CSE 344: SOFTWARE ENGINEERING

YEDITEPE UNIVERSITY

TERM PROJECT 2021

SPRING

MARIO GAME

ANALYSIS REPORT

AHMET DEMIR

GAYE EMINE CELIK

GURKAN BEDEL

UGUR ORNEK

BURAK GÜNDÜZ

YAVUZ KARABIYIK

CONTENTS

INTRODUCTION

1.1 Purpose

The purpose of this document is to give detailed information about the 2d arcade like computer game Mario, it's a reflex themed game with graphical animations and different mods. Explains the functional, non-functional requirements of the mentioned software and its functionalities through different and set of diagrams and algorithms. Moreover it describes the software's domain, issues with the existing systems and more in this particular domain along with figured out solutions.

This projects purpose is creating an improved 2D Mario game that players will spend enjoyable time while playing our game.

1.2 Background

In modern world, computer is the well known software genre and a extremely-fast rising industry as you see in social media and news. The industry is the economic sector involved in the development, marketing, and monetization of video games. It encompasses dozens of job disciplines and its component parts employ thousands of people worldwide. According to research in 2017,video gaming business estimated with 28 billion $ and is expected to reach 33.6 billion $ by the end of 2020. Another ever-growing business in parallel with computer gaming is 'e-sports'. It is simply the short name for electronic sports. Just like football players play football together, e-sports players play computer games against each other. For big tournaments with well-known players, fans from all over the world may tune in to watch the action online, and competitors may even get paid for doing it. In 2019 ,DOTA 2 International, the largest tourney for players of Valve’s MOBA, currently has a prize pool of $30,200,471.That makes it the biggest ever offered for a single e-sports event, though that’s not really a surprise when you consider how the money is raised. To sum up , it is normal to say future is now and gaming industry has not reached it s full potential yet.

Today, video games make up a $100 billion global industry, and nearly two-thirds of American homes have household members who play video games regularly but before getting to that we must know the history behind it.

Figure A: DOTA

In 1952, for instance, British professor A.S. Douglas created *OXO*, also known as noughts and crosses or a tic-tac-toe, as part of his doctoral dissertation at the University of Cambridge. And in 1958, William Higinbotham created *Tennis for Two* on a large analog computer and connected oscilloscope screen for the annual visitor's day at the Brookhaven National Laboratory in Upton, New York.

Domain Analysis

This report describes background information that we will improve the mario game. This information is to be used to guide the development of software.

General knowledge about the domain

- Most of the young people play game during their free times.
- Prefered games generally easy to understand and enjoyable ones.
- Outdated and development has stopped games are of little interest.
- Group of interest are often created to target more precisely peoples that might be interested by a certain event.
- Games played on phones or directly on the web are more popular than oversized complex games.

Figure B: "Pong" arcade machine

Figure C: "PaCman" arcade machine

Although arcade machines and games were a huge success, computer gaming hardware still needed to scale down in order for computer games to be a part of the daily life. Also, an arcade machine had a single game installed in it which meant that you need to have multiple machines in order to play multiple games and that was a big problem. In 1977, Atari released the Atari 2600 (also known as the Video Computer System), a home console that featured joysticks and interchangeable game cartridges that played multi-colored games, effectively kicking off the second generation of the video game consoles. Instead of having built in games in its system, Atari allowed users to play any game with a ROM cartridge which has game's code stored in it. Eventually it became a huge success as over 30 million of devices were sold among with hundreds of millions of games.

Later on, as computer hardware technologies become more advanced, personal computers appeared which enhanced gaming industry with their memory, graphics and sound capabilities. On top of that, it was possible to save game progress on personal computers which allowed more complex games to be developed. This and oversaturated market along with many other reasons led to the downfall of arcade games. The video game home industry began to recover in 1985 when the Nintendo Entertainment System (NES), called Famicom in Japan, came to the United States. The NES had improved 8-bit graphics, colors, sound and gameplay over previous consoles. Nintendo, a Japanese company that began as a playing card manufacturer in 1889, released a number of important video game franchises still around today, such as *Super Mario Bros.*, *The Legend of Zelda*, and *Metroid*.

Figure D: Atari VCS(Video Computer System)

FIGURE E: Nintendo

Through the years, gaming consoles with better graphics, gameplay and memory space appeared such as the “Saturn”, “N64” and “Playstation”. While all were far better than their predecessors, they were still no match to the personal computers. Apart from hardware and performance precedence, personal computers were also more practical due to being operated by a mouse and a keyboard. Altogether, they were more eligible for complex and graphically demanding games than consoles.

Nowadays, computer games became so important that even hardware companies design and develop their products according to games’ needs. Moreover, people intend to check a top tier computer game’s specifications in order to run properly before buying or building computers. In 2005 and 2006, Microsoft’s Xbox 360, Sony’s Playstation 3, and Nintendo’s Wii kicked off the modern age of high-definition gaming. Though the Playstation 3—the only system at the time to play Blu-rays—was successful in its own right, Sony, for the first time, faced stiff competition from its rivals. Computer games became more and more demanding both performance and graphic wise since 2000s. They also became accessible to almost everyone from all sorts of age and professions. Despite being a rather new industry, computer games are considered as one of the innovations of the modern era.

1.3 Motivation

Mario is a game that is played by everyone around the world and is accepted as it is. Since it lost its former popularity over time, the aim here is to improve the missing parts of the game in order to adapt it to today's games and to enrich the game with features that will make it more enjoyable to play.

Problems with the Existing System

We looked into the existing 2D arcade games in our domain through internet and other gaming platforms in order to find their weak points.

Some of the problems we encountered:

1. Only 2 game mode

2. Lack of scenario or story

3. One user gameplay

First problem is only 2 game mode in our games. This situation may not be liked by the user since there are not many game modes. Although few game modes reduced the pleasure of the game, most 2D games do not have a game mode.

Second problem lack of story in 2D computer games. Even 2D games with best gameplay mechanics either didn't have a story. This was one of the reasons that affected the user's entertainment. We also saw that there was no information to be taught in some games and we could design a button about it (How to play button).

In our think: Mario is a single player game but single player games gives the same experience every time. Multiplayer games are more competitive than a single player games because it gives a real challenge for the player. Playing against other players all over the world is less repetitive and more entertaining. If Mario has a multiplayer option and users scores are kept on a scoreboard, it would be more competitive than playing alone.

Mario game does not have a selectable music there is only classical mario background music. Since there will be multiple types of music, players will be able to choose their favorite music type, which will provide them a more enjoyable gaming experience. This feature will prevent the player from getting bored when they fail to pass a level.

In the new system, we will add multiplayer option for comparing score at the end, music option and a scoreboard to the game.

The scores will be calculated based on the number of gold coins collected. The player who collects the gold most, will have the highest score.

For the music option there will be different types of music. There will be different boxes. Box 1 represents the rock music, box 2 represents the pop music and the box 3 represents the rap music. During the game, there will be different colored boxes, each will represent a different type of music. Such as rock music, pop music and rap music. The player will be able to choose music by jump and hit the color box represented by the type they want to listen to. The music will change when player hit the box with the head second time. And there will be a separate box to turn off the music completely.

1.4 Structure of the Document

In this project, the aim is to improve the Mario game. In the introduction part, a brief information about the game was given, the domain was examined. The motivation behind our desire to develop this game, the problems of the game and how they can be improved were discussed.

Second part of the document describes the requirements. The function of the system and what the system should do were explained, after that use case diagrams was drawn to summarize the relationship between use cases, actors and systems.

Also, class diagram was drawn to explain the functions performed by the system, analyze and design the static view of the game. Mock-ups were drawn by consider the necessary functions in order to facilitate the designers work.

FUNCTIONAL REQUIREMENTS

2.1 Description of System Functionalities

When we open in to our game, 3 special buttons appear;

- New Game
- How to Play ● Settings

These buttons are located on the main screen when you first enter the game. By pressing the new game button, a new game is opened. The new game button is the button that starts the main process. After the user presses this button, the player is directed to a new game. When the how to play button is clicked, information on how to play our game is available. This screen contains information about the basic functions of the game and which keys to play.

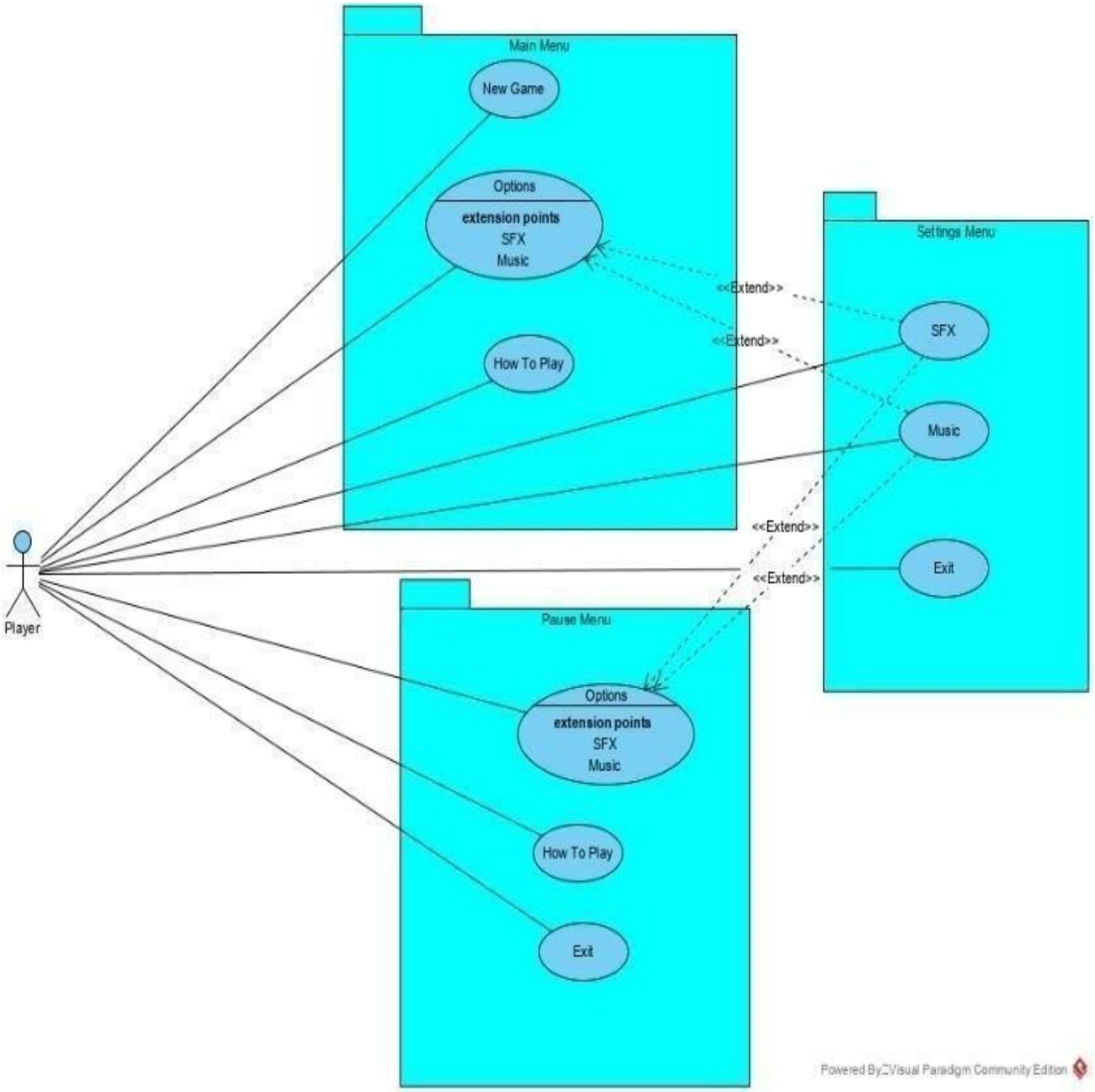
When the Settings button is pressed, some customizations are made within the game. "Music ON / OFF", "Sound ON / OFF" buttons are located here. The user can turn off the music in the game by pressing the "Music ON / OFF" button. When the "Sound ON / OFF" button is pressed, the effects in the game are turned off.

2.2 Description of System Users

Our software has a single player that interacts with the system. The user can change the game settings, navigate through menus, select game theme, pause the game and terminate the game. When the Pause button is pressed, the menus related to the game are accessed.

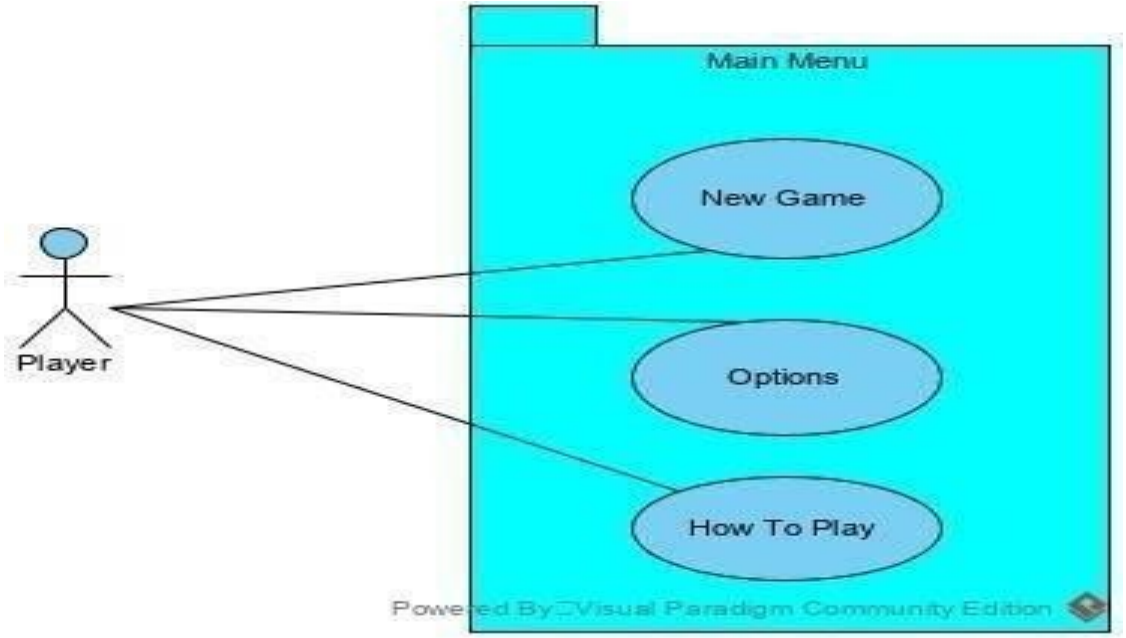
2.3 Specific Requirements

Use Case Diagram

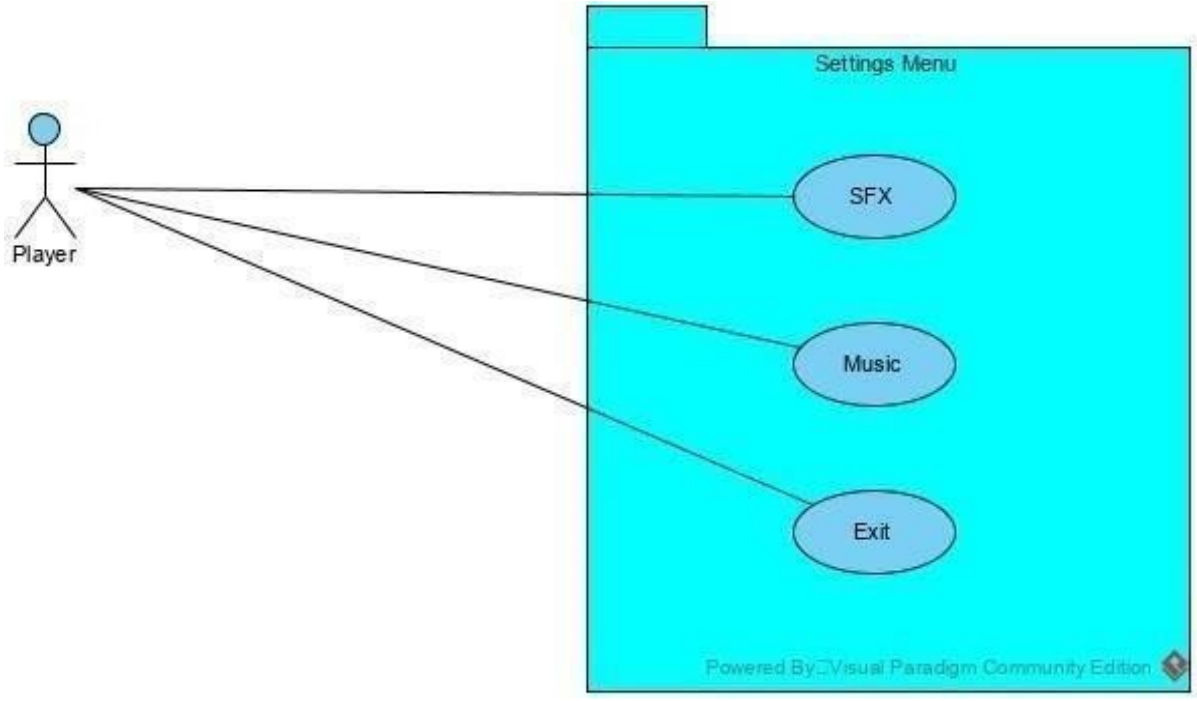


Full System Diagram

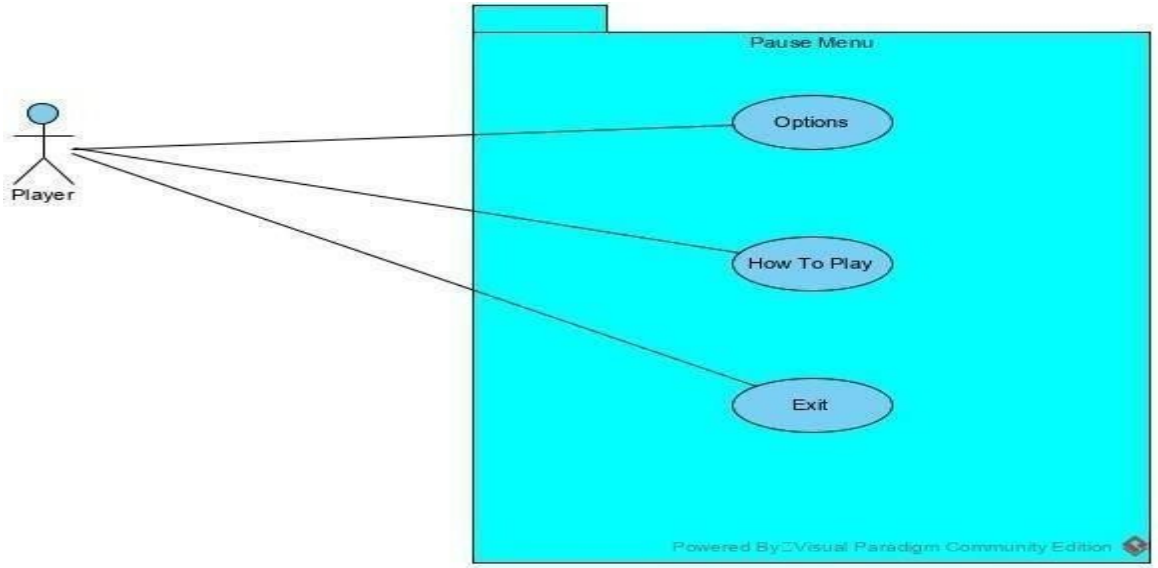
START-MENU



SETTINGS MENU



PAUSE MENU



Use Case Priority List

Priority Number	Description
1	Low Priority
2	Medium Priority
3	High Priority

START-MENU		
Use Case	**Priority**	**Rationale**
New Game	3	"New Game" is a high priority use case because user has to go through in order to play the game
Settings	2	"Settings" is a medium priority use case because it directly effects the user experience but it is also optional.
How To Play	2	"How To Play" is a medium priority use case because the game can be played without reading it but it would be much more clear with the information.

SETTINGS - MENU		
Use Case	**Priority**	**Rationale**
Background	1	"Background changing" is a low priority use case because it is optional and doesn't affect the flow of the game.
Sound	1	"Sound" Configuration is a low priority use case because it is optional and doesn't affect the flow of the game.
Back	1	Back is a low priority use case because it doesn't affect the flow of the game.

PAUSE-MENU		
Use Case	**Priority**	**Rationale**
Settings	2	"Settings" is a medium priority use case because it directly effects the user experience but it is also optional.
How To Play	2	How To Play is a medium priority use case because the game can be played without reading it but it would be much more clear with the information.
Back	1	Back is a low priority use case because it doesn't affect the flow of the game.

Use Case: New Game
ID: OUR_GAME_MAIN _01
Brief Description: The user starts a new game.
Primary Actors: The User
Secondary Actors: None
Segment Precondition: The use case starts when: • The user is on the Start Menu. **Segment Flow:**
1. The user clicks on “New Game” 2. Game Screen appears.
Segment Postcondition: This use case terminates when: • The user quits the game. • The user clicks on Back button.
Alternative Flows: None

Use Case: Settings
ID: OUR_GAME_MAIN _02
Brief Description: The user changes settings of the game.
Primary Actors: The User
Secondary Actors: None
Segment Precondition: The use case starts when: • The user is on the Start Menu. Or • The user is on the Pause Menu.
Segment Flow: 1. The user clicks on “Settings” 2. Settings Menu appears.
Extension Points: Sound Effects configuration Music configurations Back button
Segment Postcondition: This use case terminates when: The user quits the game. The user clicks on Back button.
Alternative Flows: None

Use Case: How to Play
ID: OUR_GAME_MAIN _03
Brief Description: The user checks how to play.
Primary Actors: The User
Secondary Actors: None
Segment Precondition: The use case starts when: The user is on the Start Menu. The user is on the Pause Menu.
Segment Flow: 1. The user clicks on “How to Play” 2. How to Play Screen appears.
Segment Postcondition: This use case terminates when: The user quits the game The user clicks on “Back” button.
Alternative Flows: None

Use Case: Music
ID: OUR_GAME_STNGS _01
Brief Description: The user changes music level.
Primary Actors: The User
Secondary Actors: None
Segment Precondition: The use case starts when: • The user is on the Settings Menu.
Segment Flow: The user uses the slider for setting the Music sound.
Segment Postcondition: This use case terminates when: The user quits the game The user clicks on Back button. The user returns to Settings Menu.
Alternative Flows: None

Use Case: Sound Effects
ID: OUR_GAME_STNGS _02
Brief Description: The user mutes or unmutes sound effects.
Primary Actors: The User
Secondary Actors: None
Segment Precondition: The use case starts when: • The user is on the Settings Menu.
Segment Flow: 1 - The user clicks on “Off” to mute Sound Effects. 2- The user clicks on “On” to unmute Sound Effects.
Segment Postcondition: This use case terminates when: The user quits the game The user clicks on Back button. The user returns to Settings Menu.
Alternative Flows: None

Use Case: Back
ID: OUR_GAME_STNGS _03
Brief Description: The user quits the game.
Primary Actors: The User
Secondary Actors: None
Segment Precondition: The use case starts when: • The user is on the Settings Menu. • The user is on the Pause Menu
Segment Flow: The user cliks on “Back”.
Segment Postcondition: This use case terminates when: The user quits the game. The user clicks on Back button. The user returns to Settings Menu.
Alternative Flows : None

NON-FUNCTIONAL REQUIREMENTS

3.1 Volere Template

Requirement: Collecting Coins

Requirement Type: Non-Functional

Requirement #: 1001

Description: The game's scoring system is based on collecting coins on the ground. So the more coins the player collects, the more points he gets.

Rationale: This is very important as the game's scoring is based on collecting coins.

Fit Criterion: An extra software can be used to calculate how much the coins have been collected.

Requirement: Killing Enemies

Requirement Type: Non-Functional

Requirement #: 1002

Description: In order to maintain the continuity of the game, it is necessary to stay away from enemies.

Rationale: In order for the game to continue, it must stay away from enemies. The game is over when you touch the enemies.

Fit Criterion: Extra features can be added to kill enemies like weapons.

Requirement: Replay Value

Requirement Type: Non-Functional

Requirement #: 1003

Description: The game should make users want to play itself multiple times

Rationale: A game without replay values is a mostly failed game, hence this is why this requirement is a very important one.

Fit Criterion: Adding features such as multiple game modes and making the game endless and challenging. We can make the game more challenging by introducing multiple modes (eg level or environment). Thus, the game can be competitive.

Requirement: Usability

Requirement Type: Non-Functional

Requirement #: 1004

Description: Controls of the game should be easily understandable by the users who play this game.

Rationale: The user should be comfortable with the game enough that they don't back out from playing because of the controls.

Fit Criterion: The game has a "How To Play" screen, showing the user how to play the game. On this screen, there is detailed information about which keys and how to play.

Requirement: Reliability

Requirement Type: Non-Functional

Requirement #: 1005

Description: The game should not have any bugs or glitches that can harm player's desire to play it.

Rationale: A game with major glitches and bugs considered a bad designed game, so this is an important requirement if we want to have our game to be successful.

Fit Criterion: The game should be tested by many people with different playstyles to find any glitches and/or bugs.

Requirement: Required Resources

Requirement Type: Non-Functional

Requirement #: 1006

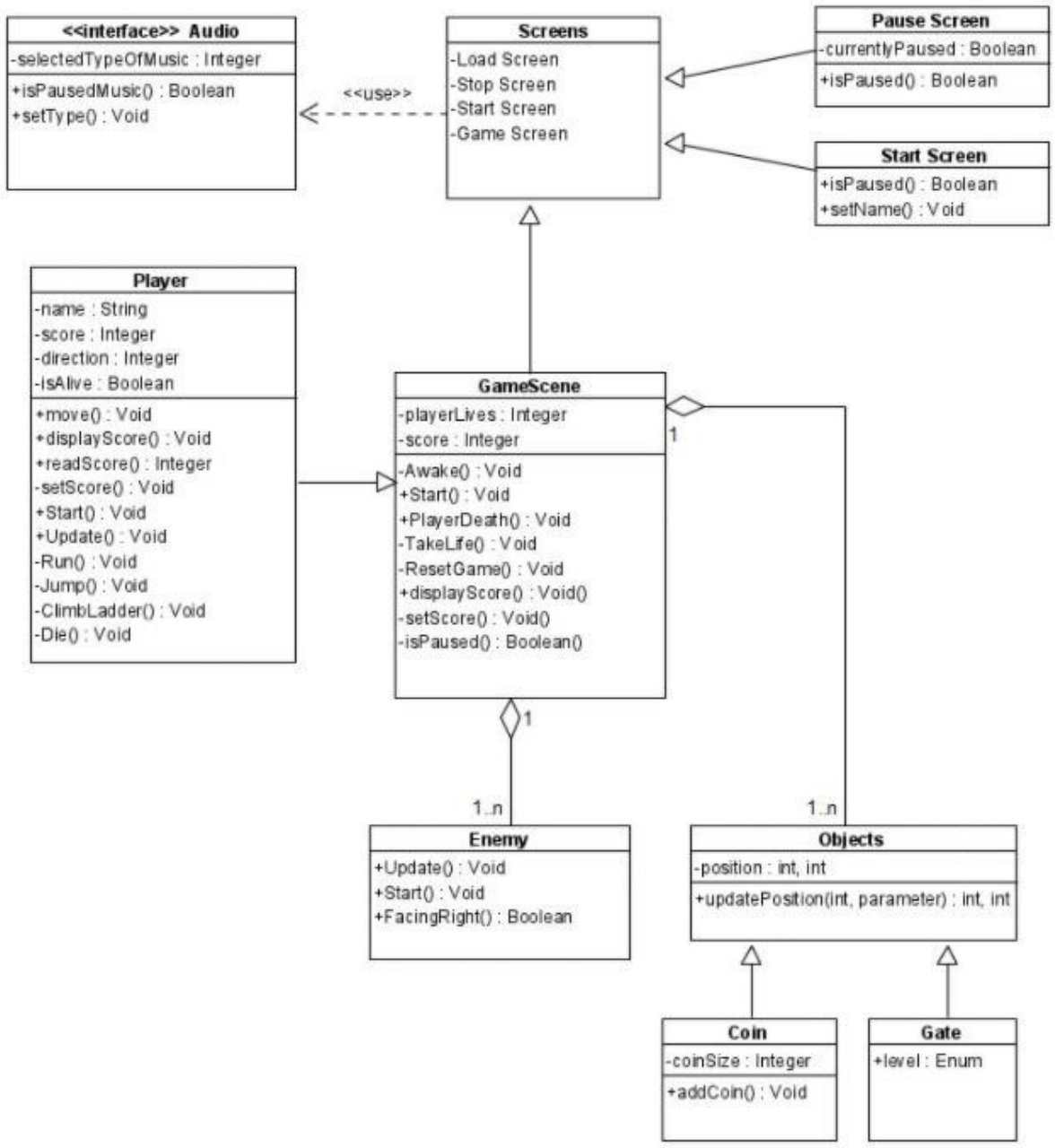
Description: The game should limit itself to have a reasonable amount of RAM and space. (i.e maximum 1024 MBs of RAM and 1GB of disk space)

Rationale: Many browsers today consume a lot of ram. That's why the game should not overwhelm the devices's system by having too much RAM or disk space.

Fit Criterion: As developers, we must minimize the amount of memory a game uses with the browser.

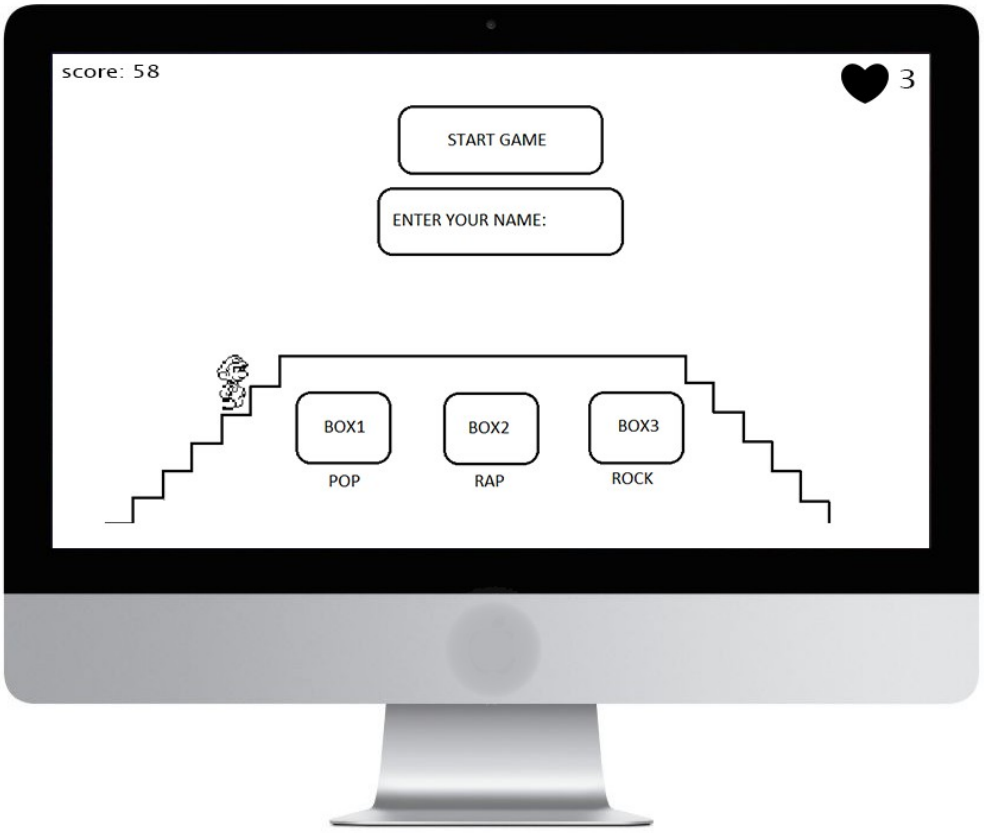
SYSTEM MODELS

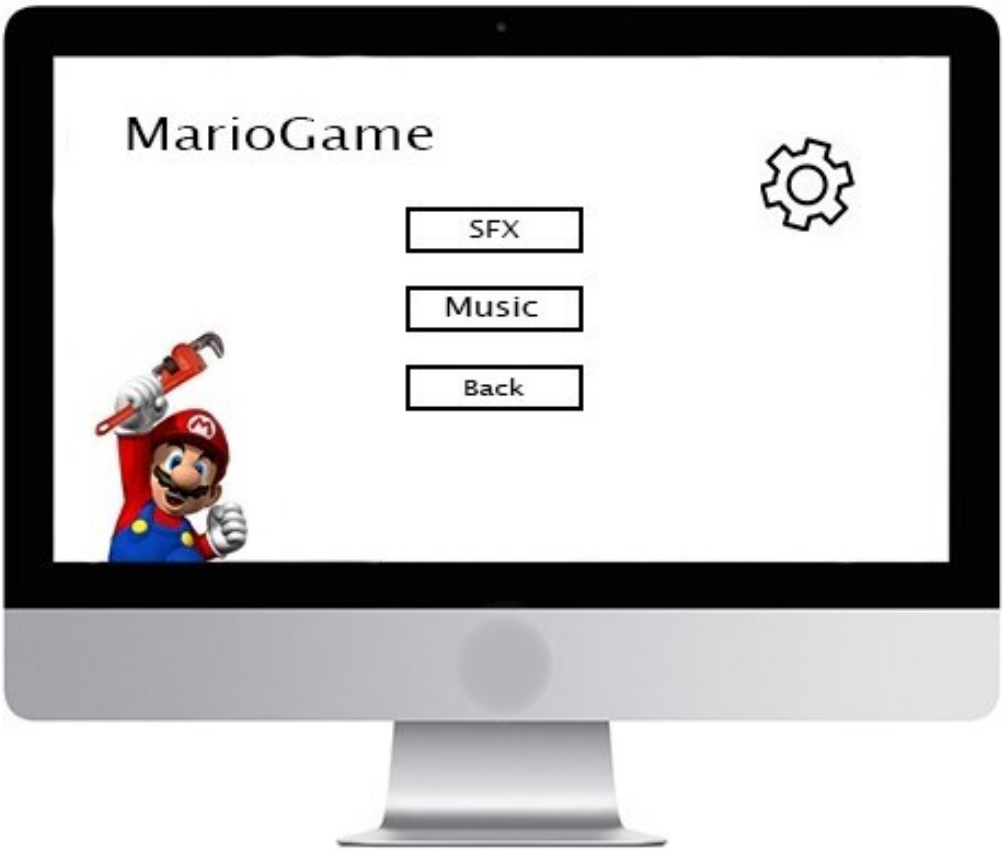
4.1 Object and Class Model



4.2 User Interface

4.2 USER INTERFACE




MarioGame
SFX
Music
Back

5. DEFINITIONS, ACRONYMS AND ABBREVIATIONS

abstraction. The separation of the logical properties of data or function from its implementation in a computer program. See: encapsulation, information hiding, software engineering.

algorithm. (IEEE) (1) A finite set of well-defined rules for the solution of a problem in a finite number of steps. (2) Any sequence of operations for performing a specific task.

algorithm analysis. (IEEE) A software V&V task to ensure that the algorithms selected are correct, appropriate, and stable, and meet all accuracy, timing, and sizing requirements.

analysis. (1) To separate into elemental parts or basic principles so as to determine the nature of the whole. (2) A course of reasoning showing that a certain result is a consequence of assumed premises. (3) (ANSI) The methodical investigation of a problem, and the separation of the problem into smaller related units for further detailed study.

application software. (IEEE) Software designed to fill specific needs of a user; for example, software for navigation, payroll, or process control. Contrast with support software; system software.

array. (IEEE) An n-dimensional ordered set of data items identified by a single name and one or more indices, so that each element of the set is individually addressable; e.g., a matrix, table, or vector.

coding. (IEEE) (1) In software engineering, the process of expressing a computer program in a programming language. (2) The transforming of logic and data from design specifications (design descriptions) into a programming language. See: implementation.

data. Representations of facts, concepts, or instructions in a manner suitable for communication, interpretation, or processing by humans or by automated means.

database. (ANSI) A collection of interrelated data, often with controlled redundancy, organized according to a schema to serve one or more applications. The data are stored so that they can be used by different programs without concern for the data structure or organization. A common approach is used to add new data and to modify and retrieve existing data. See: archival database.

design. (IEEE) The process of defining the architecture, components, interfaces, and other characteristics of a system or component. See: architectural design, preliminary design, detailed design.

development methodology. (ANSI) A systematic approach to software creation that defines development phases and specifies the activities, products, verification procedures, and completion criteria for each phase. See: incremental development, rapid prototyping, spiral model, waterfall model.

documentation, software. (NIST) Technical data or information, including computer listings and printouts, in human readable form, that describe or specify the design or details, explain the capabilities, or provide operating instructions for using the software to obtain desired results from a software system. See: specification; specification, requirements; specification. design; software design description; test plan, test report, user's guide.

implementation. The process of translating a design into hardware components, software components, or both. See: coding.

interface. (1) (ISO) A shared boundary between two functional units, defined by functional characteristics, common physical interconnection characteristics, signal characteristics, and other characteristics, as appropriate. The concept involves the specification of the connection of two devices having different functions. (2) A point of communication between two or more processes, persons, or other physical entities. (3) A peripheral device which permits two or more devices to communicate.

modeling. Construction of programs used to model the effects of a postulated environment for investigating the dimensions of a problem for the effects of algorithmic processes on responsive targets.

requirement. (IEEE) (1) A condition or capability needed by a user to solve a problem or achieve an objective. (2) A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed documents. (3) A documented representation of a condition or capability as in (1) or (2). See: design requirement, functional requirement, implementation requirement, interface requirement, performance requirement, physical requirement.

reliability. (IEEE) The ability of a system or component to perform its required functions under stated conditions for a specified period of time. See: software reliability.

software. (ANSI) Programs, procedures, rules, and any associated documentation pertaining to the operation of a system. Contrast with hardware. See: application software, operating system, system software, utility software.

system design. (ISO) A process of defining the hardware and software architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. See: design phase, architectural design, functional design.

Glossary of video game Terminology

2D graphics

Graphic rendering technique in a two-dimensional perspective, often using sprites.

adaptive music

Game music which changes and reacts to the actions of the player and/or what is happening in the game.[10]

jump

A basic move where the player jumps vertically.

level

1. A location in a game Also *area*, *map*, *stage*, dungeon. Several levels may be grouped into a world. Some games include special bonus stages or secret levels.
2. A character's experience level in a role-playing game, which increases through playing the game to train a character's abilities. It serves as a rough indicator of that character's overall proficiency.

3. A round or wave in a single-location game with increasing difficulty.

life

One of multiple chances that a player has to retry a task after failing. Losing all of one's lives is usually a loss condition and may force the player to start over. It is common in action games for the player-character to have multiple lives and chances to earn more during the game. This way, a player can recover from making a disastrous mistake. Role-playing games and adventure games usually give the player only one life, but allow them to reload a saved game if they fail.[65][66] A life may similarly be defined as the period between the start and end of play for any character, from creation to destruction.[67

multiplayer

A game that allows multiple players to play at once.

single-player

A game that allows sinle players to play at once.

6. REFERENCES

1) https://www.history.com/topics/inventions/history-of-video-games

2) https://tr.wikipedia.org/wiki/Video_oyunu

3) https://www.computerhope.com/history/game.htm

4) https://www.history.com/topics/inventions/history-of-video-games

5) https://electronics.howstuffworks.com/video-game2.htm

6) https://en.wikipedia.org/wiki/Dota_2

7) https://www.museumofplay.org/about/icheg/videogame-history/timeline

8) http://www.icollector.com/Coin-operated-arcade-machine-Baby-Pac-Man- by-Bally-c-1980-not-in-working-cond-68-H-x-23-W-x-37-D_i8652398 *(Image Source)*

9) https://en.wikipedia.org/wiki/Atari_2600 *(Image Source)*

10) https://techcrunch.com/2015/10/31/the-history-of-gaming-anevolving-community/

11) https://www.statista.com/topics/868/video-games/

12) https://en.wikipedia.org/wiki/Nintendo_Entertainment_System *(Image Source)*
13) https://www.joystixgames.com/product/pong/ *(Image Source)*

14) https://github.com/oleksandrmerchanskyi/SuperMario/wiki/Super-MarioUml-Class-Diagram

15) https://creately.com/diagram/example/hen1gqi6/Super%20Mario
16) Ian Sommerville: Software Engineering - Timothy C. Lethbirdge: ObjectOriented Software Engineering