TECNOLÓGICO DE MONTERREY

TC2005B Construcción de Software

Shhhootout!

Software Requirements

Stephan Guingor Falcón - A01029421

Francisco D. Salcedo - A01633010

Diego Mellado - A01655451

05/24/2022

Index

User Analysis

I. User Personas

Basic Information	Details
Name: Felipe Age: 15	He is a person who goes to high school and plays video games casually.

Basic Information	Details
Name: Sandra Age: 25	Sandra is a person who doesn't play many videogames but recently she has become curious.

II. User Stories

User story #1: Play the video game As a user I want to play a video game embedded in a web page	
Validation criteria: ● Play the video game on any web browser ● Play the video game only on desktop devices: PC and laptops.	Value: 10 Priority: 1 Estimate: 12h

User story #2: Design levels As a user I want to be able to design and save my levels to be played by me or other users.	
Validation: ● Design levels. ● Save them in a database. ● Show a table with the levels available. ● Play the level selected.	Value: 10 Priority: 1 Estimate: 10h

User Story #3: Quality filters As a user I want to be able to find and play the best levels within the platform	
Validation: ● Filter and show the levels by popularity	Value: 8 Priority: 2 Estimate:

User Story #4: Search by username

As a user I want to be able to find levels created by a username	
Validation: ● Search levels by username ● Show levels created by the given username	Value: 8 Priority: 3 Estimate: 8h

User Story #5: Measurable progression As a user, I want to see my stats for the levels I've played so I can monitor their progress over time.	
Validation: ● Validate the logged user ● Show stats about their performance such as time and score.	Value: 8 Priority: 3 Estimate: 7h

User story #6: Challenging gameplay As a user, I want user I want to design a level with an amount of difficulty to create a challenging gameplay	
Validation: ● Change the patrol path of the enemies	Value: 7 Priority: 4 Estimate: 9h

User Story #7: Editor elements As a user I want to be able to design my level using pre-designed elements: rooms, enemies, stage elements and interactive elements.	
Validation: ● Select predefined rooms ● Drag and drop stage elements ● Drag and drop enemies	Value: 1 Priority:10 Estimate: 9h

User Story #8: Reward System As a user I want to be punished if I don't follow the mechanics of the game.	
Validation: ● Points if the player eliminates an enemy. ● Additional points if the player completes the level without being detected by the enemies.	Value: 2 Priority: 9 Estimate: 16 h

User story #9: Edit created levels As a user I want to see the levels I have designed and re-edit them.	
Validation: ● See the designed levels ● Select them and re-design them and save them ● Play the edited level with the new features added	Value: 4 Priority: 4 Estimate: 10h

User story #10: As a user I want to be able to move around the map I designed and interact with the elements of the scenario.	
Validation: ● Players moves ● Player shot ● Player can pick-up guns	Value: 7 Priority: 6 Estimate: 9h

Relevant Software Requirements

I. Functional Requirements

Website

WF1. - The system will allow the video game to be played on the website.
WF2. - The system will display the user statistics on the website.
WF3. - The user is able to look for other player statistics, if registered.
WF4. - The user will be able to interact with the website.
WF5. - The user will be able to register on the website to view your statistics and be able to create levels within the video game.

Game

GF1. - The user will be able to load the game and complete a level.
GF2. - The user will be able to search for specific levels.
GF3. -The game will display to the user a list of the most popular levels.
GF4. - The game will give the user design elements such as enemies, pre-designed rooms, and aspects of the scenario.

Database / API

DF1. -Triggers will be in place to create all user-related data.
DF2. - The user credential shall be encrypted.
DF3. - Users should only be able to modify their own data.
DF4. - JSON Web Tokens should be used as a form of security to validate user identity.

II. Non-Functional Requirements

The website will give the user an immersive experience in the context of the videogame

- The video game will have fluid animations for every action performed: walk, die, shot.
- MySQL will be used for the database
- The application will be hosted in a remote server (free) such as Heroku.
- Typescript will be used to develop the rest API.
- The unity framework will be used to create the videogame
- The team will be comprised of three developers
- The time to develop the application will be 5 weeks.

SCRUM

I. Product Backlog

1. Develop website layout, and connect each site as expected.
2. Create error handling pages for any error that may occur.
3. Allow for user registration.
4. Embed the video game on a specified site.
5. Enable toggling animations.
6. Embed statistic querying on a specified site.

1. Develop user mechanics.
2. Develop enemy mechanics.
3. Develop map presets and environmental mechanics.
4. Develop builder system.
5. Implement DB and API on video game.

1. Define and implement triggers.
2. Implement user authentication.
3. Standardize DB.

II. Sprint #1 - Backlog

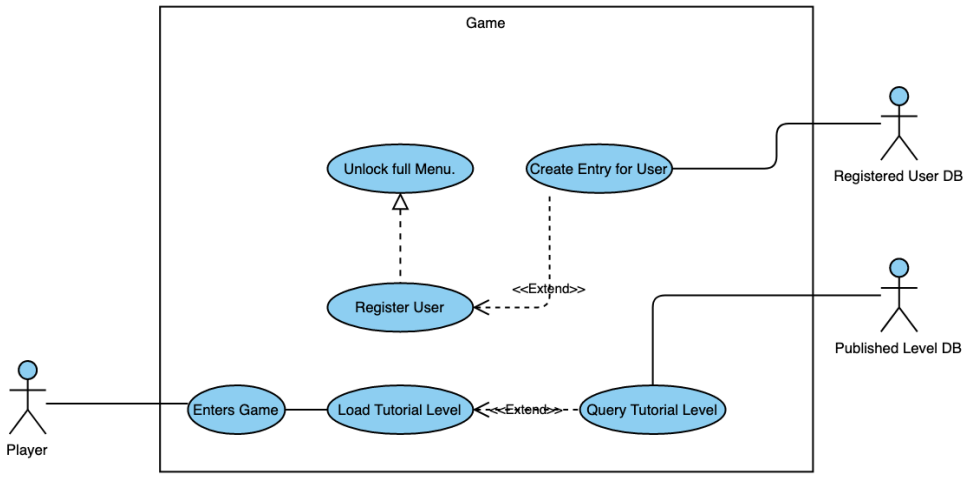
1. Develop website layout, and connect each site as expected.
    a. Create splashscreen, main site, about site, statistics site, and game site.
    b. Establish and implement layout for the content.
    c. Produce and equally cohesive design on the pages.
2. Create error handling pages for any error that may occur.
    a. Establish redirects on all pages should any error code appear.

1. Develop user mechanics.
    a. Implement 2D movement with expected keymap.
    b. Implement rotation with mouse.
    c. Implement shooting mechanic.
    d. Implement colliders with other bodies.
2. Develop map presets and environmental mechanics.
    a. Design multiple map presets.
    b. Restrict routing on maps.
    c. Implement boxes and other objects in the map.

Use Cases

Use Case Name	Complete tutorial.
Related Requirements	
Goal In Context	A user is familiarized with the platform.
Preconditions	The user has loaded the game successfully.
Successful End Condition	A user completes the tutorial level.
Failed End Condition	A user fails to complete the tutorial level.
Primary Actors	User
Trigger	The user plays the game for the first time.

Main Flow	Step	Action
	1	User enters the game for the first time.
	2	System loads Tutorial Level
	3	System loads Level Editor
	4	After Step 3 is completed successfully, User registers themself.
	5	After Step 4 is completed, user unlocks all the game modes.
Extension Flow	Step	Branching Action
	2.1	DB queries default Tutorial Level.
	4.1	DB creates entry for the newly registered user.

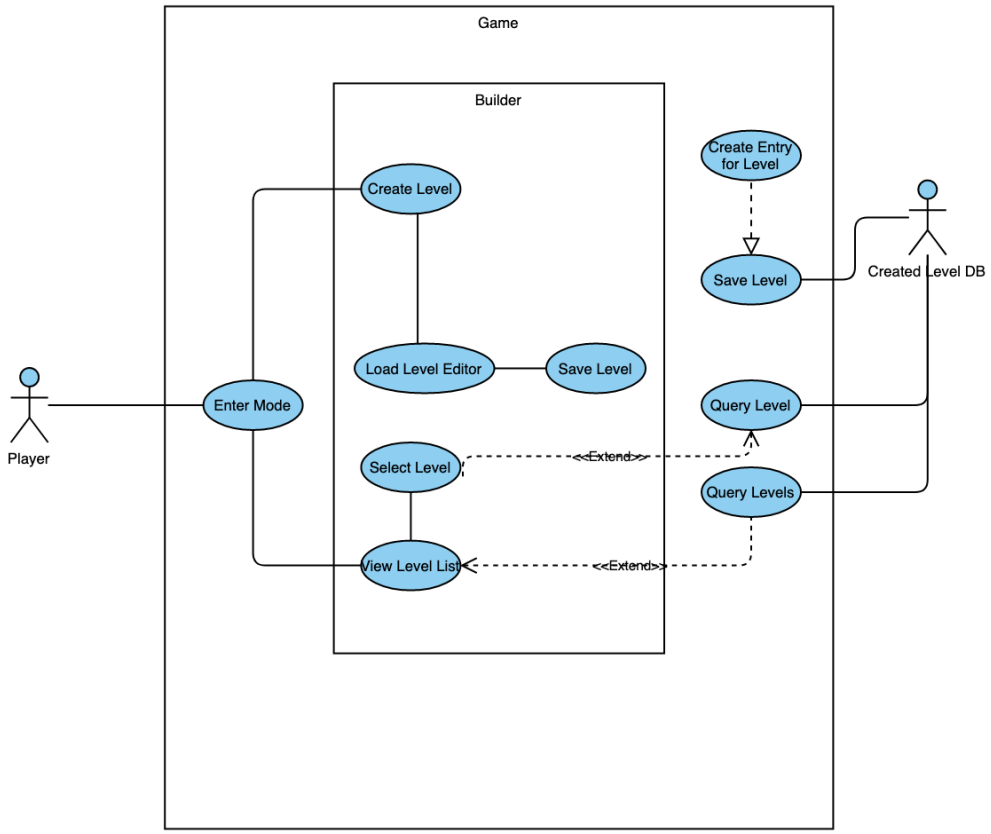


Use Case Name	Create a new level.
Related Requirements	
Goal In Context	A user builds and designs their own level.
Preconditions	The user has an accredited username. AND The user has passed the tutorial guide.
Successful End Condition	A user completes building a playable level.
Failed End Condition	A user fails to build a playable level.
Primary Actors	User
Trigger	The user enabled ‘build mode’ and selects the ‘New’ or ‘Load’ options.

Main Flow	Step	Action
	1	User enters the Builder Mode.
	2	User decides to create a new level.
	3	System loads Level Editor

	4	User decides to save the level.
	5	DB creates an entry for the new level.
	6	After Step 5 has been realized, DB stores level.

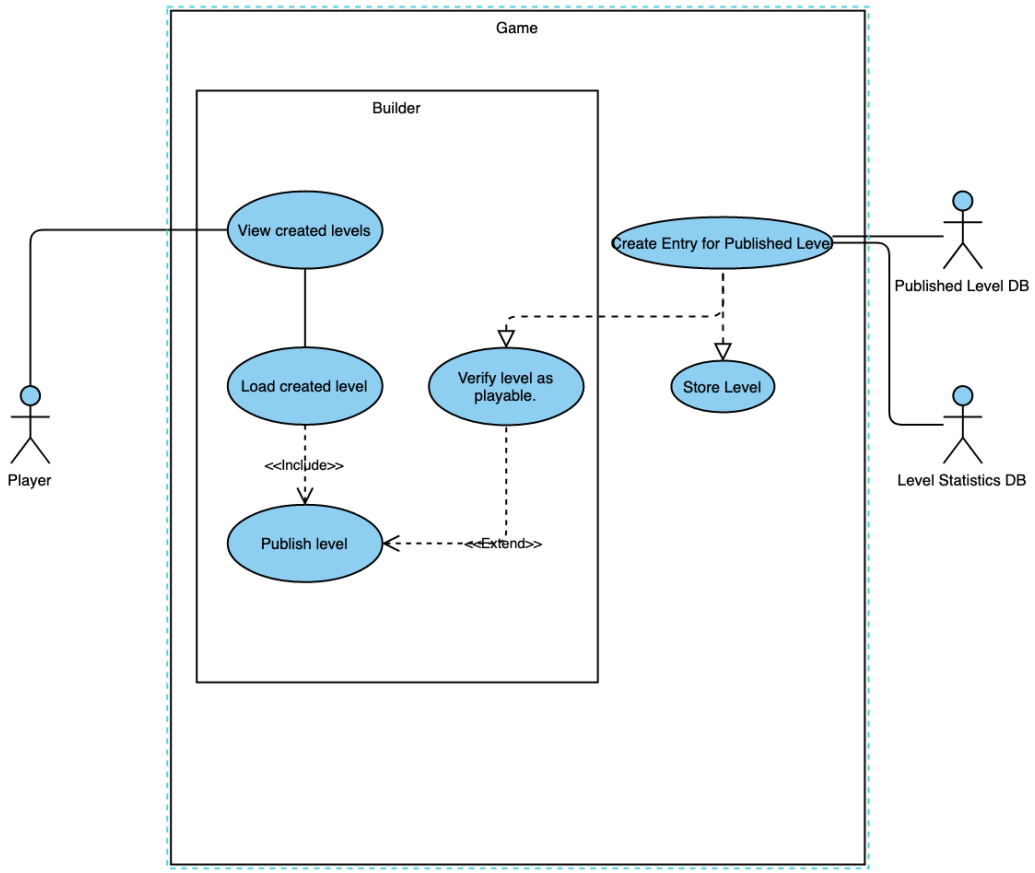
Main Flow	Step	Action
	1	User enters the Builder Mode.
	2	User views created levels.
	3	User decides to load a particular level.
	3	System loads Level Editor
	4	User decides to save the level.
	5	DB creates an entry for the new level.
	6	After Step 5 has been realized, DB stores level.
Extension Flow	Step	Branching Action
	2.1	DB queries all created levels.
	3.1	DB queries the selected level.



Use Case Name	Publish a new level.
Related Requirements	
Goal In Context	A user publishes a created level under their ID.
Preconditions	The user has an accredited username. AND The user has created a playable level.
Successful End Condition	A user publishes their level on the level list.
Failed End Condition	A user fails to publish their level.
Primary Actors	User
Trigger	The user enabled the ‘publish’ option.

Main Flow	Step	Action

	1	User enters the Builder Mode.
	2	User loads a created level.
	3	User decides to publish the level.
	4	After Step 3.1 has been realized, then the DBs create an Entry for the level.
	5	After Step 4 has been realized, then the DBs store the level.
Extension Flow	Step	Branching Action
	3.1	Builder verifies the level as playable.



Use Case Name	Play a published level.
Related Requirements	The user has passed the tutorial guide.
Goal In Context	A user loads and plays a level published by another creator.
Preconditions	The user has passed the tutorial guide.
Successful End Condition	A user loads a published level.
Failed End Condition	A user fails to load a published level.
Primary Actors	User
Trigger	The user selected the ‘community’ option. OR The user selected the ‘story mode’ option.

Main Flow	Step	Action
	1	User enters the Story Mode.
	2	User selects between differing levels of difficulty.
	3	Once, Step 2 has been realized, then the Published Level DB queries levels that match the difficulty selected.
	4	Game system loads the level for the Player.
Extension Flow	Step	Branching Action
	4.1	A player finishes the level.
	4.2	After Step 4.1 has been realized, then the statistics are collected and stored in the corresponding DB.

Main Flow	Step	Action
	1	User enters the Community Mode.
	2	User views the Level List
	3	User picks Level, and can apply filters.

	4	Game system loads the level for the Player.
Extension Flow	Step	Branching Action
	2.1	DB queries the level list.
	4.1	A player finishes the level.
	4.2	After Step 4.1 has been realized, then the statistics are collected and stored in the corresponding DB.