

Problem identification and requirements analysis

Case Study:

Customer	Icesi Games
User:	Players interacting with the graphical fishing game system.
Functional requirements	<p>RF1 Create Game Map</p> <p>RF2: Create Two Game Modes</p> <p>RF3: Hook Movement</p> <p>RF4: Movement of the Fish</p> <p>RF5: Completion of the Game.</p> <p>RF6: Graphical Interface</p>
Contexto f the problema	<p>Icesi Games wants to develop a fishing game that allows users to experience two main modes: water fishing and lava fishing. The game must be graph-based, implementing algorithms for route optimization and navigation between nodes. This system must offer an intuitive and functional graphical interface, and be capable of handling at least 50 vertices and 50 edges in the map graph.</p> <p>The goal is to create a playable environment that takes advantage of BFS (Breadth-First Search) and Floyd-Warshall algorithms to solve route and shortest path problems on interconnected maps.</p>
Non-Functional Requeriments	<p>RNF1: The system must be implemented using Java and JavaFX.</p> <p>RNF2: The interface must be intuitive and easy to use.</p> <p>RNF3: The design must be modular, allowing the integration of new algorithms or modes.</p> <p>RNF4: The system must ensure smooth performance with at least 50 vertices and 50 edges.</p>

Identifier and name	RF1 Create Game Map		
Summary	The system must allow the user to configure a map of the game, selecting locations (nodes) and the connections between them (navigable routes).		
Inputs	Input name	Date type	Condition valid values
	Location	String	Unique values. It cannot be empty.
	Connection	String	Valid names of connected nodes.
Result or Postcondition	The visual map shows the locations and their connections.		
Outputs	Output name	Data type	Format
	Visual Representation	Image/Graphic	Map with nodes and connected routes.

Identifier and name	RF2 Create Two Game Modes		
Summary	The system must allow the user to select between two game modes: Water Fishing and Lava Fishing , which affects the visual characteristics of the map and the rules of the game (such as movement costs).		
Inputs	Input name	Date type	Condition valid values
	Game Mode	Enum	WATER or LAVA. It cannot be empty.
Result or Postcondition	The rules and visual environment are updated according to the selected mode.		
Outputs	Output name	Data type	Format
	Visual Change	Image/Graphic	Environment adapted to the selected mode.

Identifier and name	RF3 Hook Movement		
Summary	The system should allow the user to move the hook between the nodes on the map, consuming energy with each movement.		
Inputs	Input name	Date type	Condition valid values

	Motion Direction	String	"Up", "Down", "Left", "Right".
Result or Postcondition	The hook moves to the indicated node, and the player's energy decreases.		
Outputs	Output name	Data type	Format
	Visual Change	Image/Graphic	Updated hook position.

Identifier and name	RF4 Movement of the Fish		
Summary	The system should automatically move the fish between nodes on the map, prioritizing routes that take it away from the hook		
Inputs	Input name	Data type	Condition valid values
	Current Position of the Fish	Node	Current node of the fish on the map.
Result or Postcondition	The fish moves to the node that maximizes the distance to the hook.		
Outputs	Output name	Data type	Format
	Visual Change	Image/Graphic	Updated fish position.

Identifier and name	RF5 Completion of the Game		
Summary	The system must determine the end of the game when any of the following conditions are met:		
	<p>The player runs out of energy. (Lava game mode only)</p> <p>The hook reaches a node close enough to the fish.</p>		
Inputs	Input name	Data type	Condition valid values
	Player Energy	int	Must be greater than or equal to 0 and only available if the game mode is lava
	Distance Between Nodes	int	Distance less than or equal to the permitted threshold.
Result or Postcondition	<input type="checkbox"/> If the power reaches zero, the system displays "Game Over" . <input type="checkbox"/> If the hook hits the fish, the system displays "You Win" .		
Outputs	Output name	Data type	Format

	Endgame Message	String	"Game Over" o "You Win".
	Final Visual Status	Imagen/Gráfico	Map at the end of the game.

Identifier and name	RF6 Graphical Interface		
Summary	The system should provide a graphical interface that allows the user to observe the current state of the game (hook position, fish, remaining energy).		
Inputs	Input name	Date type	Condition valid values
	Selected Action	String	"Move Hook", "Select Mode".
Result or Postcondition	The visual state of the interface is updated according to the actions taken.		
Outputs	Output name	Data type	Format
	Updated Status	Image/Graphic	Positions and energy reflected visually.