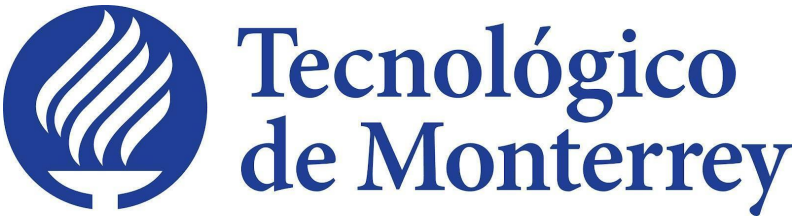


Construcción de Software y toma de decisiones

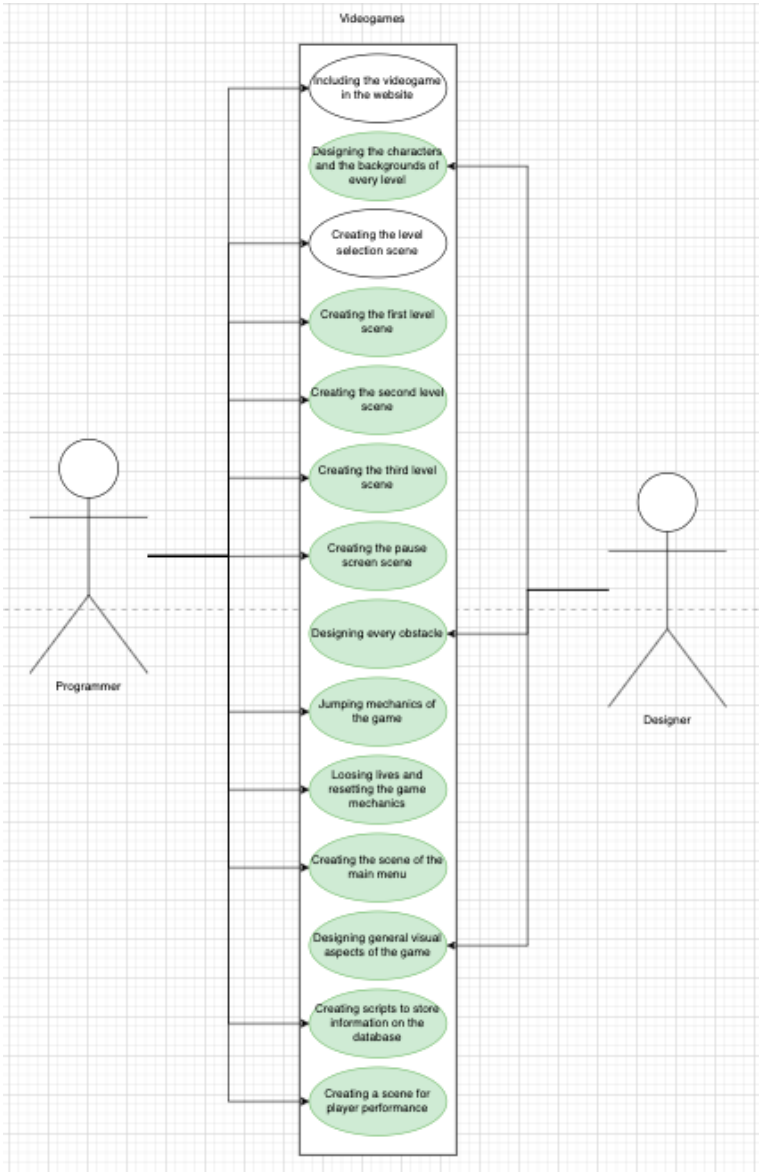
TC2005B.400

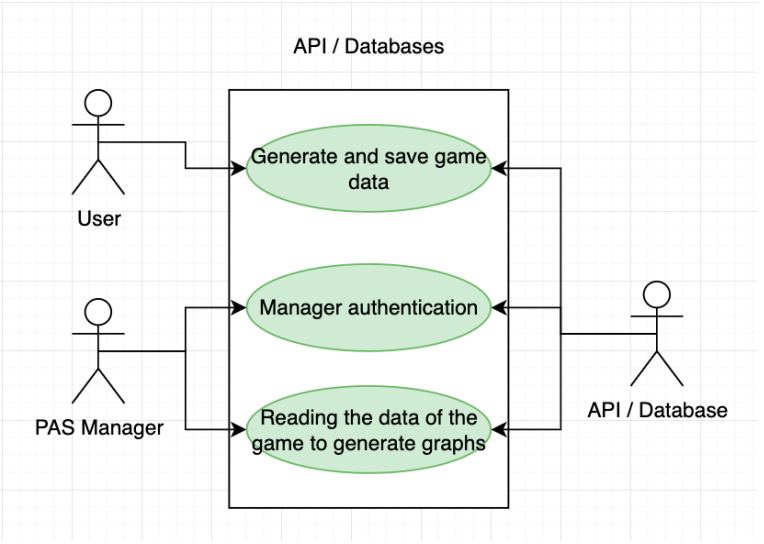
Ejercicio de actualización de una base de datos relacional en MySQL y diagramas UML

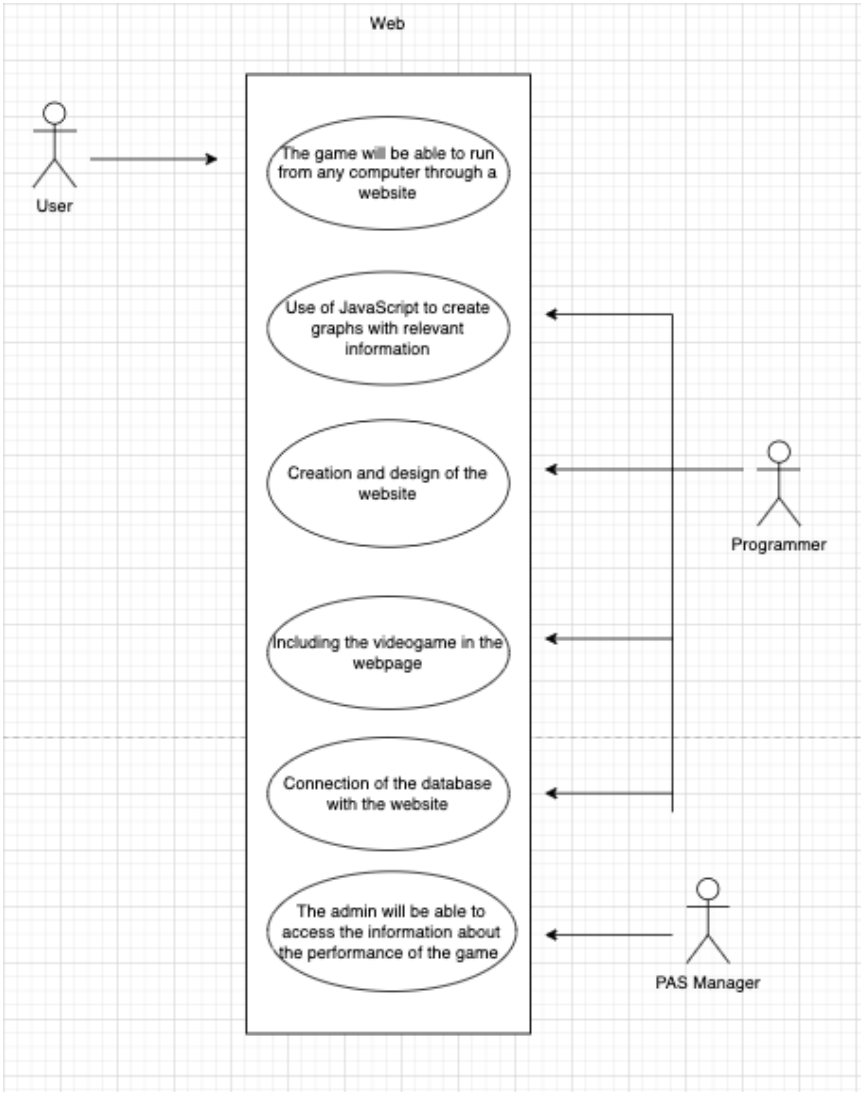
3 de mayo 2022

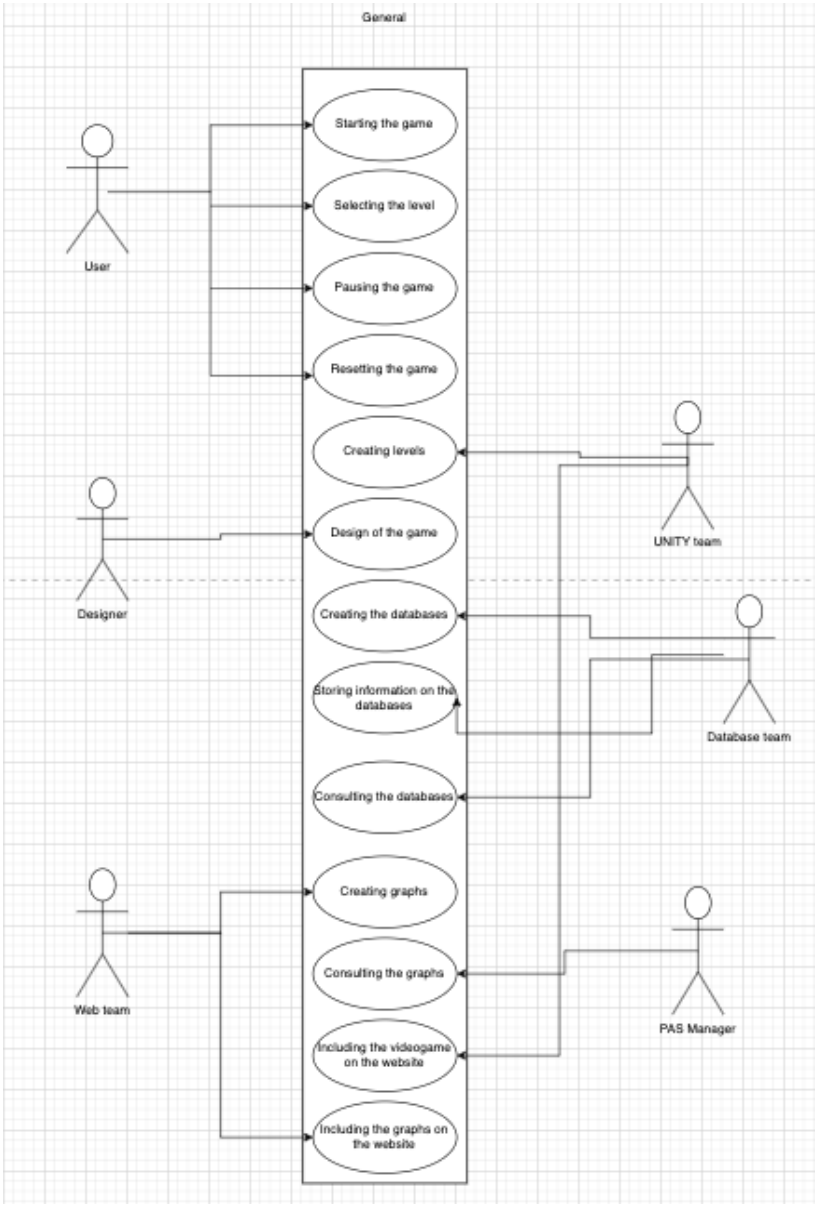
Diego Araque | A01026037
Alvaro Garcia | A01781511
Salomon Dabbah | A01028445
Marco Torres | A01025334

Use case Diagrams and tables






Web
User
The game will be able to run from any computer through a website
Use of JavaScript to create graphs with relevant information
Creation and design of the website
Including the videogame in the webpage
Connection of the database with the website
The admin will be able to access the information about the performance of the game
Programmer
PAS Manager



Use case name	Creating the scene of the first level
Related Requirements	Creating the scene of the second level Creating the scene of the third level Applying the jumping mechanics of the game Losing lives and resetting the game mechanics Designing the visual elements of the game Designing every obstacle
Goal in Context	Creating the first level of the game with proper design and mechanics
Preconditions	Have the visual elements and the design of

	the obstacles ready
Successful End Condition	A fun level that works
Failed End Condition	A level with bad design and functionality
Primary Actors	Developers
Secondary Actors	Designers
Trigger	The player chooses the first level of the game

Main Flow	
Step	Action
1	Fixating the background and the game camera
2	Adding the sprites (characters, floor and obstacles)
3	Assign to each sprite a rigidbody and designating the capsule colliders
4	Creating the following scripts: - Character controller - Player movement
5	Adding music to the game
6	Testing

Use case name	Creating the scene of the second level
Related Requirements	Creating the scene of the first level Creating the scene of the third level Applying the jumping mechanics of the game Losing lives and resetting the game mechanics Designing the visual elements of the game Designing every obstacle
Goal in Context	Creating the second level of the game with proper design and mechanics

Preconditions	Have the visual elements and the design of the obstacles ready. Completing the scene and the scripts of the first level.
Successful End Condition	A fun level that works
Failed End Condition	A level with bad design and functionality
Primary Actors	Developers
Secondary Actors	Designer
Trigger	The player chooses the second level of the game

Main Flow	
Step	Action
1	Fixating the background and the game camera
2	Adding the sprites (characters, floor and obstacles)
3	Assign to each sprite a rigidbody and designating the capsule colliders
4	Creating the following scripts: - Character controller - Player movement
5	Adding music to the game
6	Testing

Use case name	Creating the scene of the first level
Related Requirements	Creating the scene of the first level Creating the scene of the second level Applying the jumping mechanics of the game Losing lives and resetting the game mechanics Designing the visual elements of the game Designing every obstacle

Goal in Context	Creating the third level of the game with proper design and mechanics
Preconditions	Having the visual elements and the design of the obstacles ready. Completing the scene and the scripts of the first levels.
Successful End Condition	A fun level that works
Failed End Condition	A level with bad design and functionality
Primary Actors	Developers
Secondary Actors	Designers
Trigger	The player chooses the third level of the game

Main Flow	
Step	Action
1	Fixating the background and the game camera
2	Adding the sprites (characters, floor and obstacles)
3	Assign to each sprite a rigidbody and designating the capsule colliders
4	Creating the following scripts: - Character controller - Player movement
5	Adding music to the game
6	Testing

Use case name	Designing each obstacle
Related Requirements	Designing the general visual aspects of the game
Goal in Context	Creating the obstacles and the note blocks so the player will lose lives each time he collides with an obstacle

Preconditions	Having the background and the ground of each level ready to add them to the game.
Successful End Condition	A visually pleasant scene
Failed End Condition	A laggy and visually unpleasant scene
Primary Actors	Designers
Secondary Actors	Developers
Trigger	When a player collides with an obstacle he loses a life

Main Flow	
Step	Action
1	Designing the blocks each with its respective note and size
2	Assigning a rigidbody and a capsule collider to each sprite
3	Creating the script so the player loses a life each time he collides with an obstacle
4	Testing

Use case name	Losing lives and resetting the game mechanics
Related Requirements	Applying the jumping mechanics in the game Creating scripts and variables to store relevant game info on the database Designing each obstacle Designing the scene of the first level Designing the scene of the second level Designing the scene of the third level
Goal in Context	Provide the expected functionality to each level of the game
Preconditions	Having the obstacles and the jumping mechanics of the game ready

Successful End Condition	The player has 5 lives and loses one each time he collides with an obstacle, if he/she loses all lives the game will be reset
Failed End Condition	The player does not lose lives when he collides with an obstacle and/or the game does not end/reset when the player loses all lives
Primary Actors	Developers
Secondary Actors	No secondary actors
Trigger	When the player collides with an obstacle

Main Flow	
Step	Action
1	Creating the losing lives script
2	Creating the reset script
3	Integration of the scripts on the level
4	Testing

Use case name	Creating the scene of the main menu
Related Requirements	Using unity for the development of rhe game Designing general visual aspects for the game Creating the scene of level selection
Goal in Context	Showing the user that is already on the game and provide the option to go back to the level select screen
Preconditions	Install Unity and TextMeshPro, having some assets ready for the menu
Successful End Condition	The menu looks aesthetic and has a redirect button that moves you to the level select

	scene
Failed End Condition	The “Play” button of the menu is not visually pleasant
Primary Actors	Developers
Secondary Actors	Designers
Trigger	When the player clicks the “Play” button

Main Flow	
Step	Action
1	Creating a new scene
2	Styling of the scene
3	Programming and integration of the change level scene
4	Testing

Use case name	Creating the pause scene
Related Requirements	Using unity for the creation of the videogame Designing general visual aspects of the game Designing the scene of the first level Designing the scene of the second level Designing the scene of the third level
Goal in Context	Being able to pause and resume at any given time
Preconditions	Having the scenes of all three levels ready
Successful End Condition	The player selects the pause button and this redirects him to a different scene so he can resume the game or go back to the main menu
Failed End Condition	The pause button does not redirect the player to the pause menu

	The game does not pause properly The game does not resume properly
Primary Actors	Developers
Secondary Actors	Designers
Trigger	When the player selects the pause or resume button

Main Flow	
Step	Action
1	Adding the pause button to each level and creating the script for each button of the pause menu
2	Creating and designing the pause scene
3	Integrating the script to the game
4	Testing

Use case name	Jumping mechanics
Related Requirements	Using Unity for the creation of the videogame Designing the characters Losing lives mechanics Designing each obstacle Designing the scene of the first level Designing the scene of the second level Designing the scene of the third level
Goal in Context	Accomplishing the general objective of the game: making the user jump on each note
Preconditions	Having the scenes of all the levels, the characters and the obstacles
Successful End Condition	The player presses the spacebar key and the character jumps, the longer the player presses the key the longer the jump becomes
Failed End Condition	The spacebar key does not work or the

	prolonged jump is just a regular jump
Primary Actors	Developers
Secondary Actors	Designers
Trigger	When the player presses the spacebar key the character jumps

Main Flow	
Step	Action
1	Creating the script for the jumping mechanics
2	Variating the jumping speed to a desirable one
3	Integrating the script to the game
4	Testing

Use case name	Creation of the scene that appears at the completion of a level, which has information about the player's performance.
Related Requirements	Creation of the first level scene Creation of the second level scene Creation of the third level scene Applying jumping mechanics in the game Mechanics of losing lives and restarting the game Design of general visual elements for the game Design of each obstacle
Goal in Context	Score display at the end of the level
Preconditions	Some level that is already functional to check if the correct result is generated.
Successful End Condition	Score displays correctly
Failed End Condition	It does not direct you to a new scene with the score.

Primary Actors	Developers
Secondary Actors	Designers
Trigger	Player completes the level or loses

Main Flow	
Step	Action
1	Create the scene where all the necessary information is located.
2	Create a script that when you finish the game or lose all lives will direct you to this scene.
3	Integrating the script with the game
4	Testing

Use case name	Designing the general visual aspects of the game
Related Requirements	Designing each obstacle Designing the scene of the first level Designing the scene of the second level Designing the scene of the third level
Goal in Context	Creating a proper design for the elements of the game and the game itself
Preconditions	Knowing what the functionalities of the game are Being aware of the different elements that the game should have
Successful End Condition	Proper visual design of the game
Failed End Condition	Unbalanced visual design of the game
Primary Actors	Designers
Secondary Actors	Developers
Trigger	None

Main Flow	
Step	Action
1	Defining each element of the game
2	Designing each element
3	Testing all the elements together

Use case name	Character design and backgrounds for each level
Related Requirements	Design of each obstacle Design of general visual elements for the game. Design of the first level scene Design of the second level scene Third level scene design
Goal in Context	Create a proper design for the game characters and backgrounds.
Preconditions	To know the different levels that the game must have.
Successful End Condition	Proper visual design of the characters and backgrounds of the game.
Failed End Condition	Clogged and unbalanced game design with unattractive characters and backgrounds.
Primary Actors	Designers
Secondary Actors	Developers
Trigger	None

Main Flow	
Step	Action
1	Define which levels will be in the game.
2	Make the design or search for characters and backgrounds.
3	Test how they will look.

Use case name	Creating the database
Related Requirements	Creating the entity-relationship scheme Normalizing the entity-relationship scheme
Goal in Context	Store important information from the game
Preconditions	Have a normalized diagram
Successful End Condition	Database that stores everything important for the game and the managers
Failed End Condition	Relevant information isn't presented
Primary Actors	Developers
Secondary Actors	MySQL
Trigger	Commands in MySQL workbench

Main Flow	
Step	Action
1	Have an idea of what we want to store
2	Create entity-relationship diagram
3	Diagram is in the third normal form
4	With MySQL commands create what has been established

Use case name	Create entity-relationship diagram
Related Requirements	Normalize the entity-relationship diagram Create database
Goal in Context	Have the complete diagram with all the data we need
Preconditions	Have a general idea of the analisis that can be done
Successful End Condition	Have the diagram in UML 2.0
Failed End Condition	The diagram is not ready

Primary Actors	Developers
Secondary Actors	Modeling tools
Trigger	Document completed for the diagram

Main Flow	
Step	Action
1	Have an idea of what needs to be stored and analyzed
2	Use a modeling app
3	Create the diagram
4	All the developers need to check it

Use case name	Normalize the entity-relationship diagram
Related Requirements	Create the entity relationship diagram Create database
Goal in Context	Eliminate redundancy and bas practices
Preconditions	Diagram already exists
Successful End Condition	Diagram without redundancy and bad practices
Failed End Condition	Diagram with redundancy or bad practices or both
Primary Actors	Developers
Secondary Actors	Modeling tools
Trigger	Detect redundancy, add a primary key to each table, transitive and functional dependencies, etc.

Main Flow	
Step	Action
1	Entity-relationship already finished

2	Search for bad practices
3	Change this bad practices
4	Double-check

Use case name	Generate and store game data
Related Requirements	Create database in MySQL Connect database to the API Crate variables and scripts to store and send the data
Goal in Context	Users can generate data from each gamerun and it can be stored in the database
Preconditions	Connection between the API and the database
Successful End Condition	Store the game runs and update the level table
Failed End Condition	Data isn't sent or stored
Primary Actors	Developers
Secondary Actors	
Trigger	When a player passes a level.

Main Flow	
Step	Action
1	Define the in-game events that will generate data
2	Define possible values that each event can generate
3	Program the storage of the data locally
4	Code for the data to be send from the unity script to the API

Use case name	Manager Authentication
Related Requirements	Create database in MySQL Connect database to the API
Goal in Context	Make it possible for the managers to access the graphs and tables
Preconditions	Have a database connection in the API
Successful End Condition	Managers can create their account an access the graphs and tables
Failed End Condition	Authentication or create account doesn't work
Primary Actors	Developers
Secondary Actors	
Trigger	When a manager wants access to the graphs and tables

Main Flow	
Step	Action
1	Option in the login so the administrator can choose that he is an admin
2	Design the UI for the create account
3	Make tests

Use case name	Read the game data to generate the graphs
Related Requirements	Create the connection to the database in MySQL Connect the database to the API Create the variables and scripts to store relevant information from the game in the database
Goal in Context	Send data stored in the database to the client to generate the graphs

Preconditions	Connect the database to the API
Successful End Condition	Send the data correctly
Failed End Condition	Data is not sent to the database
Primary Actors	Developers
Secondary Actors	
Trigger	When graphs are accessed from the navbar

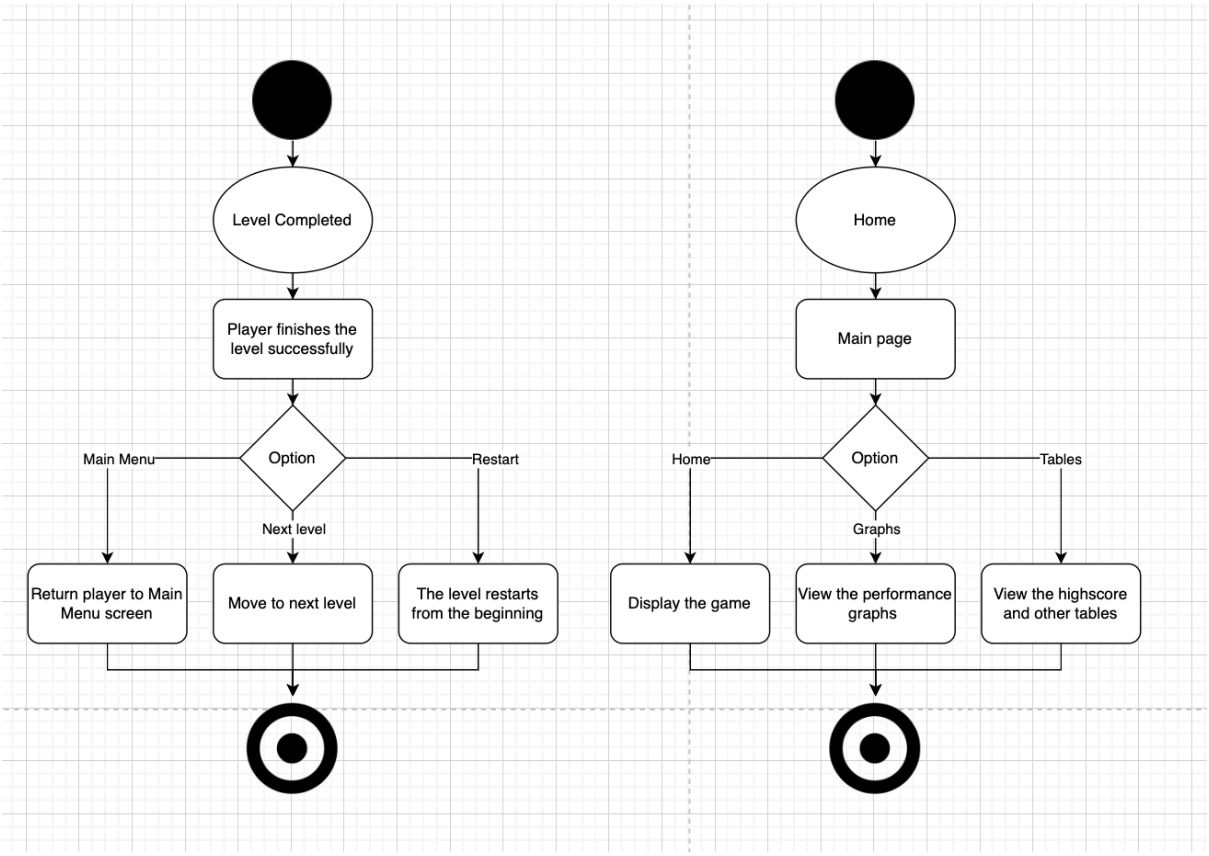
Main Flow	
Step	Action
1	Creating an endpoint in the database to receive data depending on the given parameters
2	Creating a function in the client that collects data and executes it when it is prompted
3	Testing

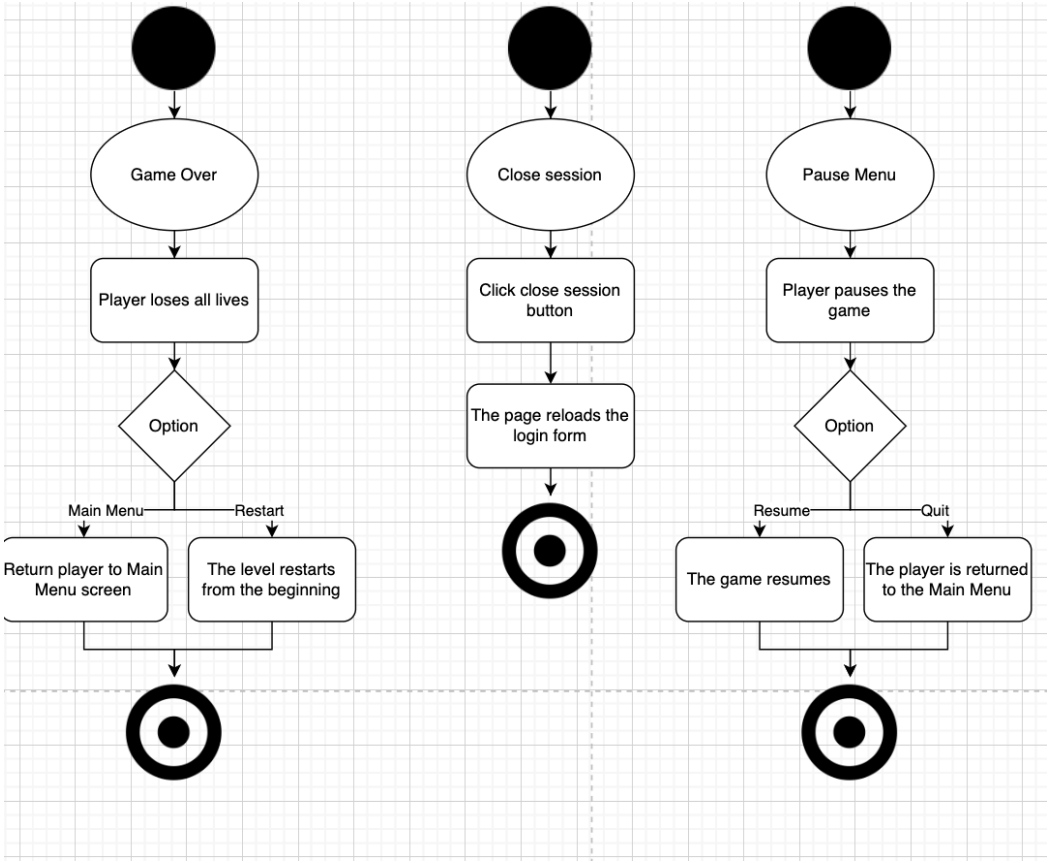
Use case name	The administrator can access the information about the game performance
Related Requirements	Connect the database in MySQL Connect the database to the API Connect the webpage with the database Creation and design of the webpage
Goal in Context	Administrators can access relevant information about the users performance in-game.
Preconditions	Finished game that sends information about the player to the database and linked to the webpage.
Successful End Condition	Administrators can visualize information about the game
Failed End Condition	Administrators can't visualize information about the game

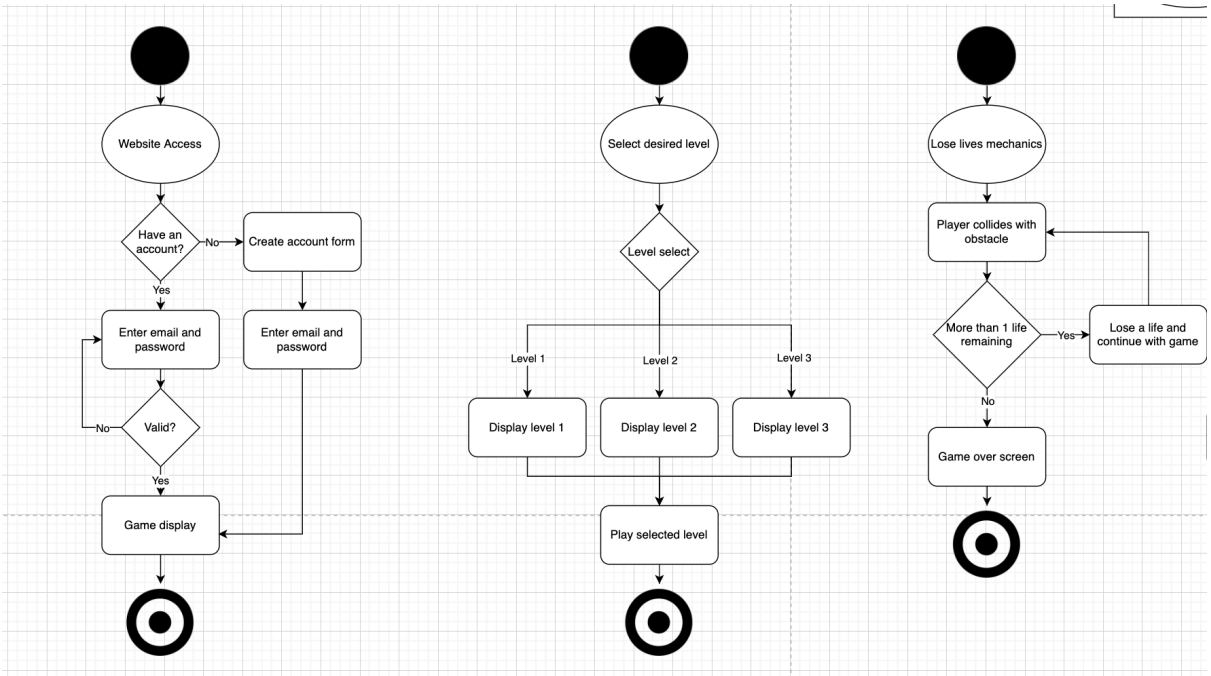
Primary Actors	Developers
Secondary Actors	-
Trigger	Administrator logs in and selects the graphs or tables option in the navbar

Main Flow	
Step	Action
1	Create authentication system
2	Register an account for the administrator from PAS, so that the page will display the graphs in the navbar, after the log in.
3	Make tests

Activity diagrams




Game Over
Player loses all lives
Option
Main Menu
Restart
Return player to Main Menu screen
The level restarts from the beginning
Close session
Click close session button
The page reloads the login form
Pause Menu
Player pauses the game
Option
Resume
Quit
The game resumes
The player is returned to the Main Menu


Website Access
Have an account?
No
Create account form
Yes
Enter email and password
Enter email and password
No
Valid?
Yes
Game display
Select desired level
Level select
Level 1
Level 2
Level 3
Display level 1
Display level 2
Display level 3
Play selected level
Lose lives mechanics
Player collides with obstacle
More than 1 life remaining
Yes
Lose a life and continue with game
No
Game over screen