

# Documentation

Mateo Rubio - A00400104

Alejandro Quiñones - A00377013

## Engineering Method

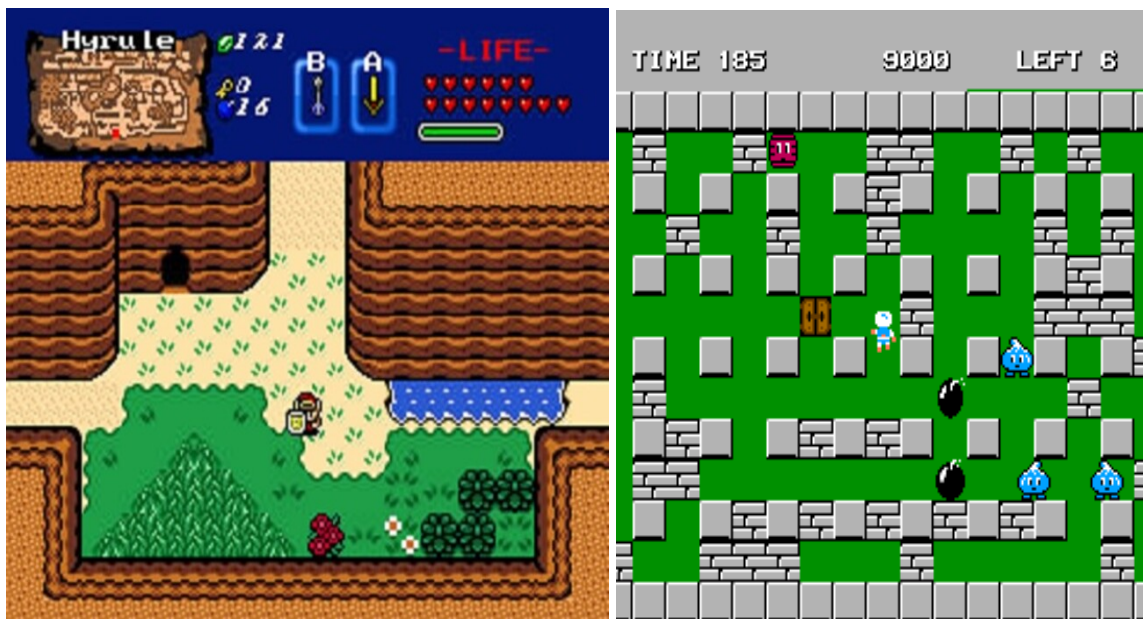
### Phase 1

<b>Client</b>	CyED professors from ICESI university
<b>User</b>	Any person that wants to play and enjoy Argentina's presidential campaign of 2023
<b>Problem Context (Identification)</b>	You are Javier Milei and the presidential elections are coming, and you forgot all about the geographical distribution of your country due to an accident so you are worried about how you are going to surpass your rivals in the campaign
<b>Functional Requirements</b>	<p><b>FR1:</b> The system need to add the most representative cities of Argentina in order to encompass an effective way to do the electoral campaign with the given resources</p> <p><b>FR2:</b> The system needs to generate random costs to travel and do your campaing in the city and through the country due to the unprecent situations that take place in the roads of Argentina.</p> <p><b>FR3:</b> The system allows the Milei to travel to a city based on which one he has been in order to campaign or check the state of the city.</p> <p><b>FR4:</b> The system shows graphically the cheapest cost to travel across argentina (from the bottom to top of the country) in order to Milei take into account how to take the maximum advantage of his budget.</p> <p><b>FR5:</b> The system needs to graphically show millei the cheapest way to travel to all the cities even though it may not end at the top of the Country but this path is the most convenient to make his proposals known in all the country.</p>

	<b>FR6:</b> The system needs to let know Milei know that he ended his campaign when he meets with Sergio Massa at the northernmost city
<b>No Functional Requirements</b>	<b>NFR1:</b> Use graphs <b>NFR2:</b> Implements algorithms as BFS, DFS, Dijkstra, Floyd-Warshall, Prim and Kruskal to solve 3 main problems, search, minimal travel cost and minimal spanning tree of all nodes.

### Phase 2:

Precedents: We have looked for some video games in which the use of a graph can be included and we have found games like bomberman, zelda (classic) in which the navigation between areas of the map is done through doors or open areas on the map for the user to enter and move.



We compare what we think with what is done in this type of games and we notice that it is possible to apply it to the problem we have, although we still have some doubts to solve, such as:

- + Distribution of the maps based on the network.
- + Use of the MST for the reduction of routes.

That is why we ended discarding these ideas and go for something a bit more applicable that is traveling with costs and how having a limited budget affects the decisions we take in order to reach an objective. Very similar to a long list of issues caused by decision making.

### Phase 3:

#### a) Brainstorm

- + Maybe use FXGL which is a JavaFX Game Development Framework, with which we can extend the JavaFX functionalities and develop the game in a more efficient way.
- + We have also taken a look at libGDX, which is a java based framework, but not in JavaFX, so we should check if we can use it.
- + We have thought about using BFS to create a hint to let the player know which path to take.
- + Use the MST to generate an easier route for the player, this can be done if the player finds a special item in the game.
- + A wall system in which certain nodes will be of wall type and the player will be able to unlock these walls if he pays to pass through, which implies an in-game coin system.
- + We also thought about which algorithms we will use for solving the problems so we evaluated all main things such as for a shortest path if we needed the whole path is better to use dijkstra but only for the shortest or the cheapest values is easier to use floyd warshall, also by touring the graph if the dfs time is relevant or if we really need it or better use the distance in bfs, does type of things.

#### b) Review list.

- + How to assign a node to doors on the map in specific areas.
- + Which framework.
- + Implementation of the coin/budget system.
- + Investigate if it is possible to generate a map randomly and make assignments after its generation.

- + which method select for solving problems

#### Phase 4 (View Phase 5 to understand de points)

A) Selecting the theme of our game was very complicated as we differed in what we thought but at the ended we focused on Milei assignment to the the balance between our criteria and time, because it is simple to understand and pleasant to code [ UND (5 pts), EFF (4 pts), ADA (6 pts), FUN (12 pts) & ENJ (4 pts)].

B) We ended up using djijkstra for getting shortest paths and kruskal for the minimal spanning three due to how well they align to our focus such as giving the cities path and not just a number or raw data, but more an ordered set of information [UND (8 pts), EFF (4 pts), ADA (9 pts), FUN (13 pts) & ENJ (5)]

C) For the graphical design we used javafx without any framework in order to have control over all our code and also know what we are doing even tho it may sacrifice some visuals or mechanics that freameworks bring [UND (9 pts), EFF (3 pts), ADA (6 pts), FUN (14 pts) & ENJ (4)]

#### Phase 5

The criteria taking into account for the decisions made before were:

- Understanding, how complex it is to comprehend methods/implementations (1 to 10)
- Efficiency, in terms of time and resources how much it consumes (1 to 5)
- Adaptability, Given the possibility to add new things to the system how easy they will fit in our code (1 to 10)
- Functionality, related to the main 3 problems how well does it accomplish or approaches to what is asked for and solving them (1 to 15)
- Enjoyabiliy, it is fun to play or enjoy the game (1to 5)