is one of the two basic units out of which graphs are constructed. Vertices of graphs are responsible for saving the information about what the graph is representing.

Edge: An edge is (together with vertices) one of the two basic units out of which graphs are constructed. Each edge has two (or in hypergraphs, more) vertices to which it is attached, called its endpoints, acting as their union in the graph.

Graph: A graph data structure consists of a finite (and possibly mutable) set of *vertices* (also called *nodes* or *points*), together with a set of unordered pairs of these vertices for an undirected graph or a set of ordered pairs for a directed graph. These pairs are known as *edges* (also called *links* or *lines*), and for a directed graph are also known as *arrows*.

Dijkstra algorithm: Dijkstra's algorithm (or Dijkstra's Shortest Path First algorithm, SPF algorithm) is an algorithm for finding the shortest paths between nodes in a graph.

CRUD Operations:

The **CRUD** concept is closely linked to digital data management. **CRUD** refers to an acronym in which the first letters of the four fundamental operations of persistent applications in database systems meet:

-Create (Create records)
-Read bzw. Retrieve (Read Records)
-Update (Update records)
-Delete bzw. Destroy (Delete Records)

Simply put, **CRUD** summarizes the functions required by a user to create and manage data. Several data management processes are based on **CRUD**, in which these operations are specifically adapted to the requirements of the system and the user, either for database management or for the use of applications. For experts, operations are the typical and indispensable access tools to check, for example, database problems, while for users, CRUD means creating an account (create yourself) and use it (read), update it(update) or delete (delete) at any time.

POSSIBLE SOLUTIONS

Idea generation methodology used: Brainstorming.

Solution 1:

Use a graph to build the cave system and dijkstra algorithm to show the difficulty of travelling between caves and the easiest route from a given cave to another

Solution 2:

Use a graph to build the cave system and Floyd-Warshall algorithm to show the difficulty of travelling between caves and the easiest route from a given cave to another

Solution 3:

Use a n-ary tree to build the cave system and travel with BFS-like algorithms.

TRANSFORMING IDEA FORMULATION TO PRELIMINARY DESIGNS

Description solution 1: The graph will be directed and weighted as to create a more realistic situation in which caves are to and from. We may take advantage of Dijkstra's algorithm capability to return all shortests paths.

Description solution 2: The graph will be directed and weighted as to create a more realistic situation in which caves are to and from. We may take advantage of Floyd-Warshall's algorithm capability to return all shortests paths and filter the one we want, or just use a rewritten logic that only looks for the path to the desired vertex.

Description solution 3: Each node in the n-ary tree will have an attribute of the difficulty of reaching there that will be used to calculate the easiest path to "an exit" which is, one of the leaf nodes.

DISCARDED IDEAS

Solution 3 was discarded because of the unmanageability of having a tree structure represent such a problem.

EVALUATION AND SELECTION OF THE BEST SOLUTION

Solution	Knowledge about the topic	Value for client	Ease of development	Flexibility	Total	Approved
Solution 1	8	10	7	9	34	Yes
Solution 2	8	9	7	10	34	No

Knowledge about the topic: Our theoretical knowledge about the topic required to implement this solution in the program.

Value for the client: The value that the implementation of this solution will visually and performance-wise give to the client, such as: Faster recognition, ease of use and resource allocation.

Ease of development: How easy and/or fast it is to develop the solution in code.

Flexibility: How flexible the code is in regards to future-proofing and fixing any errors that may arise in its use.

CHOSEN SOLUTION

We have chosen solution 1 considering dijkstra is easier to implement properly and since there will not be negative weights between caves (as it would not make sense) the algorithm should work properly without any further fixes, other important point to take into account is that dijkstra is faster than Floyd-Warshall in the context of this problem.

"In the end... We only regret the chances we didn't take"
- Lewis Carroll