Hide and Seek Game

*Final Report*

Crystal Monica Abou Jneid, Edwin Odeimi, and Mia Antoun.

**Abstract**—Hide and Seek Game is an android application used to help children who spend most of their time at home cope with their disconnection of the outside world. The following document describes in detail the features that the Hide and Seek Game presents. The document covers the user as well as the system requirements for each of the android and desktop application.

Adapted from the IEEE Computer Society template

# —————————— ◆ ——————————

# contents

[1 Introduction 2](#_Toc41054913)

[2 Background 3](#_Toc41054920)

[3 Proposal 4](#_Toc41054922)

[4 Implementation 11](#_Toc41054935)

[5 Performance Measures 16](#_Toc41054947)

[6 Conclusion 17](#_Toc41054949)

# Introduction

## **Objective**

The aim of the Hide and Seek game is to basically create an Android game whose main purpose is to help children who spend most of their time at home cope with their fear of separation with the outside world. The game will be specifically designed to be played by those children in order for them to cope with their fear of separation. However, it can also be played by any other person that may belong to a different age group. The game will mainly be used to improve the autonomy of those children, but it can also be used for entertainment purposes. The Hide and Seek game will be presented as an android mobile application. It will be accessible to any person who downloads the android application. It will first prompt the user to enter a username and then will ask him to choose between host or client mode, and according to the type of the choice the user makes a list of functionalities will become accessible to that particular user. The game will consist of hiders and seekers, who will be chosen randomly by the system. The seekers will then have to search for the hiders, and if the seekers find all the hiders, the game will end. Moreover, if after a time of fifteen minutes, the seekers were not able to find all the hiders, the game also ends. After that, a second round will begin, where the roles are interchanged, that is, the previously chosen seekers will become hiders and vice versa. The same process will then repeat, and after this process terminates the game would be considered to have ended.

## **Subject**

The solution we provide aims to facilitate the act of helping children who spend most of their time at home cope with their disconnection with the outside world. This can cause a problem in the long run since children are losing their habits of playing and missing the opportunity of having fun with other children of their age. If this trend is to go on, then, this might lead to serious psychological issues such as depression, loneliness, and more importantly, it can create in them fears of separability and some dependability issues. This document analyzes and elaborates on the high-level needs and features of that Hide and Seek Game that aims at improving these skills among children, in order for them to learn how to become more independent person with a stronger personality. The purpose of the requirements specification document is to provide a complete and comprehensive description of the Hide and Seek Game. This document defines and describes the interfaces, operations, performance, and functional requirements of the software. In addition, it also provides a description of the non-functional requirements and design constraints that are to be considered when the system is to be designed. The SRS also covers other technical concerns such as hardware, software, and communication dependencies. It also explains system constraints and interactions with other external applications. The requirements specification document is intended to be proposed to the system customers for checking that the requirements meet their needs and for updating these requirements if necessary.

## **Definitions, acronyms, and abbreviations**

## **Definitions:**

User Requirements: User requirements are statements, in a natural language plus diagrams, of what services the system is expected to provide to system users and the constraints under which it must operate

System Requirements: System requirements are more detailed descriptions of the software system’s functions, services, and operational constraints.

Functional Requirements: Functional requirements describe what the application should do, or what functionalities and services the application provides.

Non- Functional Requirements: Non-functional requirements are requirements that are not directly concerned with the specific services delivered by the application to its users. These are constraints on the services or functions offered by the application.

Context Model: The context model is a graphical representation (also known as architectural model) of the operational context of the system. It shows what lies inside and outside the system boundaries.

User Interface: The User Interface (sometimes referred to as GUI – Graphical User Interface) is the means by which the user interacts with the application and supplies his requests and receives the results.

Use-Case Model: The use-case model is a graphical representation of the interactions between the application and the user. It is similar to a scenario manager, whereby it lists the different scenarios of user-application interactions.

## **Acronyms:**

1- SRS – Software Requirements Specification

2- UML – Unified Modeling Language

3- TCP- Transmission Control Protocol

## **Overview**

The following sections will provide you with an overall description of Hide and Seek game along with its system requirements. The first part of the software requirements specification is the overall description which is divided into subsections that contain details such as the product perspective and user requirements. The system has two viewpoints which will be discussed further. Moreover, the second part describes the specific requirements that are directly related to the implementation level of the system. Both of the aforementioned parts include functional and non-functional requirements.

# Background

## **Context**

This section will give an overview of the whole system. The system will be explained in its context to show the interaction of system with other systems. It will also describe what type of users will use the system and what functionality is available for each type. At last, the general constraints and assumptions for the system will be presented. The Hide and Seek game will be used to compensate the lack of independency of children who spend most of their time at home and do not play outside with children of their age. The product we will be developing, which is the android mobile application, could be considered as an independent entity that is not related to a bigger system. On one hand, the android application will be specifically designed to be used by those children. It will provide them with capabilities to conquer their fear of autonomy and separation, providing them with joy in reunification and helping them practice and gain independence. On the other hand, the android mobile application will also be available for use by any other person that may belong to a different age group since this game will also be designed for entertainment. This application will need to communicate with a server program which will be used to return the IP address of the host.

# Proposal

The match can be at any time, in one of the ten states ordered from 0 till 9. Whenever a match starts, the Orchestrator starts at state 0 and then after doing the required steps in each state, moves to the next state until it reaches state 9, where it breaks from the state diagram.

* State 0 indicates that the match has not started yet. The Orchestrator goes through the list of guests and assigns each guest to be either a seeker or a hider, and that is done simply by generating a random number that can be either 0 or 1 and then checking for its parity.
* State 1 indicates that the Hiders are dropping on the map and in this state, a countdown is set for the hiders to hide. In fact, the countdown is set to be a subtraction of the maximum hiding time which is 30 time units by the hiding time that has already elapsed.
* State 2 indicates that the Seekers are dropping on the map and in this state, a countdown is set for the zone to start shrinking. In fact, the countdown is set to be a subtraction of the maximum seeking time which is 15 time units by the seeking time that has already elapsed.
* State 3 indicates that the zone is shrinking in size and a countdown is set for the match to end. It is important to note that the zone shrinks in terms of the match timeout which is equal to 30 and zone shrinking percentage which is equal to 0.8f (it is a multiplication of both). Therefore, every time unit the zone keeps on shrinking by the shrinking rate. The match countdown is then set. In fact, the countdown is set to be a subtraction of the maximum time the match is allocated which is 300 time units by the match time that has already elapsed.
* State 4 is used to reset everything for the second round including switching roles and displaying the results. The Orchestrator goes through the list of hiders and seekers and switches their roles respectively. The scoreboard countdown is then set. In fact, the countdown is set to be a subtraction of the maximum time the scoreboard is allocated which is 20 time units by the scoreboard time that has already elapsed.
* State 5 is used to indicate that the Hiders dropping on the map and a countdown is set for the hiders to hide. In fact, the countdown is set to be a subtraction of the maximum hiding time which is 30 time units by the hiding time that has already elapsed.
* State 6 indicates that the Seekers are dropping on the map and in this state, a countdown is set for the zone to start shrinking. In fact, the countdown is set to be a subtraction of the maximum seeking time which is 15 time units by the seeking time that has already elapsed.
* State 7 indicates that the zone is shrinking in size and a countdown is set for the match to end. It is important to note that the zone shrinks in terms of the match timeout which is equal to 30 and zone shrinking percentage which is equal to 0.8f (it is a multiplication of both). Therefore, every time unit the zone keeps on shrinking by the shrinking rate. The match countdown is then set. In fact, the countdown is set to be a subtraction of the maximum time the match is allocated which is 300 time units by the match time that has already elapsed.
* State 8 is entered when the match has ended and it displays the results. The scoreboard countdown is then set. In fact, the countdown is set to be a subtraction of the maximum time the scoreboard is allocated which is 20 time units by the scoreboard time that has already elapsed.
* State 9 is used to return to the lobby.
* As soon as the game starts, a counter is used to keep track of the number of hiders that were caught.
* As long as the game did not end and no hider is caught, the match elapsed time will not be decremented and will still be equal to its maximum value which is that of the timeout.
* In the case where the match did not end yet but no guest was found in the game, then the game is immediately stopped.
* As soon as the game starts, a specific message is broadcast depending on the state of the game. Therefore, when in states 1 and 5, it displays the hiding elapsed time, when in states 2 and 6, it displays the seeking elapsed time, and when in states 3 and 7, it displays the match elapsed time, and when in states 4 and 8, it displays the score elapsed time, and when in state 9, it displays stop.
* For every hider that was not caught, the hiding score gets incremented.
* When the host presses the start button, the scores for the hiders and seekers respectively are set to 0, and the hiders and seekers are chosen randomly in such a way that we have equal number of hiders and seekers.
* Whenever a guest sends a message, that message gets processed and in case a hider was caught, the seeker’s score gets incremented by its win value which is equal to 100. In addition, the hider that was caught gets removed from the round.
* Each player in the game is assigned some network data such as his position and movements.
* In some cases, a guest might be attempting to enter a game but he may not be admitted in the following two cases: either the maximum number of players allowed in the game was reached or the guest required to enter when the game has already started. When a new guest connects to the game, the connection occurs using TCP and the client receives some attributes. When the lobby starts, it starts on port 6321 using any IP address.

## **User Requirements**

To better understand the user requirements, we will be dividing them into two sub-categories, Functional User Requirements and Non-Functional User requirements.

## **Functional Requirements**

The functional requirements of the system are that of each type of user (client user, and host user). The following subsections will state these functional requirements followed by a description of the requirement (if necessary)

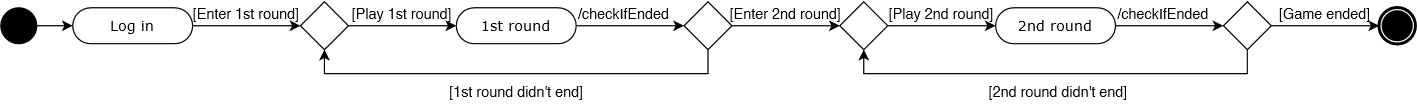


Figure -Login

User sign in:

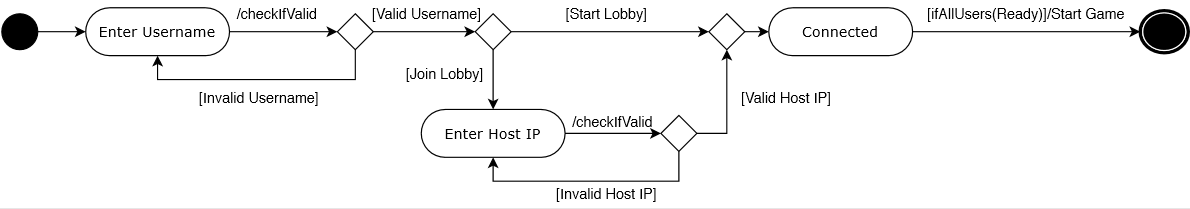


Figure -Sign in

As soon as the user opens the Hide and Seek Game, he will be prompted to enter a username.

The username must:

1. Start with either an underscore character or a letter.
2. The remainder letters must be either a number, an underscore character or a letter.
3. The username can neither be empty nor it can contain empty space characters.
4. The username must consist of at least five characters.

The user will then press the “Continue” button. Any of these conditions that is not met will generate an error message, asking the user to reenter a valid username.

|  |  |
| --- | --- |
| Condition | Action |
| Username contains less than three characters | ERROR: Check your input! |
| Username contains empty space characters | ERROR: Check your input! |
| Username does not start with a letter | ERROR: Check your input! |
| Any of the remaining letters is not a number, an underscore character or a letter | ERROR: Check your input! |

Table -Condition/Action table

After the username was successfully typed, the user’s status will be set to “Connecting”, and another scene will appear to the screen containing two buttons:

1. The “start lobby” button.
2. The “join lobby” button.

## **Host viewpoint:**

If the user presses the “start lobby” button, he will become a host. When a new host is created, a new server socket starts. In addition, the host will have an IP address specific to his own device, which will be used by the remaining players that will join the game in order to establish a TCP connection with the server.

The host is immediately directed to another scene which is referred to as the “Lobby” scene, in which he will be able to view all the players that are “Connected” to the game.

Once all players that are “Connected” are “Ready”, the host can now press the “Start” button.

In case no player was “Ready”, the game will not start, and the host will have to wait for at least one user form the opposite team to be “Ready” in order to push the “Start” button again.

N.B:

The host sends the IP address of the first user that was “Connected” to the game, to all remaining “Connected” users to the game from to the time at which all users were “Connected” until the middle of the game.

In case the host leaves the game early:

1. The first player that was “Connected” to the game will become the host.
2. The other users will attempt to connect to the newly designated host.
3. Each user will be given ten attempts in order to try to establish a connection with the host.
4. If after ten attempts the user fails to connect to the host, he will be redirected to the second scene, in which he has to choose between the “start lobby” button and the “join lobby” button.

## **User viewpoint:**

If the user presses the “join lobby” button, he will receive an incomplete IP address, with a textbox replacing the rightmost three digits. The user will then be prompted to fill this textbox.

The textbox must:

1. Contain only digits.
2. Contain at maximum three values.
3. Correspond to the rightmost three digits of the host.

After filling the textbox, the user will press the “Continue” button. Any of the above conditions that is not met will generate an error message, asking the user to reenter a valid number.

|  |  |
| --- | --- |
| Condition | Action |
| Textbox contains characters that are not digits | ERROR: Invalid IP address. |
| Textbox does not correspond to the rightmost three digits of the host | ERROR: There was a networking error! |

Table -Condition/Action table

If the textbox was correctly entered, the user’s status will then become “Connected”, otherwise it will remain “Connecting”.

When a user’s status become “Connected”, another scene will appear on the screen, which is called the “Lobby” scene. This scene shows the host as well as all the users that are “Connected” to the game.

The user will now be able to press the “Ready” button, meaning that the user is ready to start the game.

## **Player viewpoint:**

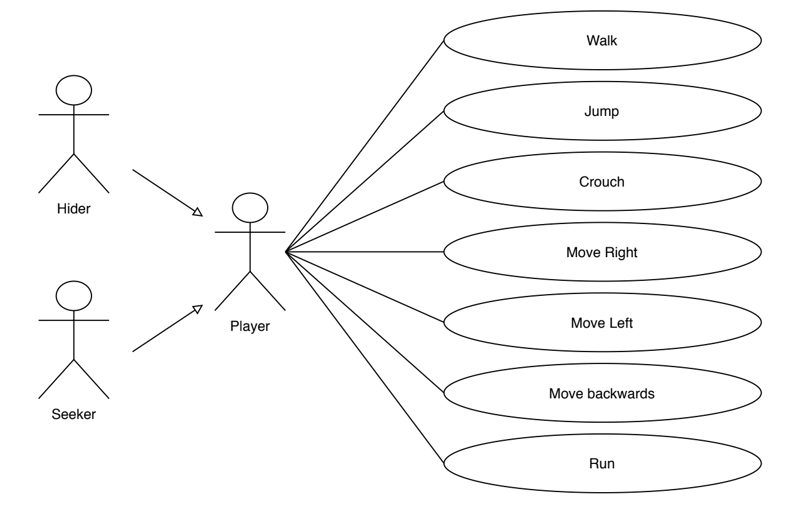


Figure -Player viewpoint

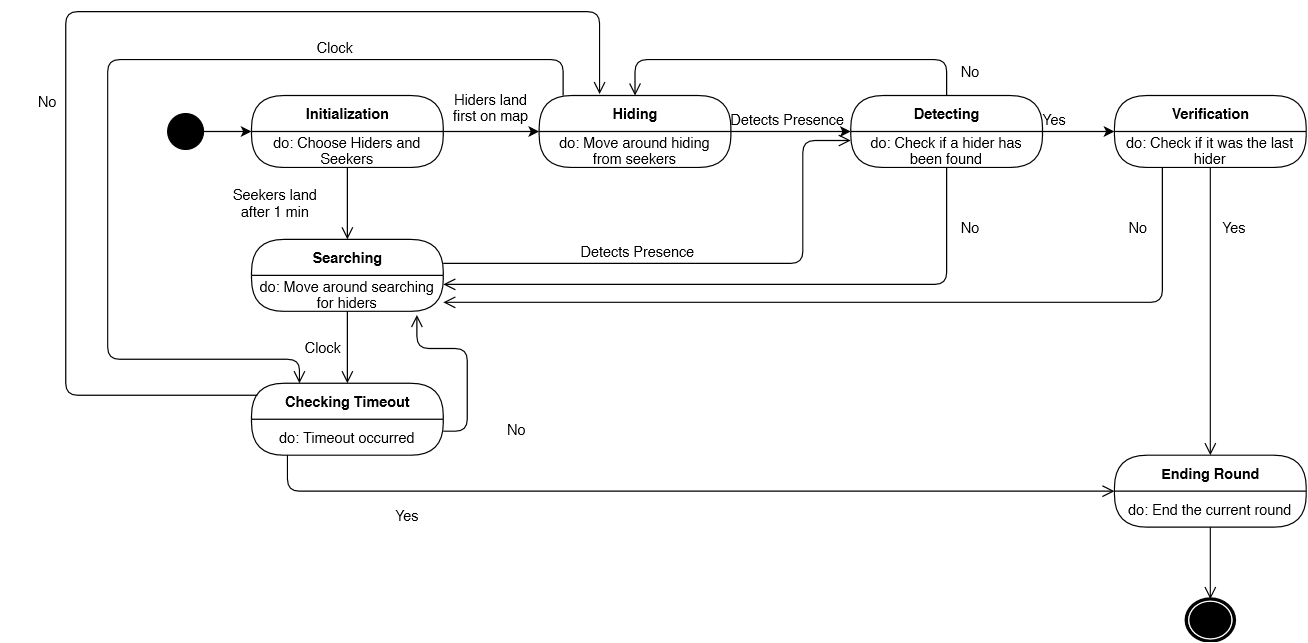


Figure -Game Description

|  |  |
| --- | --- |
| State | Description |
| Initialization | The hiders and seekers are chosen randomly such as half of the players are hiders and the other half are seekers. |
| Hiding | The hiders move around hiding from the seekers |
| Searching | The seekers move around trying to find the hiders. |
| Detecting | Checking if a hider has been detected. |
| Checking Timeout | Checks if a timeout occurred. |
| Verification | Checks if the hider that was found was the last hider. |
| Ending Round | The current round terminates. |

Table -State/Description Table

1. As soon as the host presses the “Start” button, another scene appears on the screen, which is the Game scene.
2. The “Hiders” and “Seekers” are chosen randomly such that half of the players are “Hiders” and the other half are “Seekers”.
3. All players are skydiving from a high - platform in the sky, and the map is present beneath the sky.
4. Hiders land on the map before the Seekers, and each one chooses the location on the map in which he would like to hide.
5. After one minute has elapsed, the Seekers land on the map.
6. Initially, each player is in an idle position.
7. The player can move in any direction, by using a joystick which will appear whenever the user presses on the mid-left side of the screen.
8. The player can jump, crouch whenever the user presses on the corresponding button which appear on the right side of the screen.
9. If the player pressed on the mid-right side on the screen and swiped, the character will be rotating with the camera, which is called panning.
10. The map contains a circle zone such that the center of this circle is chosen randomly.
11. This circle shrinks every other minute in size.
12. When the Hiders are outside the circle, a storm appears to them. However, this storm does not appear to the Seekers.
13. Because of this storm, the Hiders’ visibility will be compromised. It will become harder for them to hear the steps of the Seekers because of the loud storm. Moreover, the Hiders’ steps are louder outside of the circle. All these factors will make it harder for them to survive by making them easily detectable.
14. It is important to note that the Hiders cannot stay at the same location, and therefore must move on the map, which will make it inevitable for them to be under the storm at some point in time.
15. When all the Hiders are found, the first round ends.
16. If all Hiders were not found after fifteen minutes, that is there may be at least one Hider that is hiding, then the game also ends.
17. This is considered the half-time of the game. At this point, the Hiders’ and Seekers’ roles are interchanged and the same process repeats.
18. When the next half terminated, the game is considered to have ended.
19. At the end of each round, a ranking system appears for the Hiders in such a way that the Hider that was caught the first is ranked the last, and the one that was found the last is ranked the first.
20. Moreover, if some Hiders remained unfound until the end, then these are ranked randomly starting from the first position.
21. A Seeker’s score is represented by the number of Hiders he has found.
22. The ranking system appears as a table showing the Hiders’ ranks on the right, and the Seeker’s ranks on the left.
23. If the game is terminated successfully and the player is still “Connected”, then he will be redirected to the “Lobby” scene.
24. If a player gets disconnected because he was not able to connect to the newly set host, he will be redirected to the “Join and Start” scene, which is the second scene.
25. Moreover, there is a small map that appears on the top of the screen. When the user presses on this map, it maximizes on the screen. This map is used to locate the player on the map. The player can then close this map by pressing the close button, which will minimize it again.

## **Non-Functional Requirements**

## **Efficiency:**

The system should have low latency and low power consumption.

## **Maintainability:**

The application shall be easily extended. The code should be written in a way that new functions can be easily added afterwards.

## **Usability:**

The system shall be easy to use by a range of users with different levels of experience with Android smartphones and should be organized in such a way that user errors are minimized.

## **Size and Memory Management:**

The hard drive space required by the application shall not exceed 43 megabytes. Moreover, the code should be optimized in a way to minimize the amount of OS memory occupied by the application to less than 150 megabytes.

## **Security:**

The log-in communication messages shall be encrypted, so others cannot obtain the username from these messages. The system shall keep logs and history data sets of IP addresses and associated username logins. Also, an IP address shall be blocked if more than 10 consecutive unsuccessful login attempts have been made from it.

## **System Architecture**

Below is a context model of the system architecture:

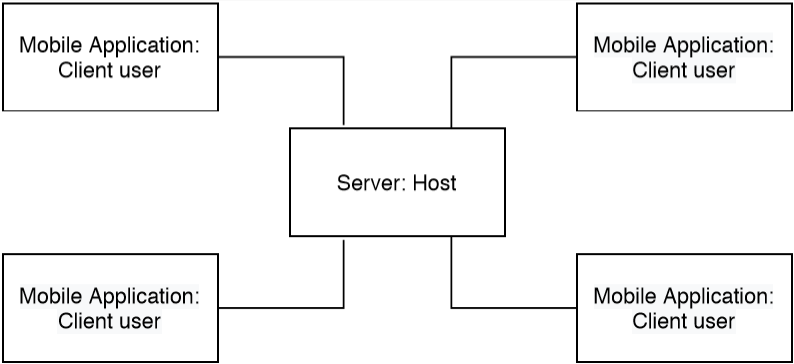


Figure -Context model

The Hide and Seek game will be used to compensate the lack of independency of children who spend most of their time at home and do not play outside with children of their age. The product we will be developing, which is the android mobile application, could be considered as an independent entity that is not related to a bigger system. On one hand, the android application will be specifically designed to be used by those children. It will provide them with capabilities to conquer their fear of autonomy and separation, providing them with joy in reunification and helping them practice and gain independence. On the other hand, the android mobile application will also be available for use by any other person that may belong to a different age group since this game will also be designed for entertainment. This application will need to communicate with a server program which will be used to return the IP address of the host. The Hide and Seek game can be accessed through a client machine such as an Android (or an emulator) device running Android 4.1 ‘Jelly Bean’ (API 16) or above. In order for the application to run, the Android (or emulator) device should have enough storage to hold and run the application. The system will use TCP protocols to enable communication between the client and the server.

# Implementation

## **User Interfaces**

The android mobile application shall provide a very simple and user-friendly interface. A login page shall appear first where the application user will be prompted to enter a username. After entering a valid username, the mobile application user will be redirected to a page that will ask him to choose between two options: start lobby or join lobby. In case the user pressed the start lobby button, he will be considered the host of the game and will be redirected to a page where he is able to see the currently logged- in users in the game. In the case where the user has pressed the join lobby option, he will be redirected to a page in which he will be asked to fill a textbox with the values that correspond to the host IP address. After entering a valid IP address, the mobile application user will be redirected to the page that contains the list of the currently connected users. The user will then be given the option of pressing the ready button. When all players are ready, the host shall start the game. It is important to note that there must be at least one ready user other than the host in order for him to be able to start a game. Moreover, the host sends the IP address of the first user that was “Connected” to the game, to all remaining “Connected” users to the game from the time at which all users were “Connected” until the middle of the game. Therefore, if the host leaves the game early, the first player that was “Connected” to the game will become the host, and the remaining users will attempt to connect to the newly designated host. Each user will be given ten attempts in order to try to establish a connection with the host. If after ten attempts the user fails to connect to the host, he will be redirected to the second scene, in which he has to choose between the “start lobby” button and the “join lobby” button.

## **Login phase:**

The first user is prompted to enter a username:



Figure -Insert username

The second user is also prompted to enter a username:

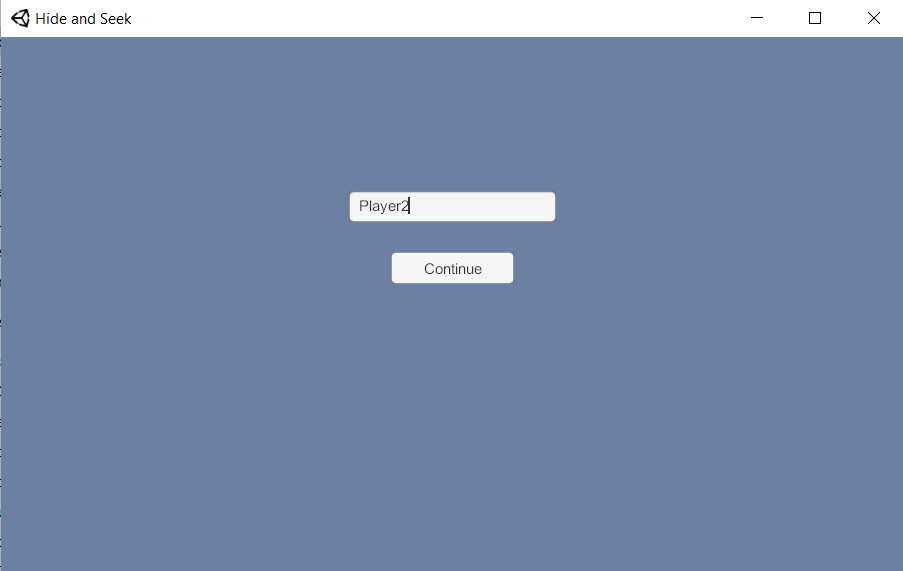


Figure -Insert Username

## **Start/Join Lobby phase:**

This is what appears next on the screens of the users Player 1 and Player 2 respectively:



Figure -Start/Join Lobby

Now the users can choose to either start a lobby or join a lobby.

We will consider the case where Player 1 is the Host, therefore Player 1 will press on Start Lobby while Player 2 will press on Join Lobby.

## **Ready phase:**

When Player 1 (the Host) presses on Start Lobby, this is what appears on his screen:



Figure -Ready Phase

When Player 2 presses on Join Lobby, this is what appears on Player 1’s screen.

As we can see, Player 1 can now see all the players which are ready, which is Player 1 and Player 2 in this case.



Figure -Ready phase

## **Game phase:**

## **Hider viewpoint:**

When Player 1 (the Host) presses on Start Match the Hiders start skydiving and a countdown is set for the Seekers so that when it reaches 0, it will become the Seekers’ turn to skydive.



Figure -Hider viewpoint

## **Seeker viewpoint:**

And this is what appears on Player 2’s screen: As we can see, this is the countdown that appears on the Seekers’ screen. Therefore, as we mentioned previously, by the time when the Hiders would have reached the ground, the Seekers would start to skydive. In this example, we can see that there is still seven seconds before the Seekers can actually start to skydive.

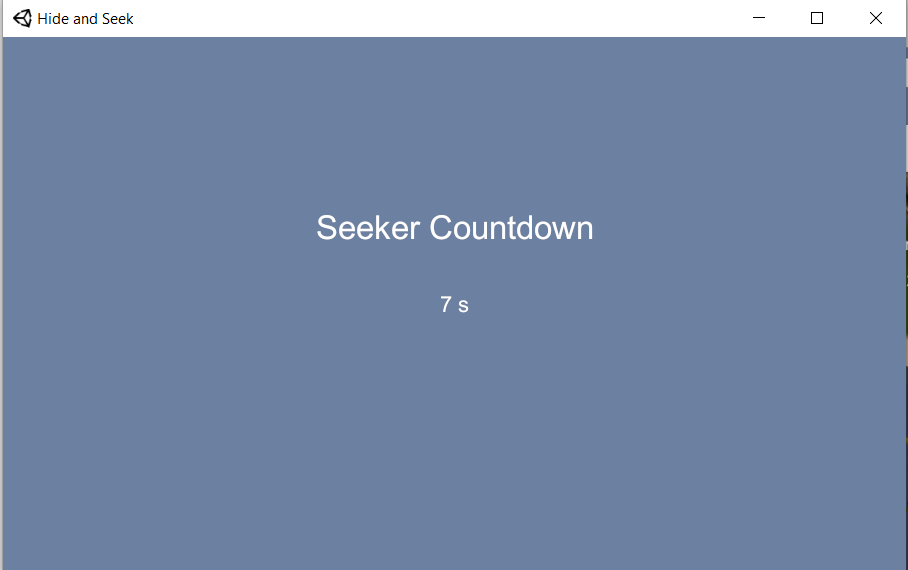


Figure -Seeker viewpoint

## **Displaying countdown:**

In the Below picture, we can see how to countdown start to decrement as soon as all the players are on the map. This countdown reaches zero when the zone size would become zero. In fact, the zone keeps on shrinking by a certain factor, and by the time it reaches zero, the game will be stopped, and the next round begins after displaying the scores.



Figure -Match Countdown

## **Zone shrinking:**

In the picture below, where we are displaying the Seeker’s viewpoint we are showing how the visibility becomes poor outside of the zone region.

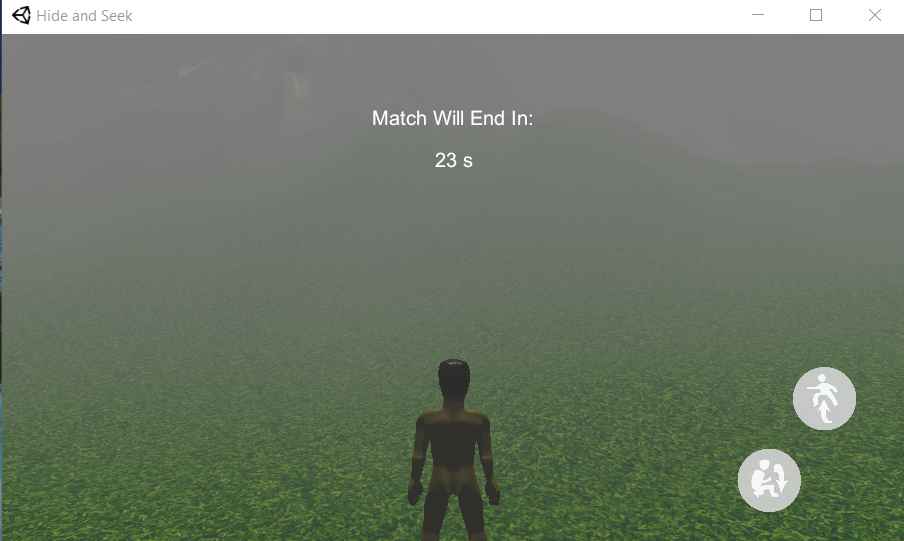


Figure -Zone Shrinking

## **Displaying results:**



Figure -Displaying Results

## **Hardware Interfaces**

The Hide and Seek game can be accessed through a client machine such as an Android (or an emulator) device running Android 4.1 ‘Jelly Bean’ (API 16) or above. In order for the application to run, the Android (or emulator) device should have enough storage to hold and run the application.

# Performance Measures

## **Statistics**

Before the round begins After the round ends

The above graph represents the case where the game would provide us with the worst performance.

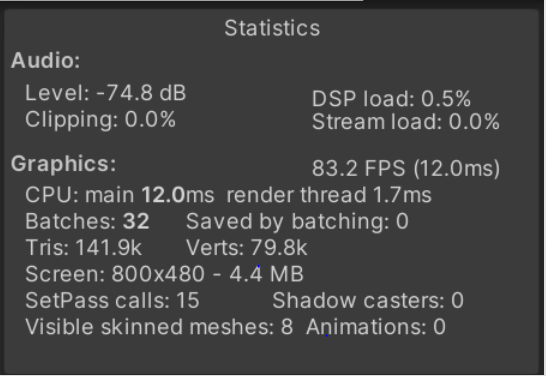


Figure -Statistics

* The Update function allows us to monitor inputs and other events regularly from a script and take appropriate action. For example, when we desire to move a character when the “forward” key is pressed. An important thing to remember when handling time-based actions like this is that the game’s framerate is not constant and neither is the length of time between Update function calls. Therefore, in our case, we chose the FPS unit (Frame per second) to be equal to 83.2. We should note that the higher the number of frames per second, the faster will be the movements of the players.

# Conclusion

The aim of this project was to create an Android game whose main purpose is to help children who spend most of their time at home cope with their fear of separation with the outside world. The game was specifically designed to be played by those children in order for them to cope with their fear of separation. However, it could also be played by any other person that may belong to a different age group. The game will mainly be used to improve the autonomy of those children, but it can also be used for entertainment purposes. The Hide and Seek game will be presented as an android mobile application. It will be accessible to any person who downloads the android application. It will first prompt the user to enter a username and then will ask him to choose between host or client mode, and according to the type of the choice the user makes a list of functionalities will become accessible to that particular user. The game will consist of hiders and seekers, who will be chosen randomly by the system. The seekers will then have to search for the hiders, and if the seekers find all the hiders, the game will end. Moreover, if after a time of fifteen minutes, the seekers were not able to find all the hiders, the game also ends. After that, a second round will begin, where the roles are interchanged, that is, the previously chosen seekers will become hiders and vice versa. The same process will then repeat, and after this process terminates the game would be considered to have ended.