Instituto Tecnológico y de Estudios Superiores de Monterrey
Campus Santa Fe

Construcción de software y toma de decisiones
TC2005B, Grupo 400

Octavio Navarro Hinojosa
Dr. Esteban Castillo Juarez
Gilberto Echeverría Furió

Especificación de requerimientos de software

Equipo #2 | Integrantes:

Manuel Barrera López Portillo | A01570669

Miguel Angel Bustamante Pérez | A01781583

Juan Muniain Otero | A01781341

abril 2022

User stories:

User story #1 Player controller script As a player I want to be able to input commands and visualize feedback from my attacks.	
Validation ● Input commands set in the script work accordingly. Points and sprites work based on user input.	**Point value:** 100 maximum 0 minimum

User story #2 Add track manager script As a player I want a smooth gameplay experience when interacting with enemy sprites following the rhythm of the song.	
Validation ● Enemy sprites move according to the beat of the song. ● Sprites spawn at their given time. ● Attacking enemies at the correct time effectively destroys them. ● Enemies will despawn after failing to attack them.	**Point value:** 100 maximum 0 minimum

User story #3 Create midi samples As a player I want to have a musical system that interacts with the player and the sprites in order to have a fluid experience with the game.	
Validation ● Timing for every drum kit part should create an accurate sequence of percussion. ● Sprites should spawn according to their corresponding midi sequence.	**Point value:** 100 maximum 0 minimum

User story #4 Add columns script As a player I want to be able to differentiate and destroy enemies based on their corresponding lane spawn.	
Validation ● Enemies are restricted to their corresponding lane ● Certain inputs effectively destroy	**Point value:** 100 maximum 0 minimum

their corresponding enemy. ● Enemies spawn as prefabs in a given order and time.	

User story #5 Add scoring script As a player I want to be able to keep track of my performance throughout a level, and be able to obtain combos, and calculate accuracy.	
Validation ● ‘Good’, ‘perfect’, and ‘miss’ sprites spawn accordingly. ● Score updates accordingly when performing attacks. ● Combo resets when missing attacks. ● Accuracy is calculated at the end of the level based on accumulated hit and miss data.	**Point score:** max 100 points min 0 points

User story #6 Add combo / score hud As a player I want to be able to visualize my results at the end of a level.	
Validation ● Score text updates according to the attack result (good, perfect, miss). ● Information is displayed in an intuitive, colorful way for easy feedback.	**Point score:** max 100 points min 0 points

User story #7 Design Main Menu As a player I want a main menu that displays level options and difficulties.	
Validation ● The game should have a main menu which lets the player display a level select submenu. ● The game has to have another menu that allows the player to select the level difficulty.	**Point score:** max 100 points min 0 points

User story #8 Website Design As a user

I want that each time I enter the web page, I can access medical information / recommendations about drumming.	
Validation ● The website loads correctly. ● The website has an appropriate and intuitive design. ● The game runs smoothly within the site.	**Point score:** max 100 points min 0 points

User story #9 Setup MySQL queries As a database admin I want to be able to insert, extract, update, and delete information from the database in an efficient way.	
Validation ● Integrity restrictions are followed (updated information is not invalid, primary and secondary keys are set to prevent data repetition) ● Data manipulation in one table affects related tables which share foreign keys.	**Point score:** max 100 points min 0 points

User story #10 Setup MySQL tables As a database admin I want to be able to distribute information within tables where each represents game information.	
Validation ● Data is structured / normalized based on normal forms. ● Data is distributed within its entity / table.	**Point score:** max 100 points min 0 points

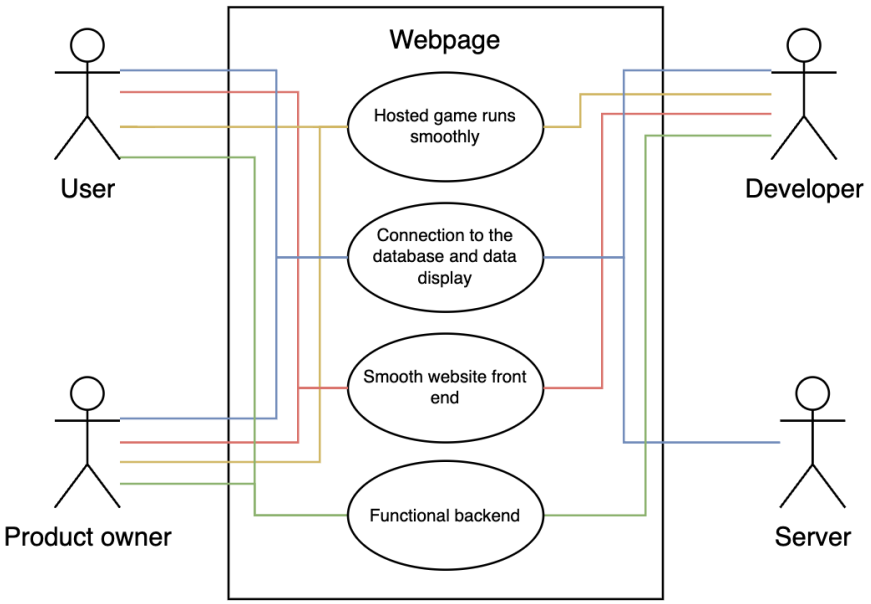
User story #11 Setup web-database link As a website admin I want to be able to extract information from the database and display it with statistics.	
Validation ● Player information generated after ending a level is reflected in the webpage. ● High score information is updated accordingly.	**Point score:** max 100 points min 0 points

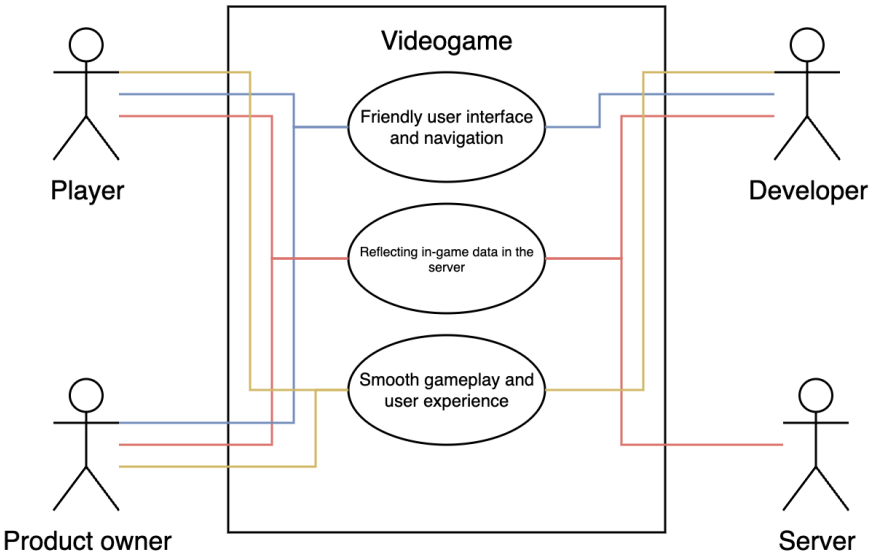
Cada óvalo del UML va a ser una tabla

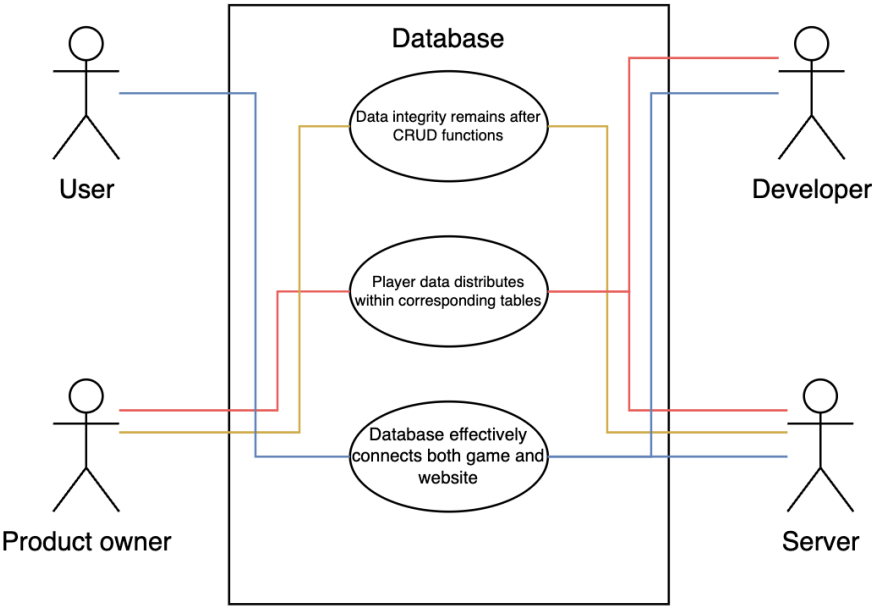
Functional requirements

- Add player controller script.
- Add track manager script.
- Create midi samples.
- Add columns script.
- Add scoring script.
- Add combo / score hud.
- Design main menu.
- Create website.
- Setup MySQL queries.
- Setup MySQL tables.
- Setup API.

UML Diagram







Use Case tables: Website

Use case name	Hosted game runs smoothly
Related Requirements	Requirements of Video Game development
Goal in Contex	To assure the full functionality of the Videogame in our web page
Preconditions	The Web Page must have its related assets fully included and as well functional
Successful end condition	The video game cycle runs correctly. The web page is interactive and has no visual glitches

<table>
<tr><td></td><td>The relevant data related to the database is taken and recorded correctly.</td></tr>
<tr><td>Primary Actors</td><td>User</td></tr>
<tr><td>Secondary Actor</td><td>Server</td></tr>
<tr><td>Trigger</td><td>The user access the website and then can access the game and relevant information of the game</td></tr>
<tr><td></td><td><table>
<tr><td>Step</td><td>Execution</td></tr>
<tr><td>1</td><td>the User enters the web page</td></tr>
<tr><td>2</td><td>The server loads the information relevant to the webpage
(html js css and game)</td></tr>
<tr><td>3</td><td>The page is loaded successfully</td></tr>
<tr><td>4</td><td>The user can interact with the page: move the page, click the relevant information without glitches.</td></tr>
<tr><td>5</td><td>The data base makes a successful connection</td></tr>
<tr><td>6</td><td>The database updates itself, getting the new imputed data.</td></tr>
<tr><td>7</td><td>The website stays in this state without problems</td></tr>
<tr><td>8</td><td>The user can close the website</td></tr>
</table></td></tr>
</table>

Use case name	Connection to the database and data

<table>
<tr><td></td><td>display</td></tr>
<tr><td>Related Requirements</td><td>Requirements related to the creation of the database</td></tr>
<tr><td>Goal in Contex</td><td>To record all highscores of the game over time to understand how people interact with percussion skills like rhythm detection</td></tr>
<tr><td>Preconditions</td><td>The database must have the correct parameters for its correct functioning such as normalization.</td></tr>
<tr><td>Successful end condition</td><td>The database has a proper connection to the webpage and receives through the internet the Highscores of each player</td></tr>
<tr><td>Primary Actors</td><td>Server</td></tr>
<tr><td>Secondary Actor</td><td>Database developer</td></tr>
<tr><td>Trigger</td><td>The player finishes playing a level of the game the High Score is recorded</td></tr>
<tr><td></td><td><table>
<tr><td>Step</td><td>Execution</td></tr>
<tr><td>1</td><td>The user loads the web page and plays the game</td></tr>
<tr><td>2</td><td>The user finishes a level</td></tr>
<tr><td>3</td><td>The web page communicates with the database</td></tr>
<tr><td>4</td><td>The web page inserts the username date and highscore of the user</td></tr>
<tr><td>5</td><td>The insertion is confirmed as correct</td></tr>
</table></td></tr>
</table>

Use case name	Smooth website front end
Related Requirements	The design of the Html and css files of the

<table>
<tr><td></td><td>website</td></tr>
<tr><td>Goal in Contex</td><td>To permit a functional, well design and informative website for the user</td></tr>
<tr><td>Preconditions</td><td>The correct arrangement in the server of the corresponding html css and js scripts.</td></tr>
<tr><td>Successful end condition</td><td>The website should load with the correct design without visual glitches or design errors</td></tr>
<tr><td>Primary Actors</td><td>Server</td></tr>
<tr><td>Secondary Actor</td><td>Errors</td></tr>
<tr><td>Trigger</td><td>The user accesses the server</td></tr>
<tr><td></td><td><table>
<tr><td>Step</td><td>Execution</td></tr>
<tr><td>1</td><td>The user accesses the website address</td></tr>
<tr><td>2</td><td>The html and css must load correctly</td></tr>
<tr><td>3</td><td>The website displays correctly</td></tr>
</table></td></tr>
</table>

Use case name	Functional Backend
Related Requirements	Smooth frontend functional
Goal in Contex	The augmentation of the functionality of the website to a more interactive version
Preconditions	The related js modules must be functional
Successful end condition	The interactive displays such as the game space must appear correctly
Primary Actors	Server
Secondary Actor	User Developer
Trigger	The user accesses the website

	<table><tr><td>Step</td><td>Execution</td></tr></table>

Use Case tables: Videogame

Use case name	Friendly user interface
Related Requirements	Functional requirements for the videogame
Goal in Contex	To make a more approachable experience for the user as it plays the videogame
Preconditions	The website must load correctly The website must load the game correctly
Successful end condition	The website will show the starting screen of the videogame.
Primary Actors	Server
Secondary Actor	User
Trigger	The user loads the website
	<table><tr><td>Step</td><td>Execution</td></tr><tr><td>1</td><td>The user enters the website</td></tr><tr><td>2</td><td>The server loads the website</td></tr><tr><td>3</td><td>The server loads the video game</td></tr><tr><td>4</td><td>The video game starts</td></tr><tr><td>5</td><td>The video game shows the</td></tr></table>

Use case name	Reflecting in game data to the database
Related Requirements	The development of the video game The creation of the database

	The website development
Goal in Contex	To acquire relevant information related to the user's performance in the videogame.
Preconditions	The loading of the video game The initialization of the database
Successful end condition	The database receives information relevant to the user
Primary Actors	Server Developer
Secondary Actor	User
Trigger	Finishing a video game level
	<table><tr><th>Step</th><th>Execution</th></tr><tr><td>1</td><td>The player finishes the level</td></tr><tr><td>2</td><td>The video game starts communicating with the database</td></tr><tr><td>3</td><td>The database records the High score</td></tr><tr><td>4</td><td>Database ends communication</td></tr></table>

Use case name	Smooth gameplay user experience
Related Requirements	Those related to the game development
Goal in Contex	If a game has no positive outlook nor a positive impact then it is useless as a user experience
Preconditions	The game must be uploaded to the webpage and also must be functional
Successful end condition	The game will end smoothly without glitches
Primary Actors	Server

	Developer
Secondary Actor	User
Trigger	The player loads the game and the start game screen appears
	<table><tr><th>Step</th><th>Execution</th></tr><tr><td>1</td><td>The game loads</td></tr><tr><td>2</td><td>The game initializes</td></tr><tr><td>3</td><td>The game runs</td></tr><tr><td>4</td><td>The game ends</td></tr></table>

Use Case tables: Database

Use case name	Data integrity remains after CRUD functions
Related Requirements	Setup MySQL queries Setup MySQL tables Add scoring script
Goal in Contex	To maintain data stability and prevent erroneous information with data integrity when inserting, deleting, updating or extracting from or to the database
Preconditions	Connection between game and website Initialization of the game Hardware for the server must be up and running Data types and integrity restrictions must be specified
Successful end condition	Player data is inserted, deleted, updated or extracted with no errors, and valid data is displayed within the game or website.
Primary Actors	Server, developer
Secondary Actor	Player
Trigger	<table><tr><th>Step</th><th>Execution</th></tr><tr><td>1</td><td>Player loads game</td></tr></table>

	2 | Player starts level 3 | Player generates data while playing the game 4 | Data is sent to the server 5 | Data is stored in the server 6 | Data is updated while the player generates more data 7 | Player ends level 8 | Data is displayed within the website

Use case name	Player data is distributed within corresponding tables
Related Requirements	Setup MySQL queries Setup MySQL tables Add scoring script
Goal in Contex	To effectively distribute player data generated while in gameplay and after, to its corresponding table following normal forms
Preconditions	Connection between game and website Initialization of the game Hardware for the server must be up and running Tables must be set up according to the data to be processed Tables must be normalized
Successful end condition	CRUD functions with player data do not generate errors Data can be visualized within the webpage
Primary Actors	Developer, Server
Secondary Actor	Product owner
Trigger	

	<table><tr><th>Step</th><th>Execution</th></tr><tr><td>1</td><td>Player loads game</td></tr><tr><td>2</td><td>Player selects level</td></tr><tr><td>3</td><td>Player generates data</td></tr><tr><td>4</td><td>Data is inserted or updated in the server</td></tr><tr><td>5</td><td>Data is extracted for display</td></tr></table>

Use case name	Database effectively connects both game and website
Related Requirements	Setup MySQL queries Setup MySQL tables Create website Setup API
Goal in Contex	To connect the game and website in order to manipulate user generated data and display it in the webpage.
Preconditions	Initialization of the game CRUD queries must be set up Hardware for the server must be up and running Tables must be set up according to the data to be processed
Successful end condition	Data sent and received in the server can be manipulated with CRUD functions after user or player interaction with the game and website.
Primary Actors	Developer, server
Secondary Actor	User
Trigger	<table><tr><th>Step</th><th>Execution</th></tr><tr><td>1</td><td>User enters webpage</td></tr></table>

	2	Player loads game
	3	Player loads level
	4	Player generates data
	5	Data is processed with CRUD functions
	6	Player ends level
	7	Data is displayed within the webpage