Software Requirements Specification

for

<Capturing Hand Gestures using Leap Motion Controller and games in rehabilitation of motor >

Version <1.0>

Prepared by

Group Name: <Team O(n)>

<Dursun Alperen Çoban>	<1800003537>	<1800003537@stu.iku.edu.tr>
<Mertcan Ercan>	<1800004219>	<1800004219@stu.iku.edu.tr>
<Yunus Emre Güneş>	<1800001868>	<1800001868@stu.iku.edu.tr>
<Muhammet Furkan Özara>	<1800001915>	<1800001915@stu.iku.edu.tr >
<Deniz Hacızade>	<1800000110>	<1800000110@stu.iku.edu.tr>

Github: https://github.com/yunusemregunes/CSE0462

Instructor: **<Mehmet Demir>**

Course: **<CSE0462>**

Date: **<06/01/2023>**

Contents

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
Initial Draft	Muhammet Furkan Özara(1.1),(1.2),(1.3),(1.4), Mertcan Ercan (1.2), Dursun Alperen Çoban(1.2), Yunus Emre Güneş (1.2)	Made the introduction of the project and prepared a kind of infrastructure. Preliminary preparations have been completed. Revisions part done. Functional and Non-functional requirements were determined. Designing the use cases and other diagrams and made formatting changes. All diagrams completed.	02/05/2022
Version 1.0	Dursun Alperen Çoban(2.1),(2.2),(2.2.1),(2.2.2),(2.2.3)	We wrote the functional and non-functional requirements.	06/05/2022
Version 1.1	Mertcan Ercan (4.1) Yunus Emre Güneş (4.2),(4.3),Muhammet FurkanÖzara (4.3.1)	Designing the use cases and other diagrams and made formatting changes	10/05/2022
Version 1.2	Mertcan Ercan (5)	We have completed the design of the interfaces.	15/05/2022
Version 1.3	Yunus Emre Gunes(6)Mertcan Ercan(7)	Final checking operations were made and it was made ready to use by users.	22/05/2022

Version	Primary Author(s)	Description of Version	Date Completed
Initial Draft	Deniz Hacızade Yunus Emre Güneş Muhammet Furkan Özara (1.2),(1.3),(1.4)	Made the introduction of the project and prepared a kind of infrastructure. Preliminary preparations have been completed. Revisions part done. Functional and Non-functional requirements were determined. Designing the use cases and other diagrams and made formatting changes. All diagrams completed.	01/05/2022
Version 1.0	Mertcan Ercan (2)	We wrote the functional and non-functional requirements.	05/05/2022
Version 1.1	Dursun Alperen Çoban (4.1) Mertcan Ercan (4.2),(4.3) Deniz Hacızade(4.3.1)	Designing the use cases and other diagrams and made formatting changes	12/05/2022
Version 1.2	Deniz Hacızade Yunus Emre Güneş (5)	We have completed the design of the interfaces.	16/05/2022
Version 1.3	Muhammet Furkan Özara (6),(7)	Final checking operations were made and it was made ready to use by users.	21/05/2022

1 Introduction

1.1 Project Purpose and Scope, and Objectives

The purpose of this project is to rehabilitate people with motor disorders using leap motion sensor in video games.

Our objective is to support people with motor disorders with our video game. The patient's rehabilitation will be tracked with the point and time system in the game. Also we will see how the patient progresses with the different level difficulties.

Our scope is to reach out and rehabilitate every person with motor disorders.

1.2 Roles and responsibilities

1-Muhammet Furkan Özara: Designed and implemented log and score system. Creating the Throw counter system. The scoring and shot counter system is integrated into the game. Logging system and exporting of the game has been done.
2-Dursun Alperen Çoban: Designed background according to a story, find suitable assets to use in the background, cubes and baskets. Added atmospheric sounds. Created a timer system.
3-Mertcan Ercan: Arranged the cubes physics, set up leap motion, created a scene change system, designed a menu.
4-Yunus Emre Güneş: Arranged the baskets physics, created animations for all the levels, designed a menu.

MAZE GAME
1-Deniz Hacızade: Adjusted the collision and collection mechanics, regulated background music and sound effects, design of menu and levels
2-Muhammet Furkan Özara: Created the scene changer mechanic, prepared the skeleton of the levels,design of cutscene and levels
3-Yunus Emre Güneş: Prepared the skeleton of the levels,integrated time and scoring system

1.3 Technical Assumptions and Constraints

-Leap Motion Sensor
-Unity Version 2021 9.0 or more
-Computer with 2GB ram or more
-4GB empty space
-Windows 8 or more
-2GB Graphics Card

1.4 Naming Conventions

We use CamelCase as a practice for writing names of methods, variables, classes, packages, constants, e.g.
Class → UpperCamelCase
Method → lowerCamelCase
Local variables→ lowerCamelCase
For example, ScoreViewer.
stopwatch -> timerActive
throwCounter -> throwCount

2 Requirements

2.1 Functional Requirements

Logging system: It stores all the records in the game. It stores them as txt files within the game files. In this way, the player can compare himself on all systems in the previous games he played.

Scene change system: With this system, it is possible to switch between levels without interruption.

Menu interface: It is the screen that appears when the user enters the application and starts the game. With this screen, the player can play the game. Likewise, at the end of the game, the ending menu appears and guides the player at the end of the game.

Animation system: It is the system required for the objects to move in the game.

Scoring system: It allows the player to score points according to his performance in the game.

Throw counter system: This system counts how many throws the player has made in the game.

Time counter system: This system calculates how long it takes the player to complete the level.

2.2Non-Functional Requirements

2.2.1 Performance Requirements

The system loads at an acceptably fast rate. The loading time is minimized both when opening the game and between levels. Since the game can be played offline, the speed of the game is not connected to the internet.

The minimum system requirements of the game are at an acceptable level. It can be easily played on today's computers.

The game can be played independently from the internet. It can be played offline after a one-time download.

Since the game logs are kept in the txt file, the log files do not take up much space and are system friendly.

2.2.2 Safety and Security Requirements

Precautions for Play Area: Open up an empty space for yourself to play comfortably. You will probably move around while playing the game, so be careful that all areas that you might move into are clear. Make sure that no objects or people are around you while the game is being played, so you don't accidentally bump into them while playing. Also, it is recommended to stay at least one (1) meter from the monitor.

General Health and Eyestrain: Avoid playing the game excessively. Take breaks for an average of 10 minutes every hour, even if you don't think you need it.

2.2.3 Software Quality Attributes

Reliability: The system saves every record no matter what. Even if the game is interrupted, it saves the scoring of the part up to where the player came from. however, there will be no data loss or deletion in the game, no matter what. All data is saved in txt file.

Usability: By keeping the gameplay of the game simple, we made a design that the players can easily get used to. Removed factors that might confuse the player.

Maintainability: The game codes were kept easy to understand on the next developer groups. In this way codes can be easily developed and upgraded.

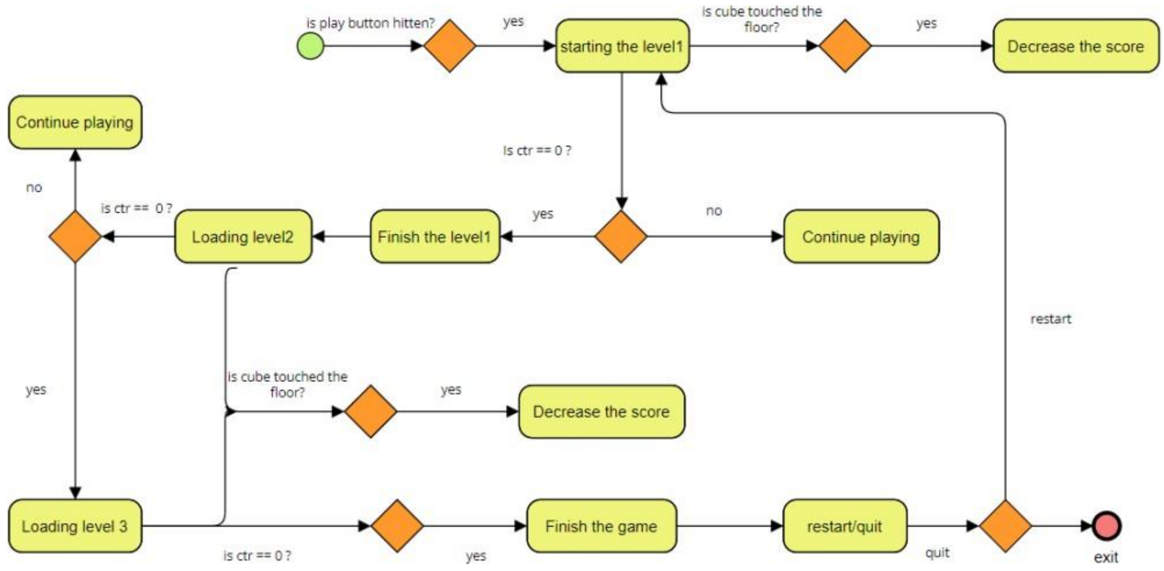
Portability: Since unity is used while making the game, the range of platforms where the game can be played is wide. You can play on many platforms.

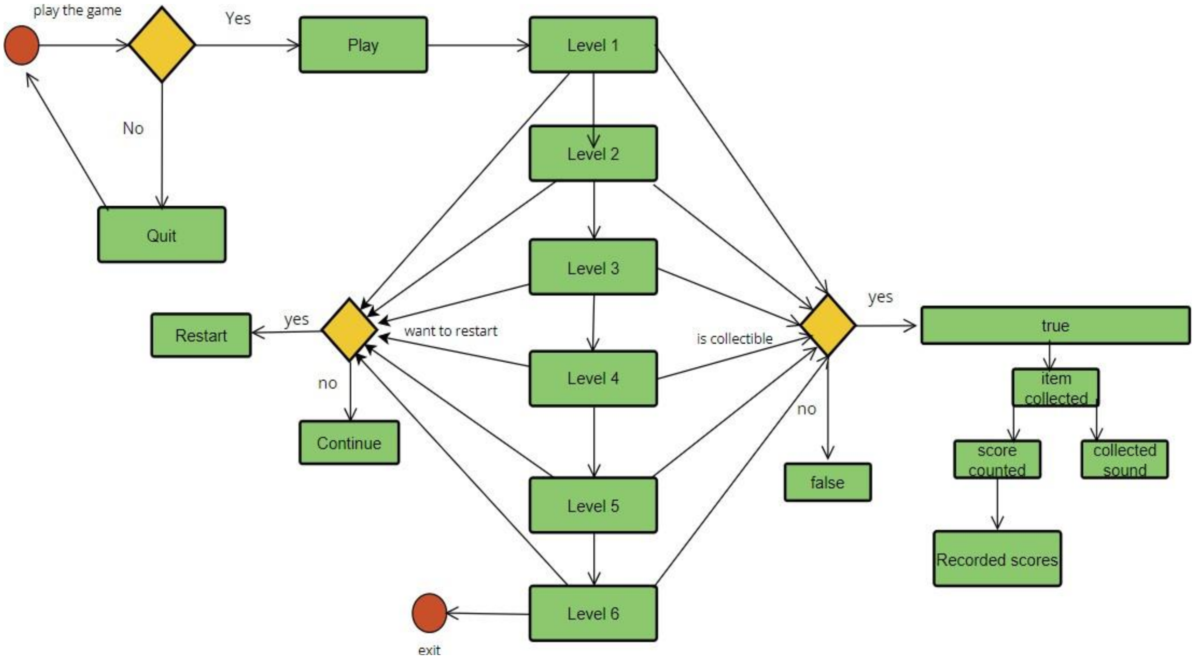
Testing: We made our test process using Unity Test runner extension. 1238 different tests have been performed and 1228 scenarios have been succeeded.

3 Other Requirements

4 System Architecture and Architectural Design

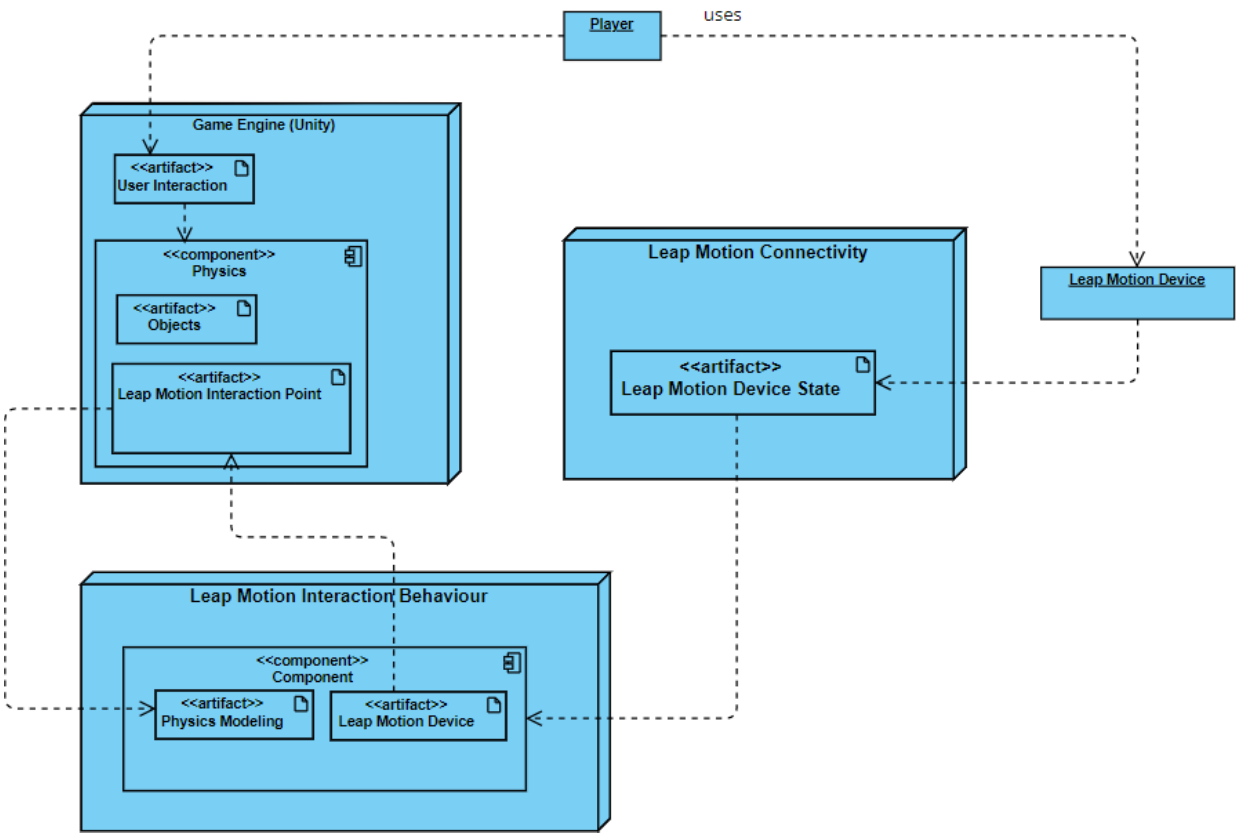
4.1 Logical View




play the game
Yes
Play
Level 1
No
Level 2
Quit
Level 3
yes
Restart
want to restart
is collectible
yes
true
Level 4
no
item collected
no
Continue
score counted
collected sound
false
Level 5
Recorded scores
Level 6
exit

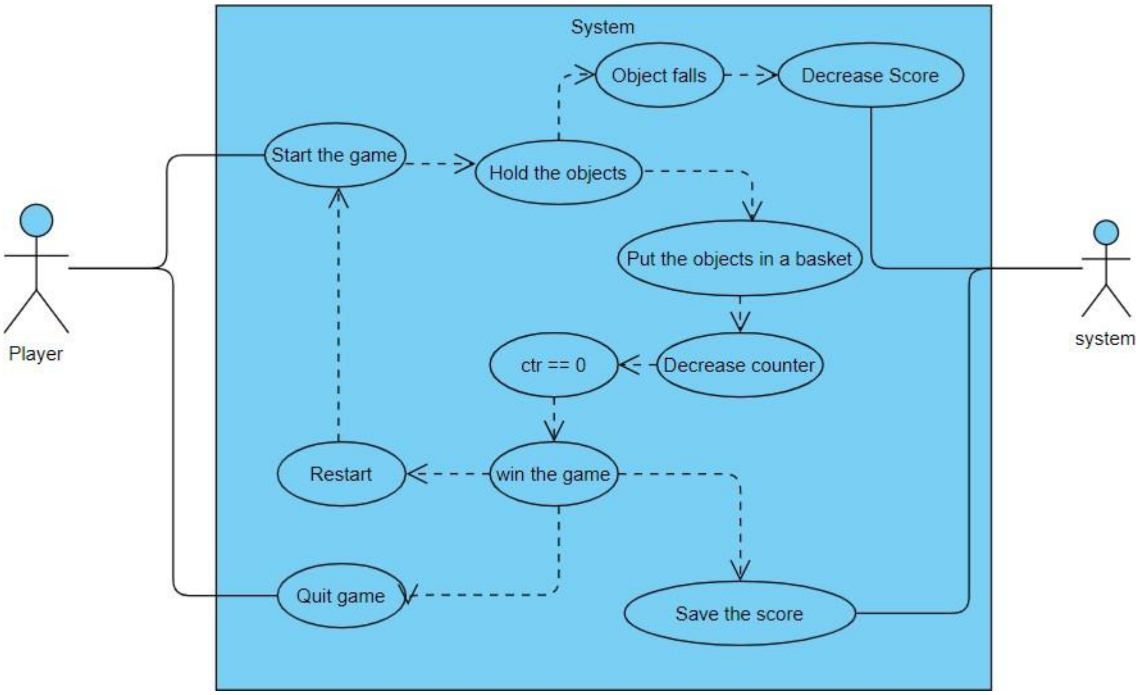
MAZE GAME

4.2 Deployment View

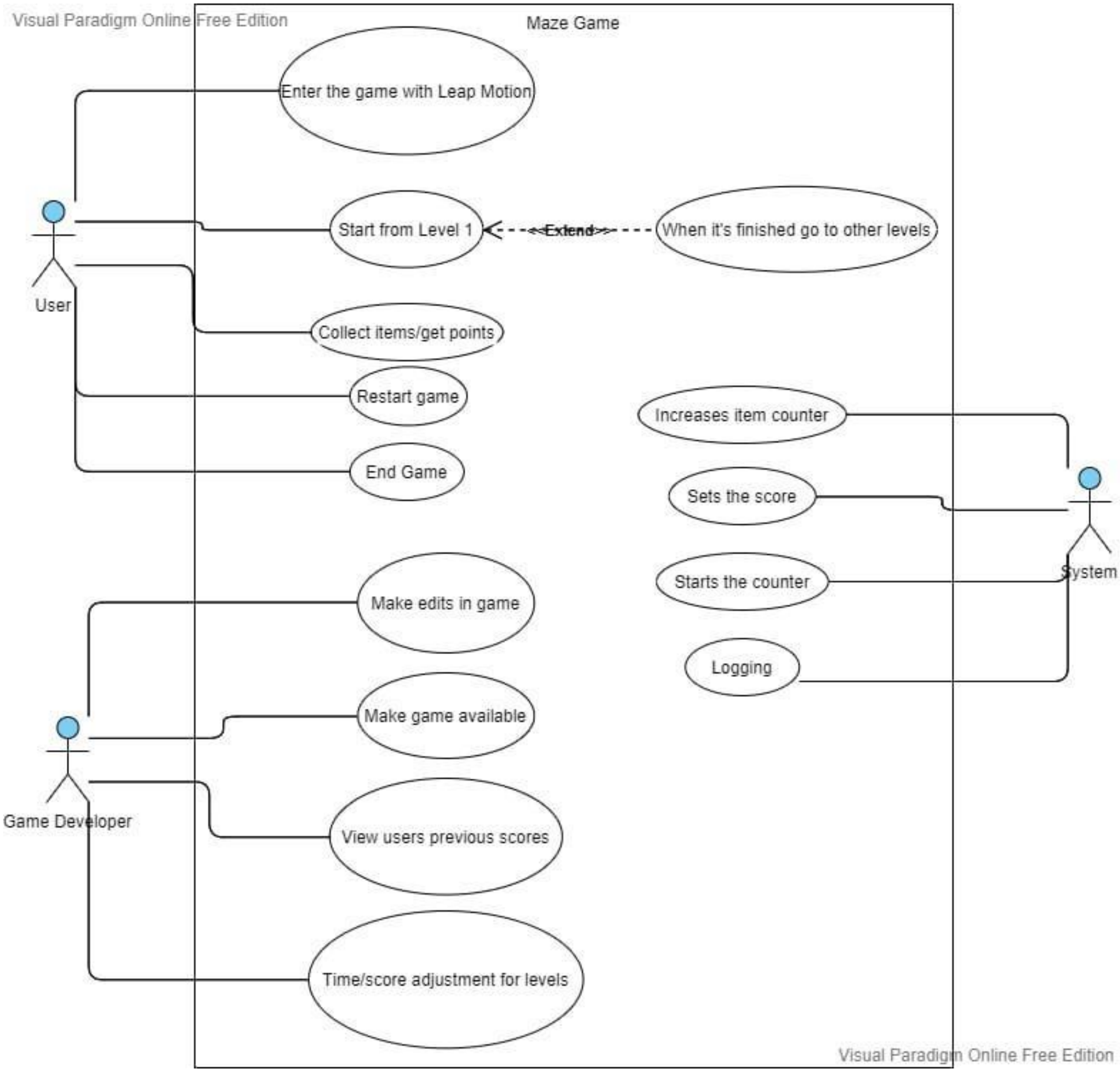


(CUBE GAME-MAZE GAME)

4.3 Use Case View



(CUBE GAME)



MAZE GAME

4.3.1 Use Case Scenarios

Use case name	*Easy level*
Scenario	*Patient throws cubes to the basket*
Triggering event	*Patient enters the main menu and clicks the start button.*
Brief description	*Patient must throw all the cubes into the basket to reach to the medium level.*
Actors	*Patient and System*
Related use cases	*None*
Stakeholders	*None*
Preconditions	*Patient must have a leap motion sensor*
Postconditions	*Easy level must be completed*

Flow of Activities	*PATIENT 1-Enters the main menu. 2-Starts the game. 3-Picks up cubes and throws them into the basket successfully. 4-Watches the basket to load into the ship. 5-Watches the ship to leave the port. SYSTEM 1-Shows the Point 2-Shows the Time 3-Saves the point and trying number to the log file. 4-Loads the medium level.*
Exception Conditions	*-If patient can not menage to throw the cubes into the basket, patient must start the game again.*

Use case name	*Medium level*
Scenario	*Patient throws white logs to the white cart, brown logs to the brown cart.*
Triggering event	*Patient passes the easy level*
Brief description	*Patient must throw right logs to the right cart to reach hard level.*
Actors	*Patient and System*
Related use cases	*Easy level*
Stakeholders	*None*
Preconditions	*Patient must pass the easy level*
Postconditions	*Medium level must be completed*
Flow of activities	*PATIENT 1-Picks up white logs and throws them into white cart. 2-Picks up brown logs and throws them into brown cart. 3-Watches woodcutters to move wood carts. SYSTEM 1-Shows the Point 2-Shows the Time 3-Saves the point and trying number to the logfile. 4-Loads the hard level.*
Exception conditions	*-If patient can not manage to throw right logs into the right cart, patient must start the game again.*

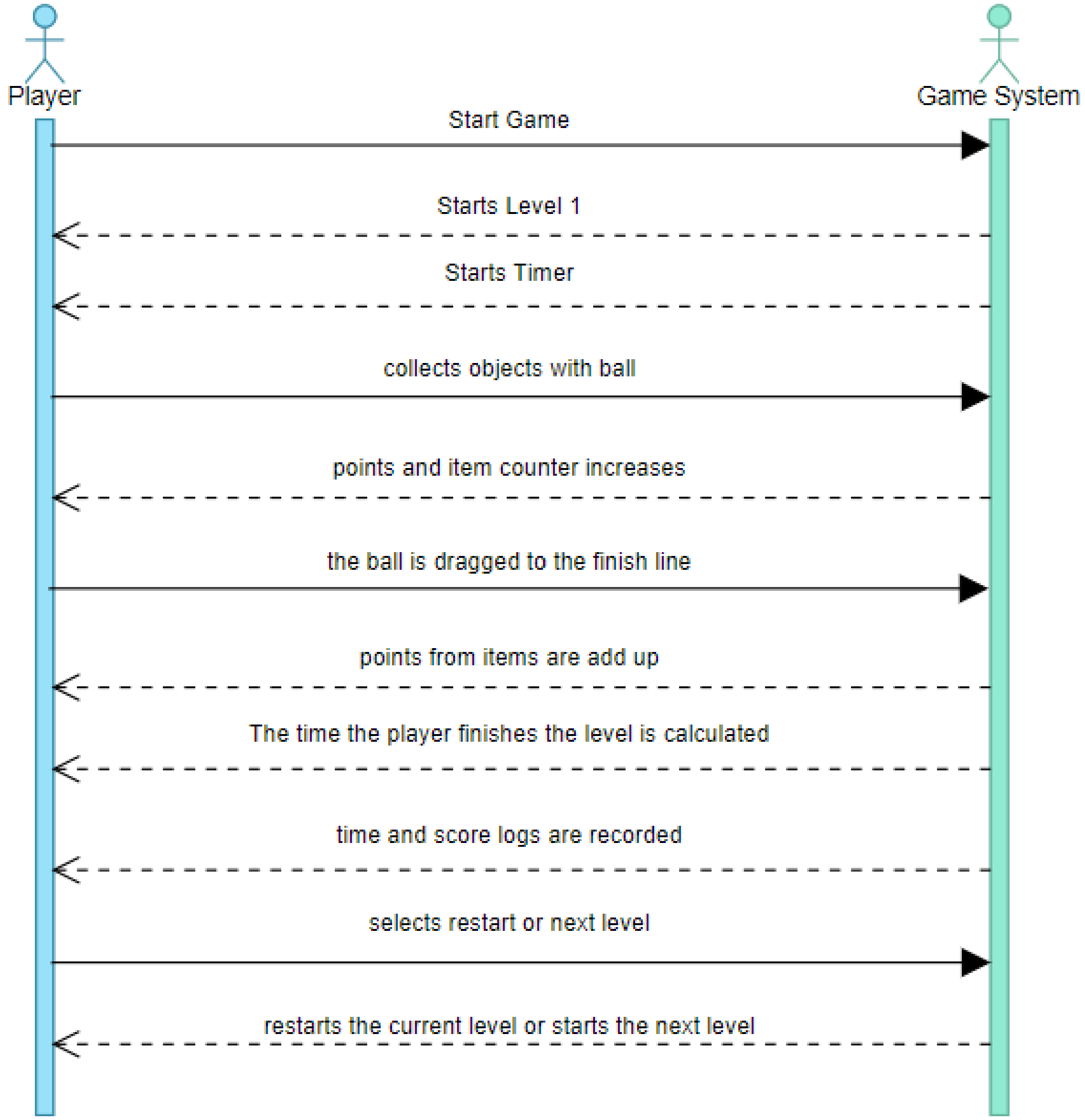
Use case name	*Hard level*
Scenario	*Patient throws wooden buckets into moving minecart.*
Triggering event	*Patient passes medium level*
Brief description	*Patient must throw wooden buckets into minecart at right time and right place to complete the game.*

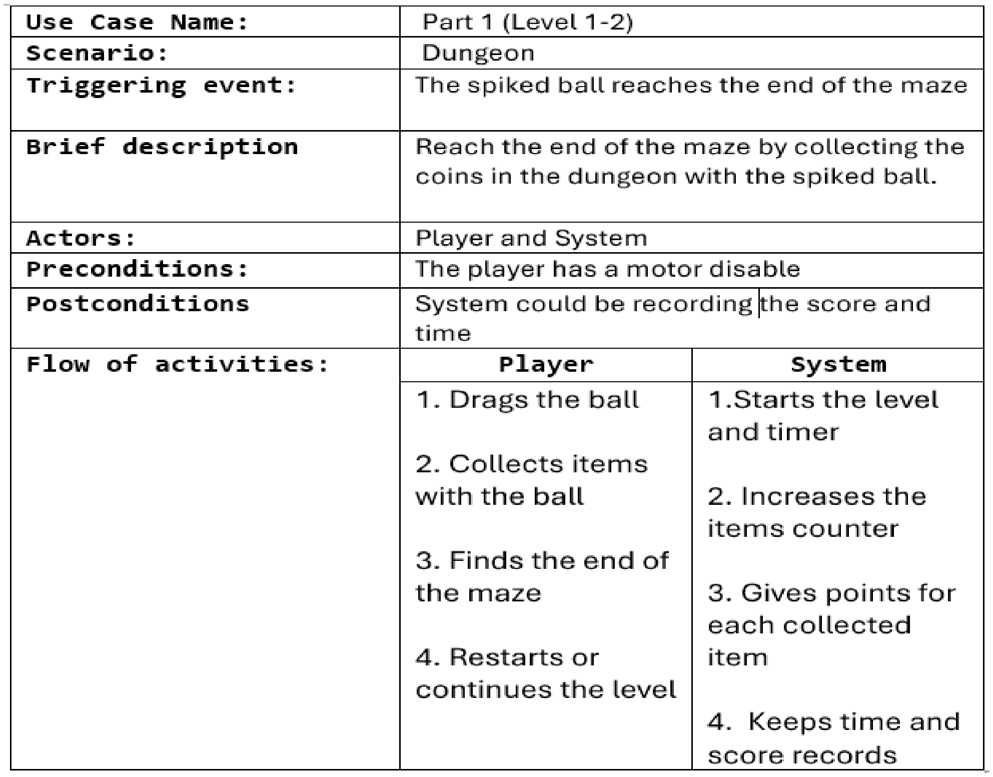
Actors	Patient and System
Related use cases	Easy level, Medium level
Stakeholders	None
Preconditions	Patient must pass the medium level
Postconditions	Hard level must be completed
Flow of activities	PATIENT 1-Picks up wooden buckets and waits for the right time. 2-Throws the buckets into the minecart. 3-Watches minecart to reach to the mine. SYSTEM 1-Shows the Point 2-Shows the Time 3-Saves the point and trying number to the log file.
Exception conditions	-If patient can not manage to throw the wooden buckets into the minecart at the right time, patient must start the game again.

Use case name	Check out the log
Scenario	Patient checks out the logs to see their improvement.
Triggering event	Patient searches and enters the log file.
Brief description	Patient must find the right log file in the computer to check out.
Actors	Patient and System
Related use cases	Easy level, Medium level, Hard level
Stakeholders	None
Preconditions	Patient must complete the game.
Postconditions	None
Flow of activities	PATIENT 1-Completes the game 2-Searches the log file 3-Finds the log file 4-Enters the log file SYSTEM 1-Saves the point and trying number to the log file.
Exception conditions	None

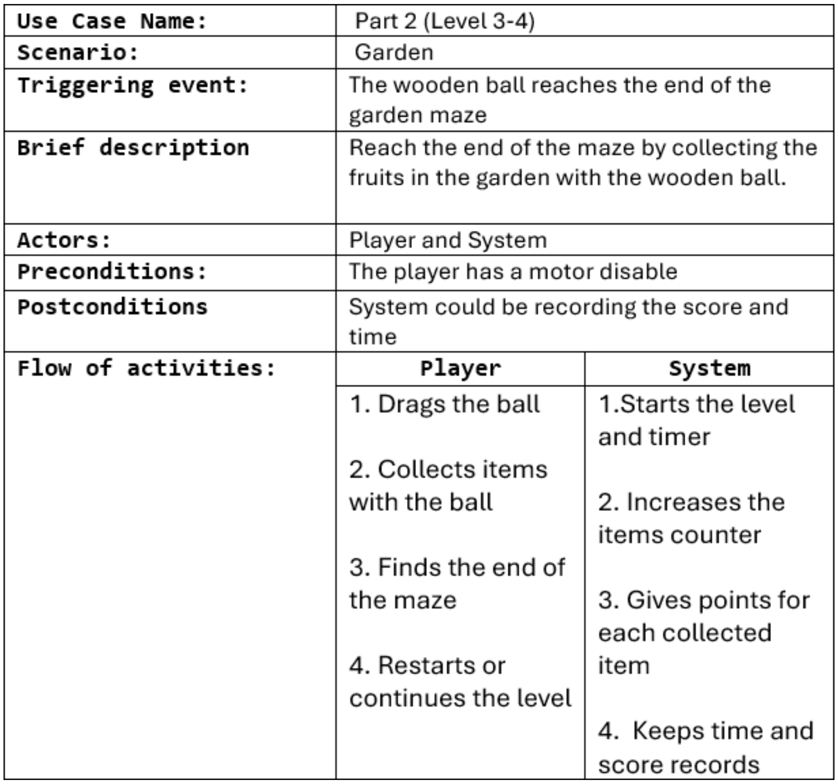
(CUBE GAME)

MAZE GAME

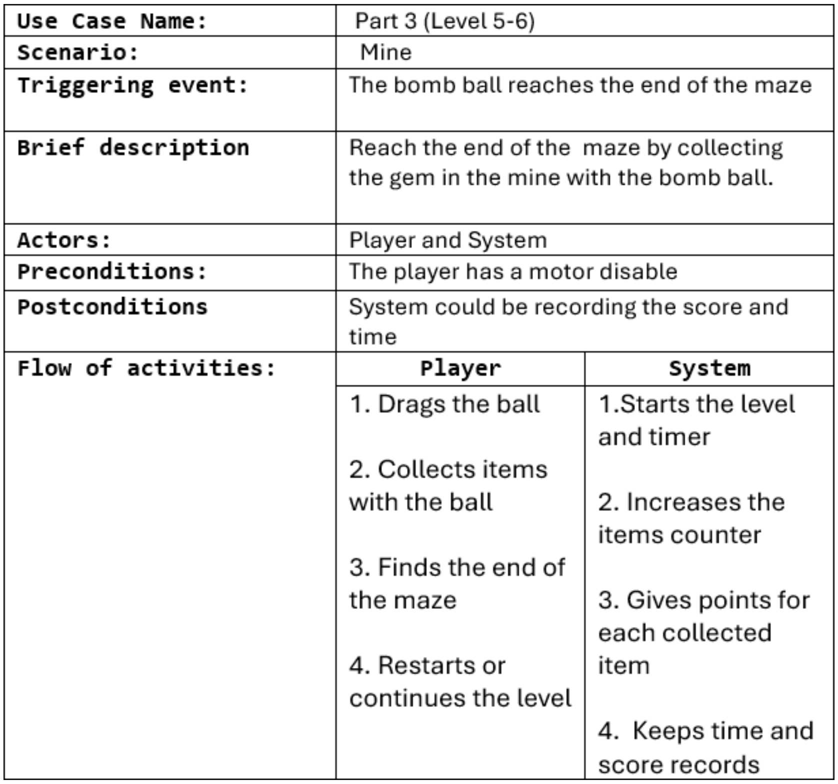

Player
Game System
Start Game
Starts Level 1
Starts Timer
collects objects with ball
points and item counter increases
the ball is dragged to the finish line
points from items are add up
The time the player finishes the level is calculated
time and score logs are recorded
selects restart or next level
restarts the current level or starts the next level



Use Case Name:	Part 1 (Level 1-2)	
Scenario:	Dungeon	
Triggering event:	The spiked ball reaches the end of the maze	
Brief description	Reach the end of the maze by collecting the coins in the dungeon with the spiked ball.	
Actors:	Player and System	
Preconditions:	The player has a motor disable	
Postconditions	System could be recording the score and time	
Flow of activities:	**Player**	**System**
	1. Drags the ball 2. Collects items with the ball 3. Finds the end of the maze 4. Restarts or continues the level	1.Starts the level and timer 2. Increases the items counter 3. Gives points for each collected item 4. Keeps time and score records

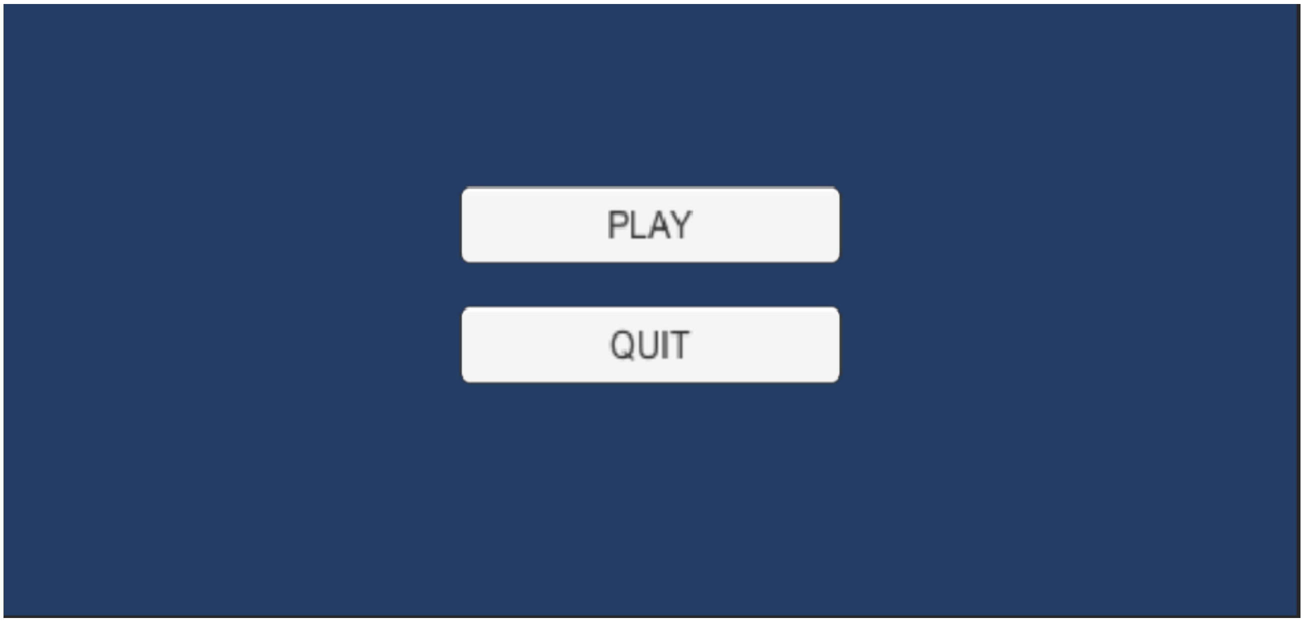


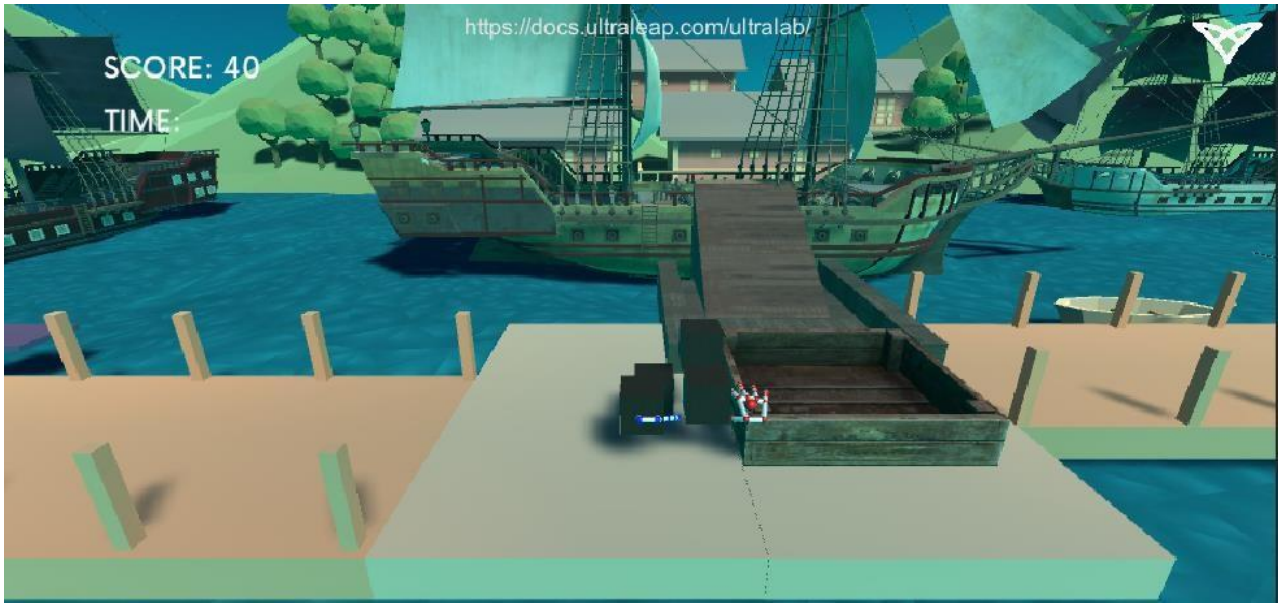
Use Case Name:	Part 2 (Level 3-4)	
Scenario:	Garden	
Triggering event:	The wooden ball reaches the end of the garden maze	
Brief description	Reach the end of the maze by collecting the fruits in the garden with the wooden ball.	
Actors:	Player and System	
Preconditions:	The player has a motor disable	
Postconditions	System could be recording the score and time	
Flow of activities:	**Player**	**System**
	1. Drags the ball 2. Collects items with the ball 3. Finds the end of the maze 4. Restarts or continues the level	1.Starts the level and timer 2. Increases the items counter 3. Gives points for each collected item 4. Keeps time and score records

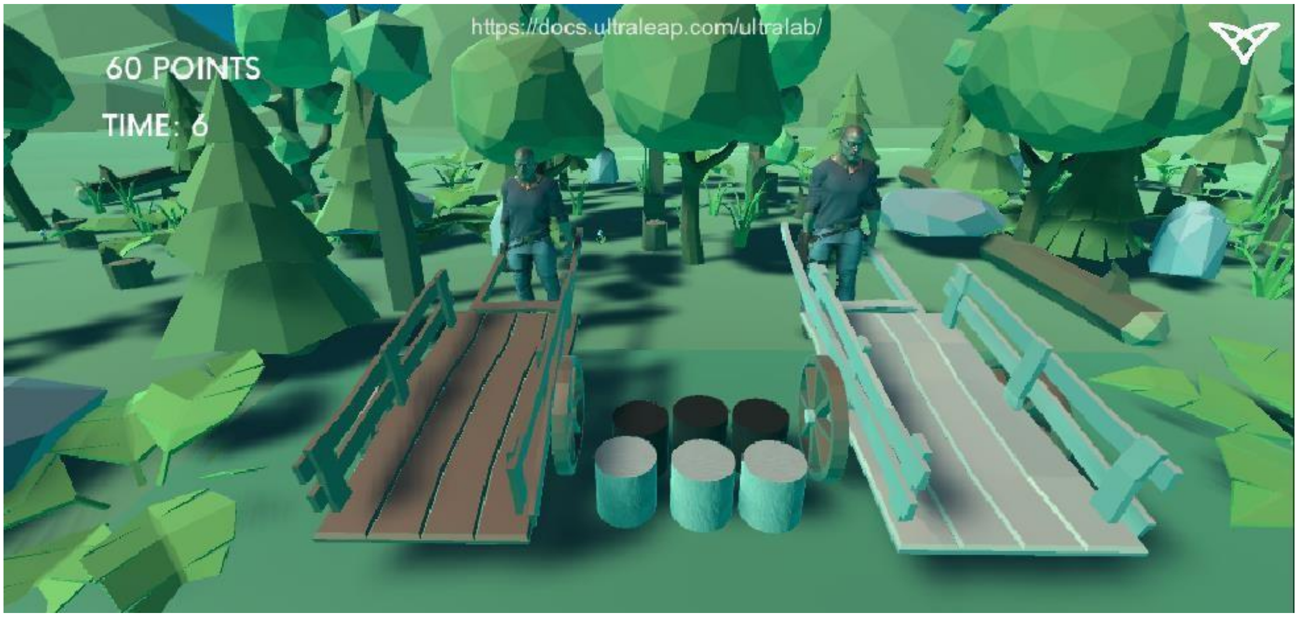


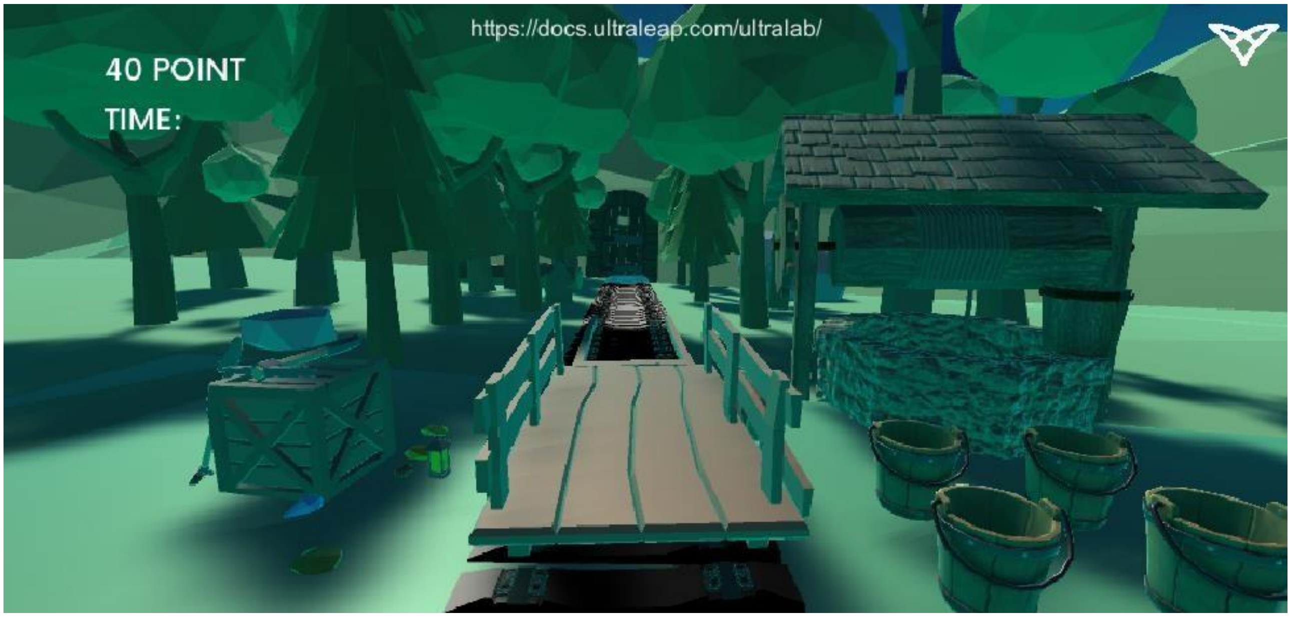
Use Case Name:	Part 3 (Level 5-6)	
Scenario:	Mine	
Triggering event:	The bomb ball reaches the end of the maze	
Brief description	Reach the end of the maze by collecting the gem in the mine with the bomb ball.	
Actors:	Player and System	
Preconditions:	The player has a motor disable	
Postconditions	System could be recording the score and time	
Flow of activities:	**Player**	**System**
	1. Drags the ball 2. Collects items with the ball 3. Finds the end of the maze 4. Restarts or continues the level	1.Starts the level and timer 2. Increases the items counter 3. Gives points for each collected item 4. Keeps time and score records

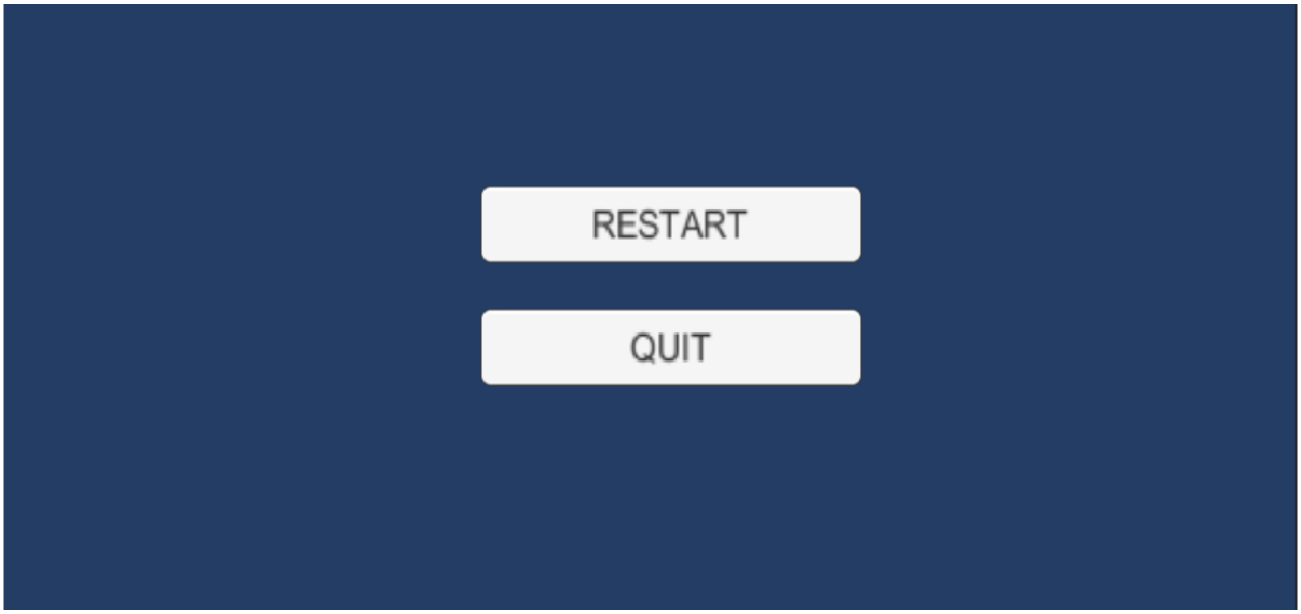
5 Design and Implementation



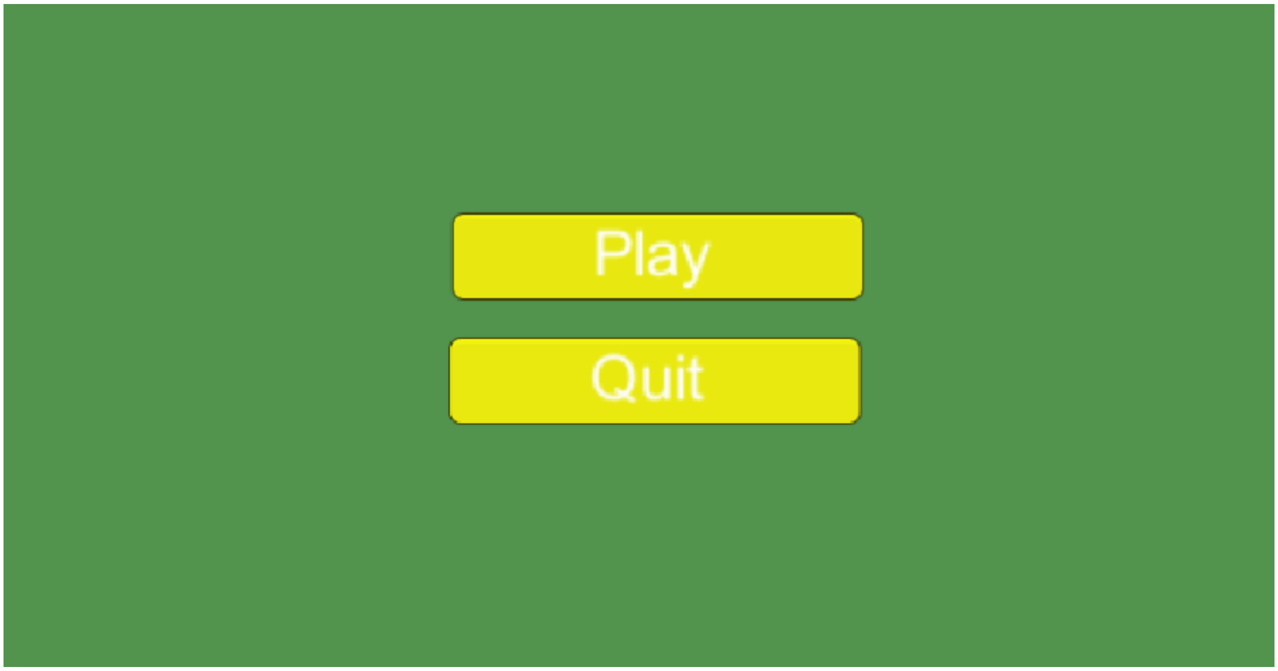


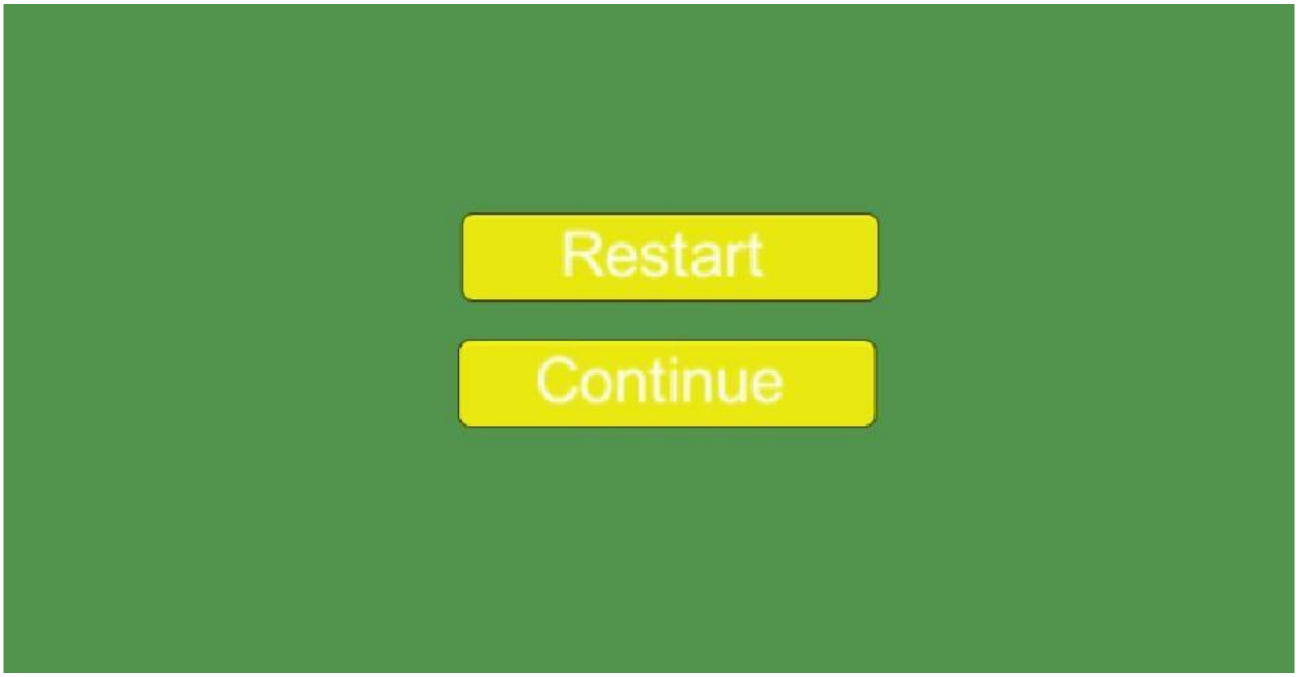

https://docs.ultraleap.com/ultralab/
60 POINTS
TIME: 6

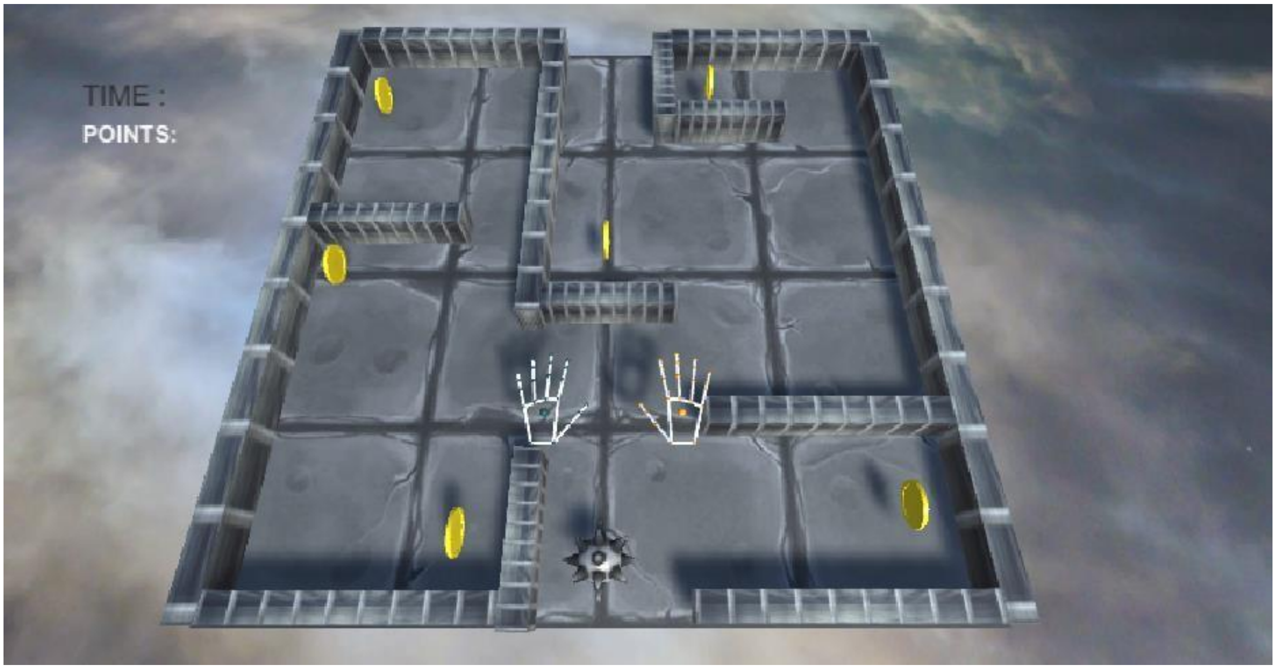

https://docs.ultraleap.com/ultralab/
40 POINT
TIME:

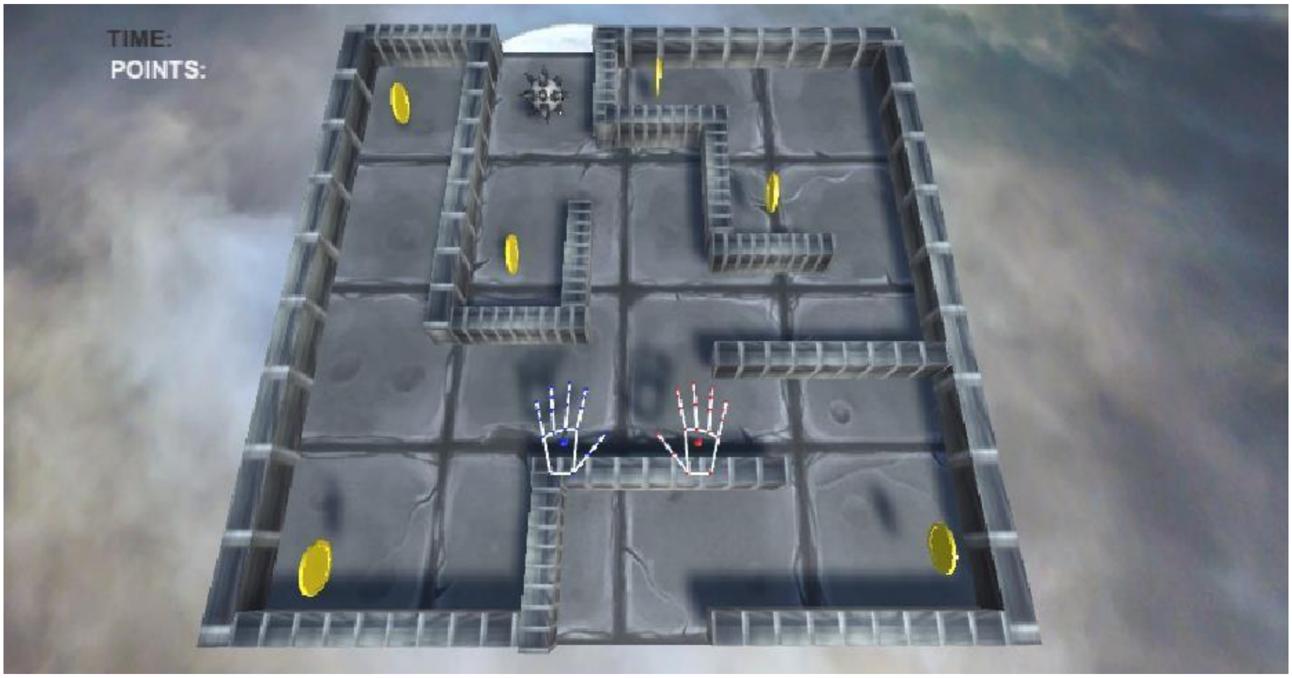


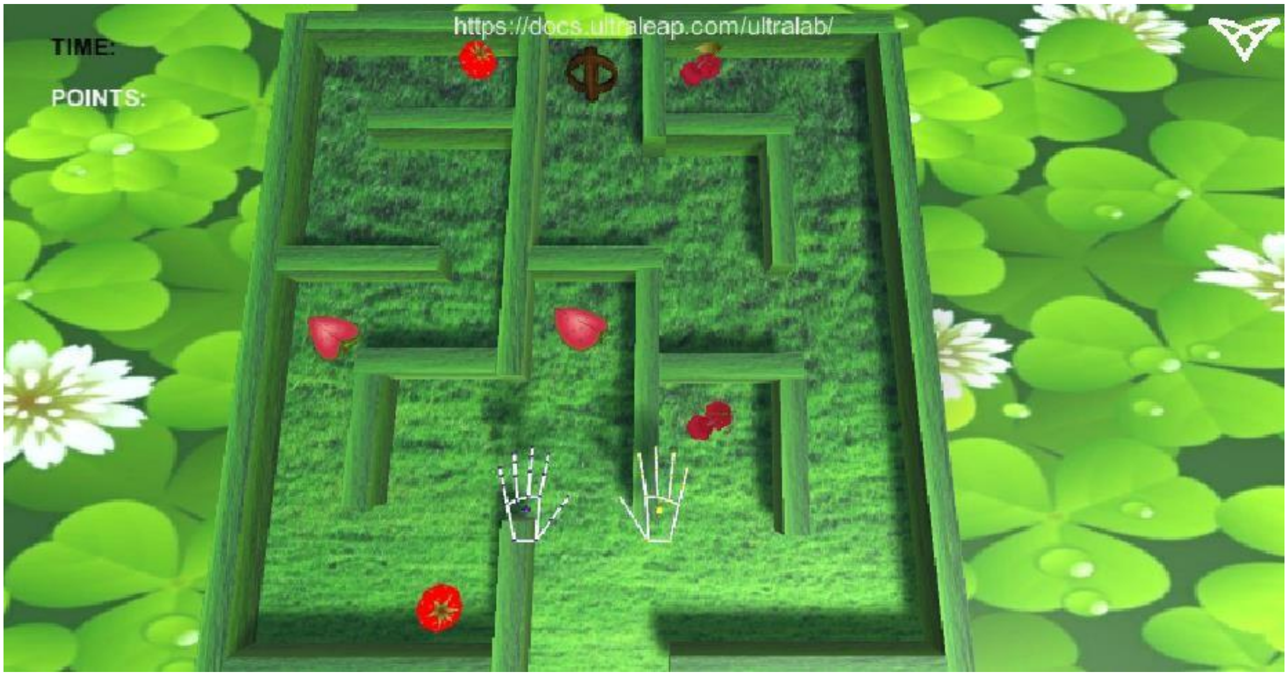
(CUBE GAME)

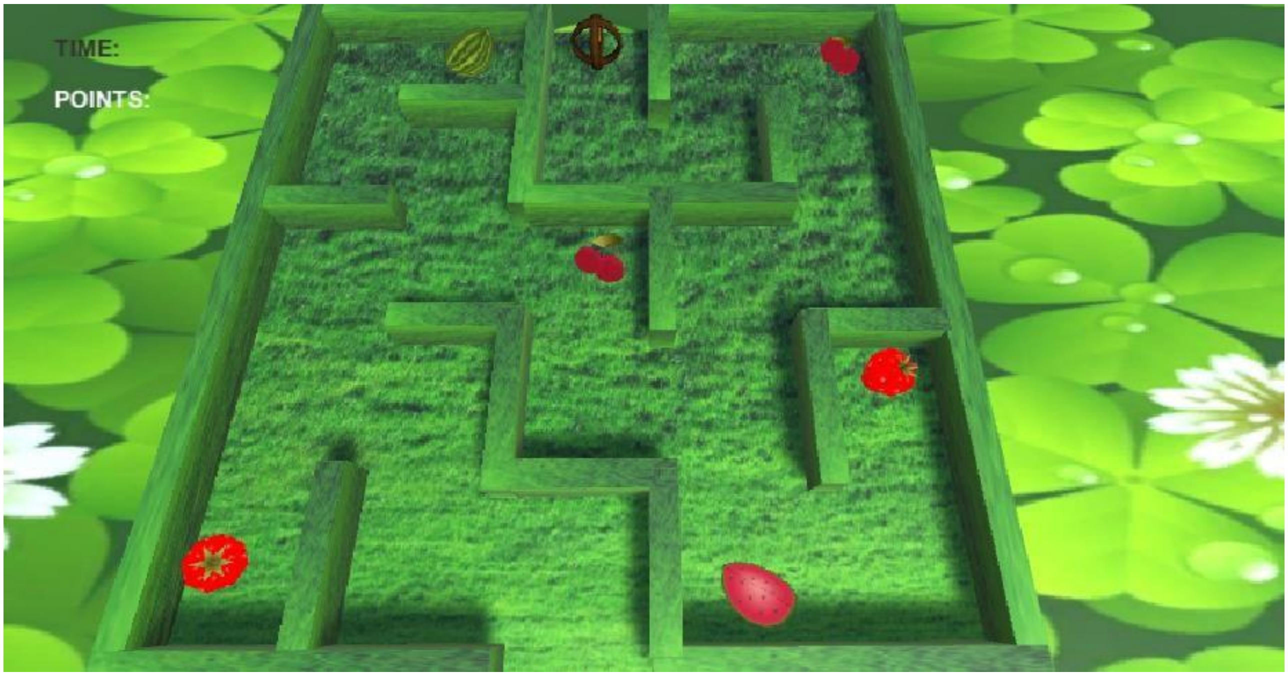


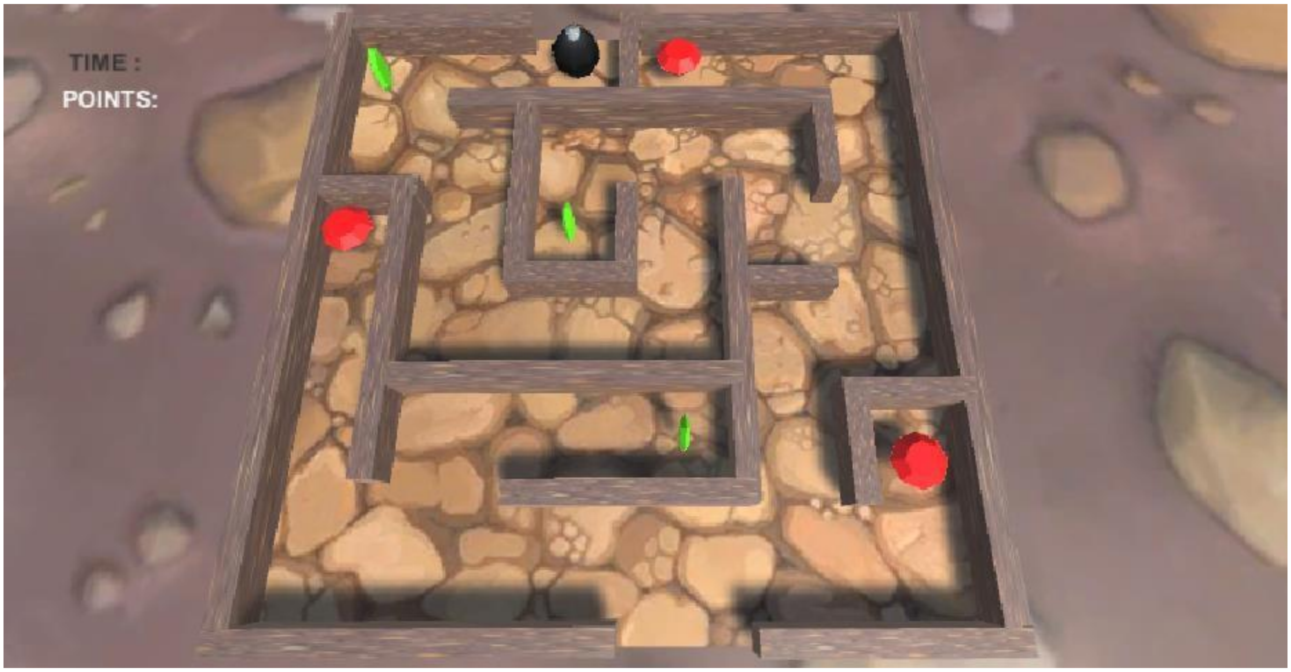

Restart
Continue

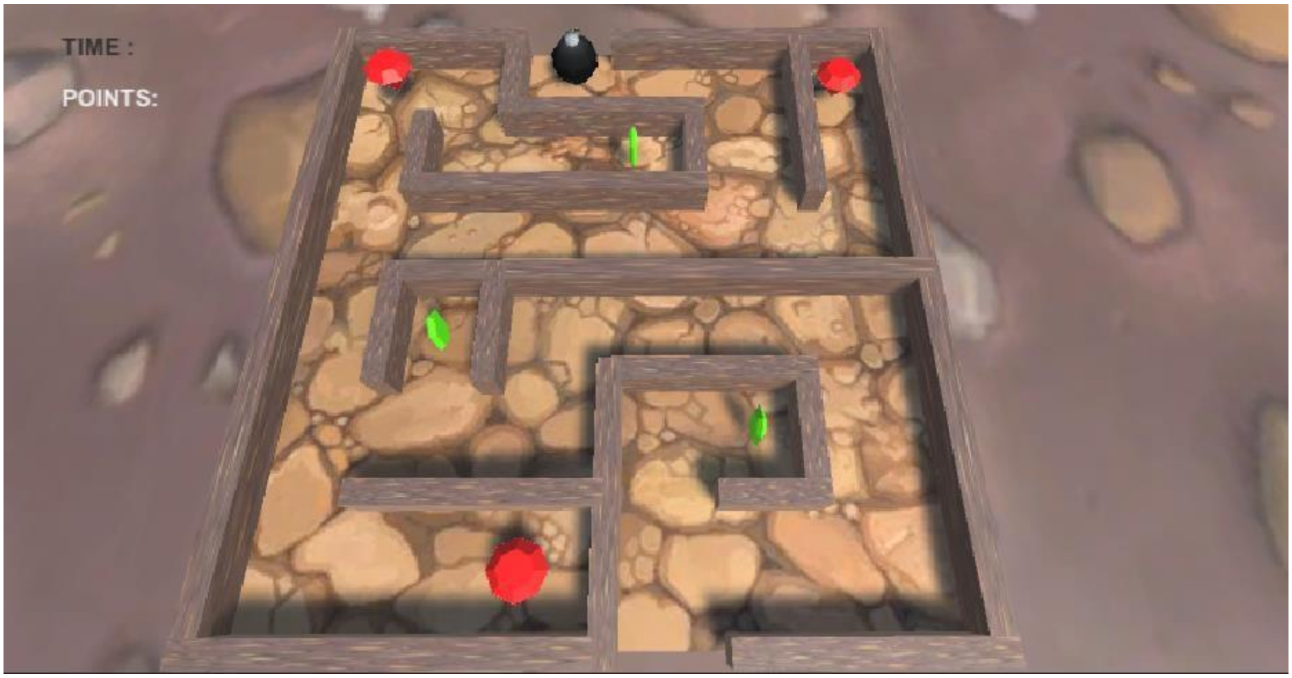

TIME :
POINTS:


TIME:
POINTS:


https://docs.ultraleap.com/ultralab/
TIME:
POINTS:


TIME:
POINTS:


TIME :
POINTS:



MAZE GAME

6 Other Supporting Information

https://gamedev.stackexchange.com/questions/150508/attach-component-of-other-object-to-a-variable-within-unity-inspector
https://answers.unity.com/questions/1430940/move-object-in-a-specific-angle.html
https://answers.unity.com/questions/1133794/how-to-see-if-two-objects-are-touching-each-other.html
https://answers.unity.com/questions/1189918/how-to-connect-two-c-scripts-and-connect-varibles.html
https://docs.unity3d.com/ScriptReference/GameObject-tag.html
https://assetstore.unity.com/packages/tools/input-management/the-essential-leap-motion-gesture-detection-111791

(CUBE GAME)
----

https://www.studytonight.com/3d-game-engineering-with-unity/maze-game

https://answers.unity.com/questions/225213/c-countdown-timer.html

https://answers.unity.com/questions/1038757/how-to-make-a-timer-that-counts-up-in-seconds-as-a.html

https://answers.unity.com/questions/372595/how-to-detect-collisions-in-c.html

https://forum.unity.com/threads/make-basic-script-to-collect-items.136955/

https://assetstore.unity.com/packages/audio/sound-fx/soundbits-free-sound-fx-collection-31837

MAZE GAME

7 References

https://assetstore.unity.com/packages/tools/physics/interactive-physical-door-pack-163249#publisher
https://assetstore.unity.com/packages/3d/characters/humanoids/humans/adventure-character-201384#description
https://assetstore.unity.com/packages/2d/textures-materials/sky/fantasy-skybox-free-18353#description
https://assetstore.unity.com/packages/3d/environments/landscapes/low-poly-simple-nature-pack-162153#description
https://assetstore.unity.com/packages/3d/environments/historic/colonial-ship-70472?aid=1011lGoK&pubref=Ship&utm_campaign=unity_affiliate&utm_medium=affiliate&utm_source=partnerize-linkmaker#description
https://assetstore.unity.com/packages/2d/textures-materials/water/stylize-water-texture-153577#publisher
https://assetstore.unity.com/packages/3d/props/tools/stylized-western-wooden-bucket-191513
https://assetstore.unity.com/packages/3d/props/wood-made-props-pack-1-29324#publisher
https://docs.unity3d.com/Manual/

(CUBE GAME)

https://assetstore.unity.com/packages/3d/props/simple-collectibles-pack-123092
https://assetstore.unity.com/packages/3d/props/ball-pack-446
https://assetstore.unity.com/packages/3d/props/food/low-poly-fruit-pickups-98135#description
https://assetstore.unity.com/packages/3d/props/simple-collectibles-pack-123092
https://assetstore.unity.com/packages/3d/props/ball-pack-446
https://assetstore.unity.com/packages/3d/props/food/low-poly-fruit-pickups-98135

MAZE GAME