XboGame Spaco TV-Centric

Ali Ghazi Abu Saleh Abdallah Wael Mustafa

May 29, 2020

Supervised by Dr.Samer Arandi

An-Najah National University
Faculty of Engineering and Information Technology
Computer Engineering Department

Submitted May,2020 , in partial fulfillment of
the the requirements for Bachelor degree in **BSc Computer Engineering**

Contents

List of Figures

Acknowledgement

We are overwhelmed in all humbleness and gratefulness to acknowledge our depth to all those who have helped us to put these ideas, well above the level of simplicity and into something concrete.

We would like to express our special thanks of gratitude to our supervisor **Dr.Samer Arandi** who gave us the golden opportunity to do this wonderful project on this project , which also helped us in doing a lot of Research and we came to know about so many new things. we are really thankful to him.

we would like to thank **our parents** who helped us a lot in gathering different information, collecting data and guiding us from time to time in making this project , despite of their busy schedules ,they gave us different ideas in making this project unique.

Last but not least, we are thankful and fortunate enough to get encouragement and support from all teaching staff of Computer Engineer Department.

Disclaimer

This report was written by Ali Abu Saleh and Abdallah Mostafa at the Computer Engineering Department, Faculty of Engineering, An-Najah National University. It has not been altered or corrected, other than editorial corrections, as a result of assessment and it may contain language as well as content errors. The views expressed in it together with any outcomes and recommendations are solely those of the students. An-Najah National University accepts no responsibility or liability for the consequences of this report being used for a purpose other than the purpose for which it was commissioned.

Abstract

The gaming stations are improving around the world and it is moving to the stage are required to buy strong devices to play smoothly, on the other side the televisions is also improving and nowadays it supports modern operating systems, but still used as normal television but with connection to the Internet. One of the well-known systems is the Android Operating system.

Our goal is to use the android television in to play games., and it is an end-to-end tools to create educational games that can run on android TV and any android device as a controller. These games which are built using this system is based on interactive learning games for children under the supervision of parents and built on Unity engine-based and install it on the Android TV Box, and a controller built on Android devices used as a controller, the communication between the games and the controller based on Network service discovery (NSD), and working on a local network without getting access to the internet. Moreover, this system is loading the games either dynamically or statically, depends on the Android TV Box, a new Java application game wizard is built to test the interactive system and it requires free-code which allows programmers and non-programmers to build a new game on this system. The report discusses the benefits of the TV gaming concept, presents the communication over the local network using the Network service discovery (NSD) and the robust communication between the TV game and the android controller.The programming tools and platforms that was used to implement the project are: Java for Android TV Controller,JSON for output files of the wizard and Java for the wizard game engine.In addition to Unity (C) for implementing the games.

1 Introduction

There are many trends for video games such as educational, intelligence, entertaining, and strategy games, the focus will be in educational video games.

The Video Educational Game has significant benefit in spelling and decoding area (9) , also this type of Video games encourage children to learn ,it can be a good tool for teaching in the classroom in the future (10).

XboGame Spaco is completely educational video game , which is designed for preschooler kids , this game is similar in principle the PlayStation games , and consist of controller such as Tablet or android phone to control the game on android TV .

The game is to three mods to motivate kids to learn , these mods is Learn , Guess , and match , every mode of these is Focuses on developing a player's specific ability , such as intelligence .

In Learn mode , list of objects will be shown in controller ,when player click on it ,it will shown in TV with option of sound , spelling , extra information about this object .

Guess mode is different from learn mode , random object will shown in TV , and the player should know the name of the object , the Guess Mode is similar to Guess Mode but with additional two object .

To make more challenge in the game designed to support multiplayer , which two player can Make Guess and Match challenge together , also to make game more dynamic to change the content a wizard is designed to realizes this feature .

Note that the picture of the game in this report is example of group of objects that related to sea animals .
This report Contains Constraints & Earlier course work chapter , Literature Review chapter , Methodology chapter, Results and Discussion chapter , and Conclusion & Discussion chapter .

2 Constraints, Standards/ Codes and Earlier course work

This Section discuss the Constraints, Earlier course work .

2.1 Constraints

- **Time to complete the project** : the given time to fulfill the project is not enough because reviewing the basic courses and study external courses is approximately take quarter of the time given.
- **Android Documentation**:the backbone of the project is Network Service Discovery(NSD) , there is no example of this property ,also the documentation not clear .
- **Unity not Free** :there are many of assets are not free such as load 3D object at Run time , Stylish User Interface (UI) elements .
- **Design** : We don't have that knowledge in designing and UI/UX so it will be better if we know how to build a better desgin.

2.2 Earlier course work

The essential courses to fulfill the project is Computer programming , Object Oriented programming ,and Computer Networks 1 .

Another external courses and tutorials have been studied such as basic C# programming language , network service discovery (NSD),LitJson Library for JSON Files ,and Java for Android .

3 Literature Review

The new powerful TVs are playing important role in social media and gaming and while these games is used to be controlled over the remote control, and they are a simple games such as 'Wize' game which allows the users to playing a quiz game while watching a show (1).In the other hand, the smart mobile devices is the new trend of gaming which has more features than the PCs or laptops, the TV is having a touch screen, motion sensors, WIFI connection and many others features that allows the users to be more attractive to the game as it's used to communicate and play with friends, these features has an advantage for the users but at the same time it's a challenges for game developers as the input is changing from normal keypad to touch screen, microphone and many others (2).The TV-centric concept is the coming generation of gaming, because it gives the users the most experience and make the user be part of the game not only interact with it using his smart phone(3).

Games is not only for adults, it's being used by the children also, the games is also used to assist children with special needs like Dyslexia and Attention Deficit Disorders(4). The educational games have to be attractive to attract the children to use it, but not to be addictive one to not affect the children(5).Well, the TV-Centic games until now is built by the game developers and it either uses the internet access to interact or using a small range control like WIFI direct or Bluetooth, in this project the system built for children using the local network using Network Service Discovery (NSD) without internet and under the supervision of parent, it provides multiplayer modes in case of the child is needs helps or he/she has special needs and building a TV-Centric Games without needs programming skills.

4 Methodology

The main point of our game is to provide a educational game for kids using TV-centric approach, and to build a game where the parents are included in the game with their kids.

In-order to clarify how our project is working and how the communication between the TV-Centric game and the Android controller, the detail will be discussed in this section.

4.1 Network Service Discovery(NSD)

We have mentioned Network Service Discovery (NSD) many time, and this is the core of our project since it's handle the registration for the server, handle the atomic discovery and the first message sent from the client to the server, the following figure (Fig. No. 1) shows the basic principles of how service discovery woks

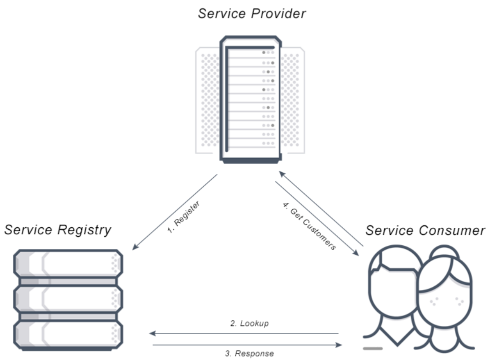

Figure 1: Network Service Discovery (NSD) principles

so the NSD lets your app find other devices on the local network that's support this type of discovery, that's includes Printers, webcams, mobile devices and HTTP servers to make it faster than finding using the internet or even using Bluetooth or WiFi direct, and it turns the local network into social network.

NSD is based on two protocols, DNS-SD and Multi-cast DNS

4.1.1 Multi-cast DNS (M-DNS)

This type of DNS is different from the known DNS, it's completely disrupted so there's no central server, and to use M-DNS it has requirements (6)

- Devices must be on the same broadcast doming (and the same subnet mask).
- Network must support multicast traffic

4.1.2 DNS Service Discovery (DNS-SD)

so the idea is to register the services on the local network and broadcasting it to the, nearby peers(7), by using the same idea of DNS you register on the local area, so in the following figure (DNS Service Discovery (DNS-SD) example) the client is asking about a music in the network and getting the response from the multi-sharing device instead of getting it from the internet

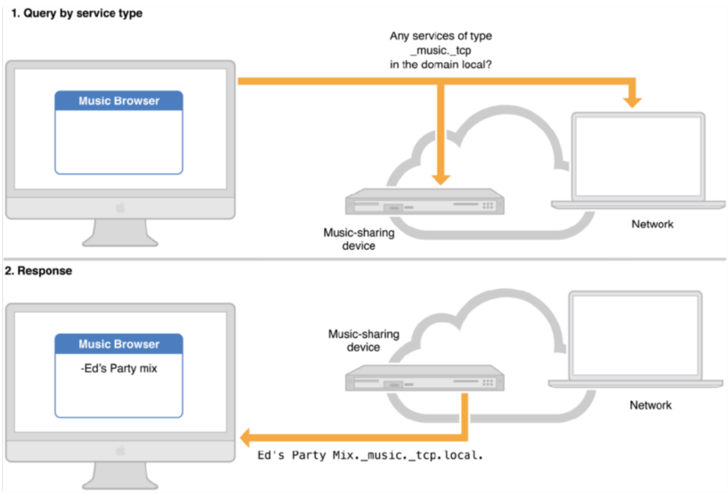

Figure 2: DNS Service Discovery (DNS-SD) example

Back to NSD ,we will see how it works in simple steps, as follows(8)

- Firstly, the server provider which is the Android TV in our case must register and broadcast to the local network that there's a server called "X" has been registered
- Secondly, the Client start looking for a server and start a new thread for waiting
- Thirdly, the service response to the client with the server information, it includes the Server's service name, the IP address, the Port that's listening on and the service type (TCP, UDP, HTTP ..etc)

- finally, the client get all the information needed for the server and start talking to the server.

and this part of the project is provided in Google Developers documentations as follows in the figures

Server NSD

- Step 1: setup the listener for the service and override the required functions for it.

```java
public void initializeRegistrationListener() {
    registrationListener = new NsdManager.RegistrationListener() {

        @Override
        public void onServiceRegistered(NsdServiceInfo NsdServiceInfo) {
            // Save the service name. Android may have changed it in order to
            // resolve a conflict, so update the name you initially requested
            // with the name Android actually used.
            serviceName = NsdServiceInfo.getServiceName();
        }

        @Override
        public void onRegistrationFailed(NsdServiceInfo serviceInfo, int errorCode) {
            // Registration failed! Put debugging code here to determine why.
        }

        @Override
        public void onServiceUnregistered(NsdServiceInfo arg0) {
            // Service has been unregistered. This only happens when you call
            // NsdManager.unregisterService() and pass in this listener.
        }

        @Override
        public void onUnregistrationFailed(NsdServiceInfo serviceInfo, int errorCode) {
            // Unregistration failed. Put debugging code here to determine why.
        }
    };
}
```

Figure 3: Setup the listener for the service (Step 1)(Server)

- Step 2: Register the service on the network and fill the service information. [1]

[1] The port used here is a hard-coded one, in our application we used a dynamic method to get an available port by creating a new server socket
ServerSocket serverSocket = new ServerSocket(0);

```java
public void registerService(int port) {
    // Create the NsdServiceInfo object, and populate it.
    NsdServiceInfo serviceInfo = new NsdServiceInfo();

    // The name is subject to change based on conflicts
    // with other services advertised on the same network.
    serviceInfo.setServiceName("NsdChat");
    serviceInfo.setServiceType("_nsdchat._tcp");
    serviceInfo.setPort(port);
    ...
}
```

Figure 4: Register your service on the network using NSD (Step 2)(Server)

Client NSD

- Step 1: Discover the available services on the local network[2]
- Step 2: Connect to the service

Note: The NSD for server and client are robust that automatically register if it disconnect for some reasons.

// Store the chosen port.
int localPort = serverSocket.getLocalPort();

[2]the service name is a step forward to verify that client is looking for that server, plain text as password, more over in our application we send a random number that's appears on the screen over the TCP connection which allow us to make further step in security aspect.

```java
public void initializeDiscoveryListener() {

    // Instantiate a new DiscoveryListener
    discoveryListener = new NsdManager.DiscoveryListener() {

        // Called as soon as service discovery begins.
        @Override
        public void onDiscoveryStarted(String regType) {
            Log.d(TAG, "Service discovery started");
        }

        @Override
        public void onServiceFound(NsdServiceInfo service) {
            // A service was found! Do something with it.
            Log.d(TAG, "Service discovery success" + service);
            if (!service.getServiceType().equals(SERVICE_TYPE)) {
                // Service type is the string containing the protocol and
                // transport layer for this service.
                Log.d(TAG, "Unknown Service Type: " + service.getServiceType());
            } else if (service.getServiceName().equals(serviceName)) {
                // The name of the service tells the user what they'd be
                // connecting to. It could be "Bob's Chat App".
                Log.d(TAG, "Same machine: " + serviceName);
            } else if (service.getServiceName().contains("NsdChat")){
                nsdManager.resolveService(service, resolveListener);
            }
        }

        @Override
        public void onServiceLost(NsdServiceInfo service) {
            // When the network service is no longer available.
            // Internal bookkeeping code goes here.
            Log.e(TAG, "service lost: " + service);
        }

        @Override
        public void onDiscoveryStopped(String serviceType) {
            Log.i(TAG, "Discovery stopped: " + serviceType);
        }

        @Override
        public void onStartDiscoveryFailed(String serviceType, int errorCode) {
            Log.e(TAG, "Discovery failed: Error code:" + errorCode);
            nsdManager.stopServiceDiscovery(this);
        }

        @Override
        public void onStopDiscoveryFailed(String serviceType, int errorCode) {
            Log.e(TAG, "Discovery failed: Error code:" + errorCode);
            nsdManager.stopServiceDiscovery(this);
        }
    };
}
```

Figure 5: Discover the services (Step 1) (Client)

```java
public void initializeResolveListener() {
    resolveListener = new NsdManager.ResolveListener() {

        @Override
        public void onResolveFailed(NsdServiceInfo serviceInfo, int errorCode) {
            // Called when the resolve fails. Use the error code to debug.
            Log.e(TAG, "Resolve failed: " + errorCode);
        }

        @Override
        public void onServiceResolved(NsdServiceInfo serviceInfo) {
            Log.e(TAG, "Resolve Succeeded. " + serviceInfo);

            if (serviceInfo.getServiceName().equals(serviceName)) {
                Log.d(TAG, "Same IP.");
                return;
            }
            mService = serviceInfo;
            int port = mService.getPort();
            InetAddress host = mService.getHost();
        }
    };
}
```

Figure 6: Discover the services (Step 2) (Client)

4.2 Android Controller

In this Sub-section we will talk about the controller for Android Devices and how we implement it.

4.2.1 Layout

The layout for the controller is built dynamically using the JSON file (the output for the wizard) and it's also adoptive to the screen size of the device.

- Login Page

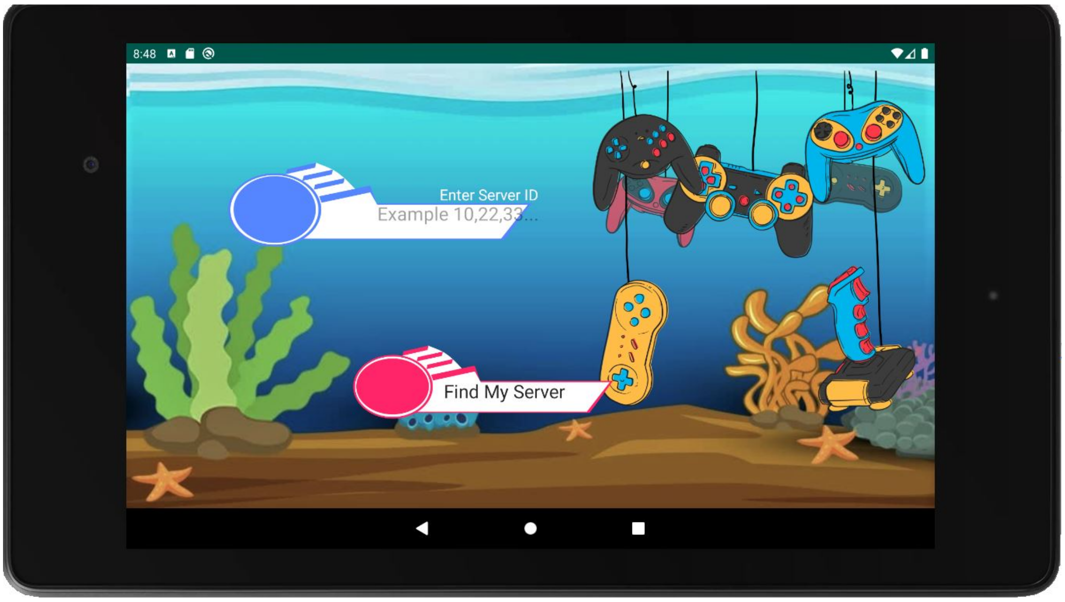

Figure 7: Login Page for Controller

- Choose the Mode page: in this page you can select which mode you want to play after complete the registration

Figure 8: Modes page for Controller

– Learn Mode

Figure 9: Learn Mode for Controller

– Match mode

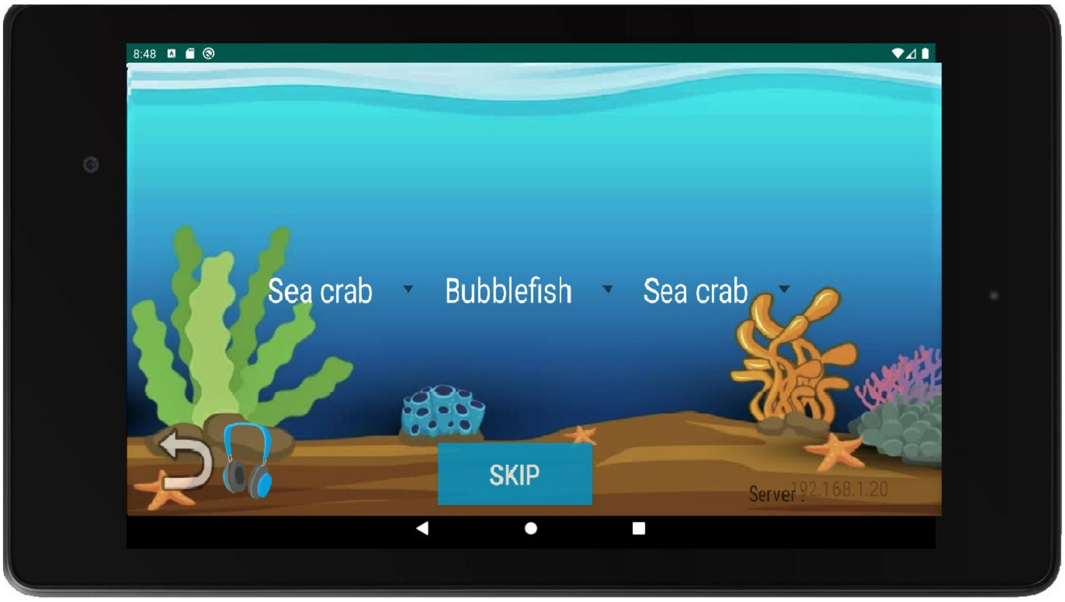

Figure 10: Match Mode for Controller

– Guess mode

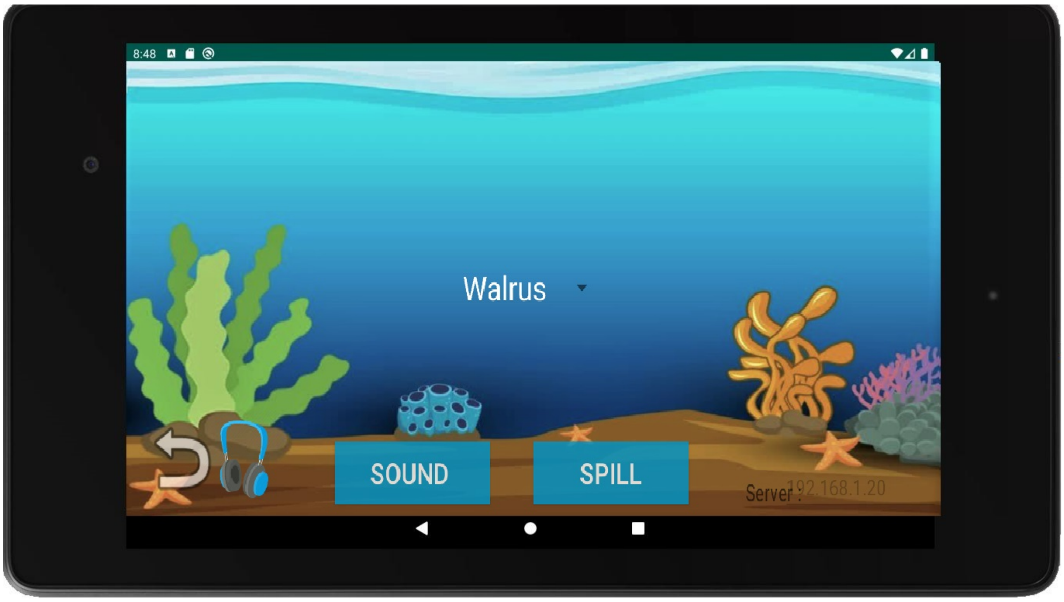

Figure 11: Guess Mode for Controller

4.3 Back-end for Controller

- Login page: in this page which is build in the top of NSD, you must enter the ID that's appears on the TV login page to login, it's used as password for the connection.
- Modes pages: in this page all the connection is over TCP connection

the process is going over the following flowchart

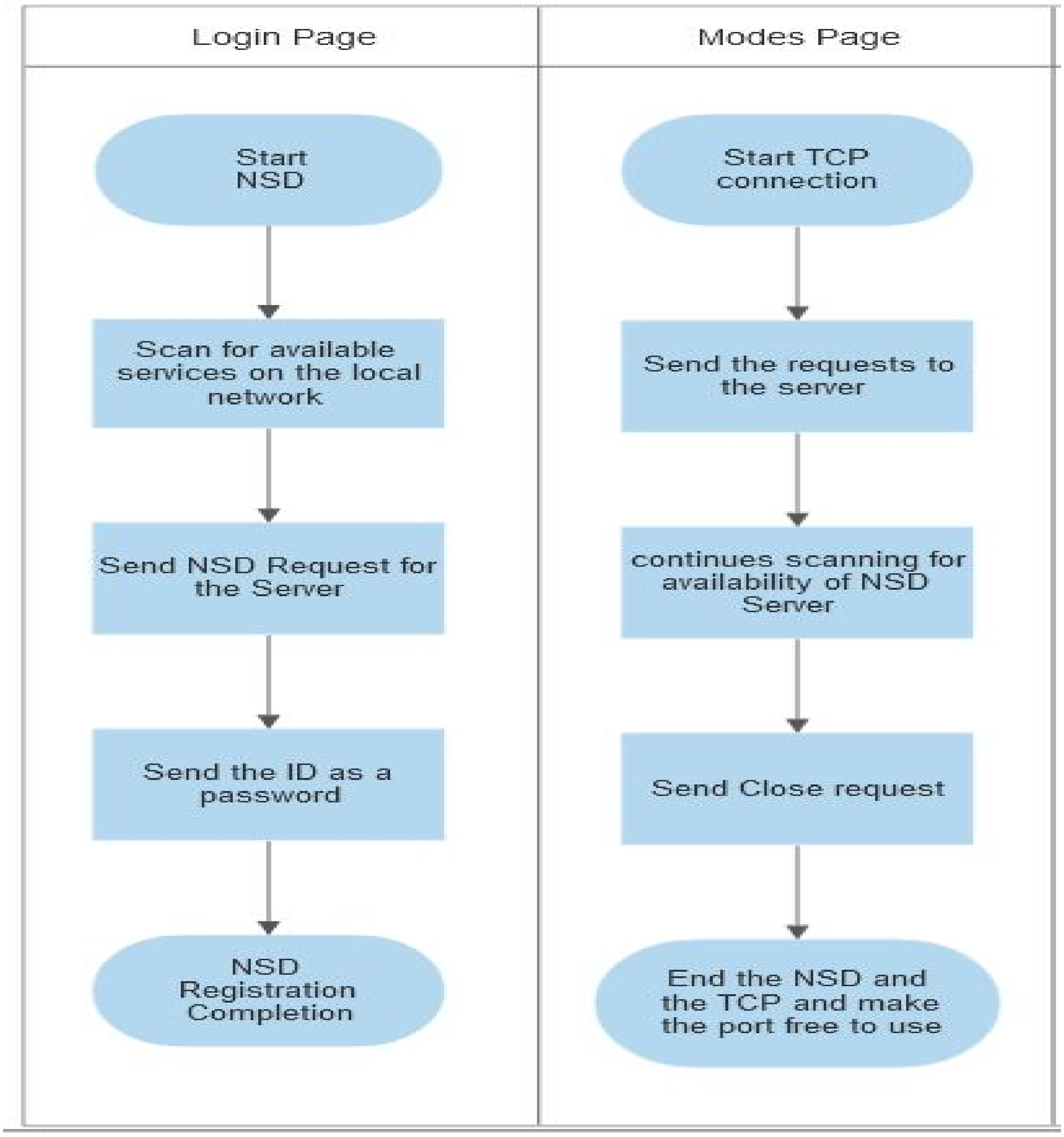

Figure 12: Login page and Modes page Flowcharts

- Learn Page: the kid can choose any object, it will appear on the TV and he/she can play the sound of the object or spell the alphabet.
- Match page: The kid should match the photo on the TV with the suitable and correct answer on the controller or he/she can skip that level.

- Guess page: the kid should know which object's sound is playing in the background or can skip the level also.

Notes:

The game is supporting multi-player so the parents can help the kid by playing with the kid on the same level at the same time.

Also there's a sound button on each page so the kid can turn on/off the sound on the controller.

4.4 TV Game

4.5 Layout

- Home page, this page is a android page build to start the NSD Server our application is support up to 4 players on the same game.

Figure 13: Login page for TV game

- Modes page, in this page we choose the mode from the controller
  - Learn Mode

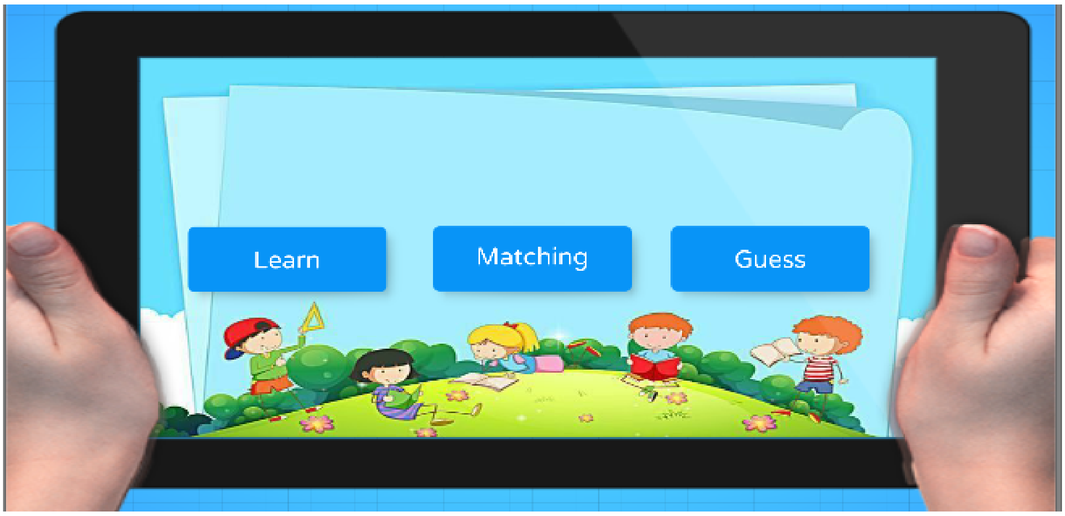

Figure 14: Login page for TV game

Figure 15: Learn mode page for TV game

– Match Mode

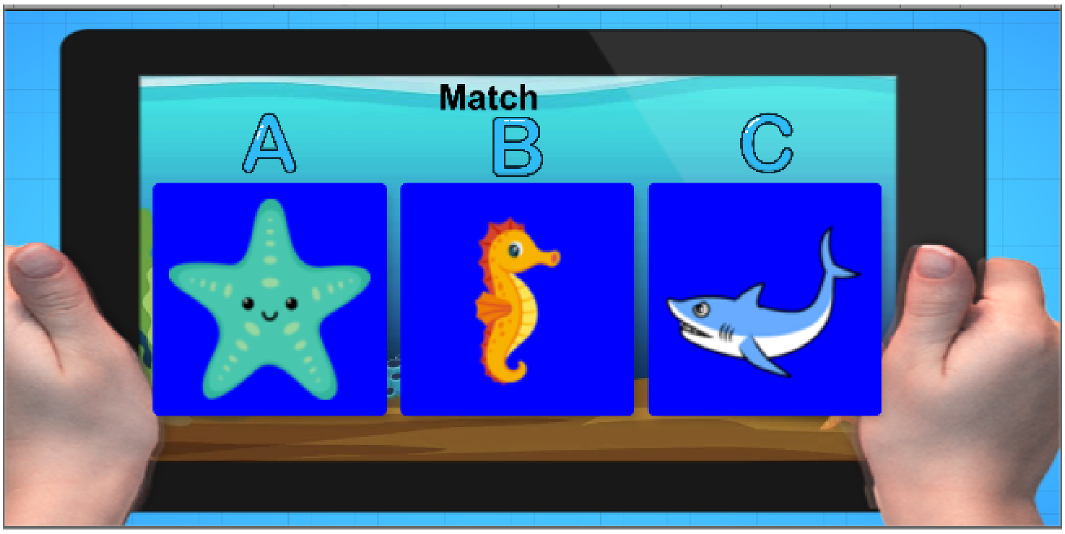

Figure 16: Match mode page for TV game

– Guess Mode

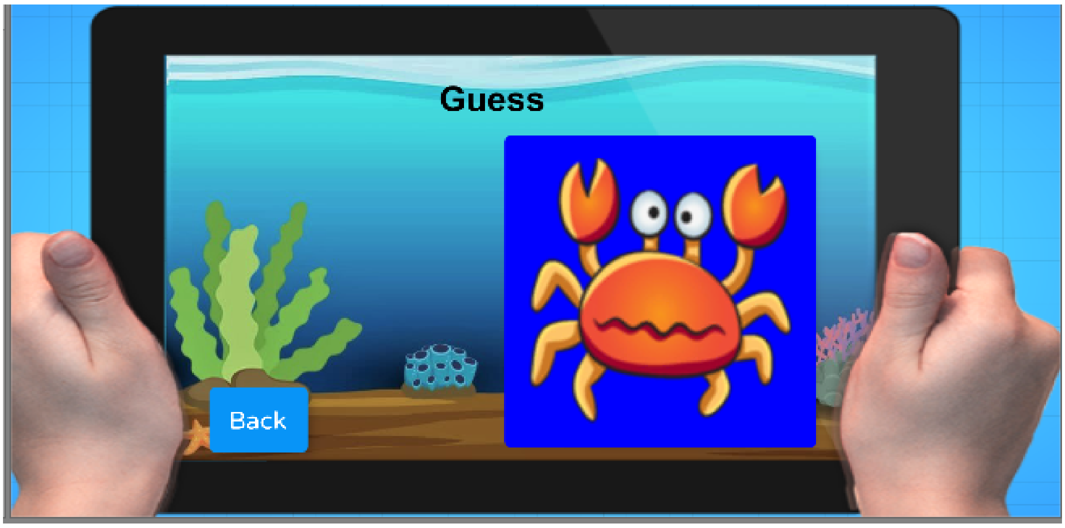

Figure 17: Guess mode page for TV game

4.6 Back-end for TV Game

this section illustrate the back-end of TV Game .

- **Main Page**:

  This page is the main in TV part , the player can choose any mode he want to play , these mods are Guess , Match , and learn .

  From any Mode we can Back to the main page by click back button .

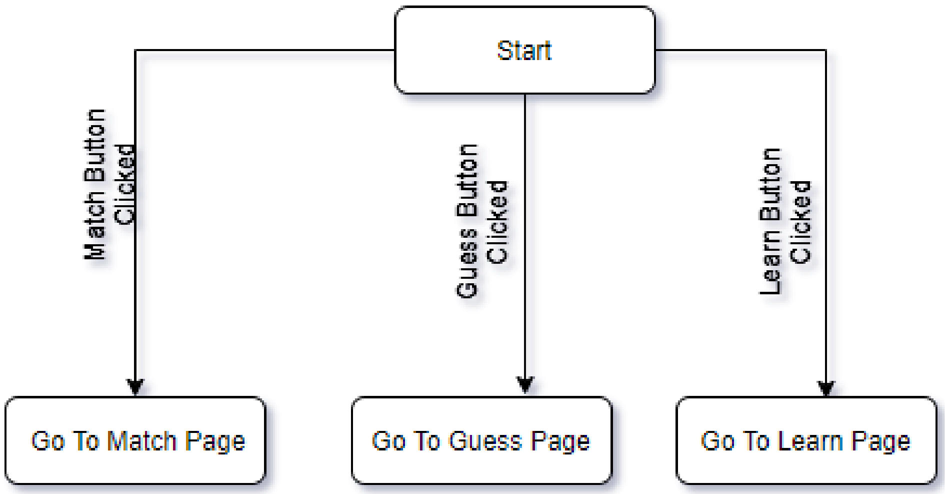

Figure 18: TV Main Page FlowChart

- **Learn Page**:

  In this mode the player click in any object in controller , it will be shown in TV with ability to hear the sound and the spelling of this object .

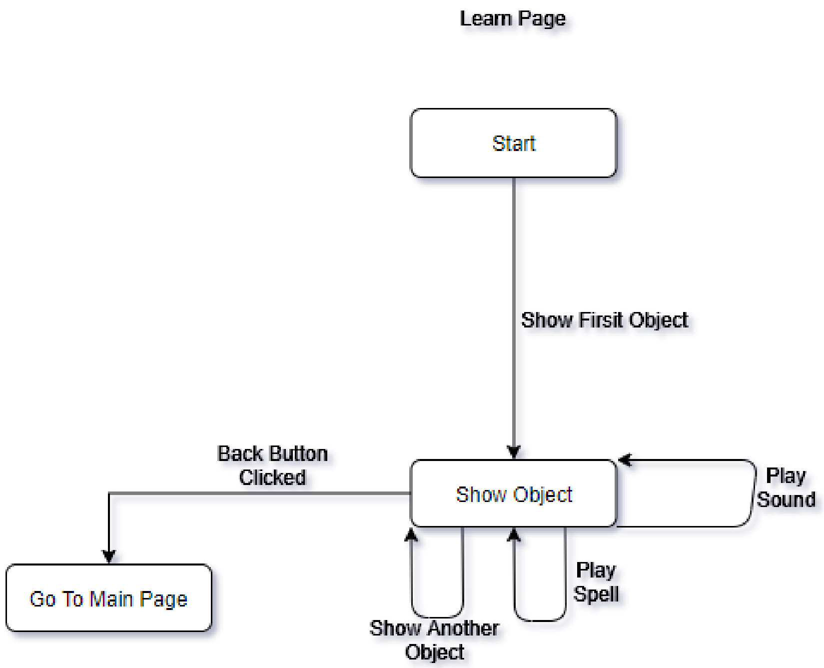

Figure 19: TV Learn Page FlowChart

- **Match Page**:

  in this type of challenges TV will generate new challenge of three random object , and the player should answer know the names of this object , if the answer is true the name will appear in TV .

  if the player want to skip current challenge he can generate new challenge by press in Skip Button .

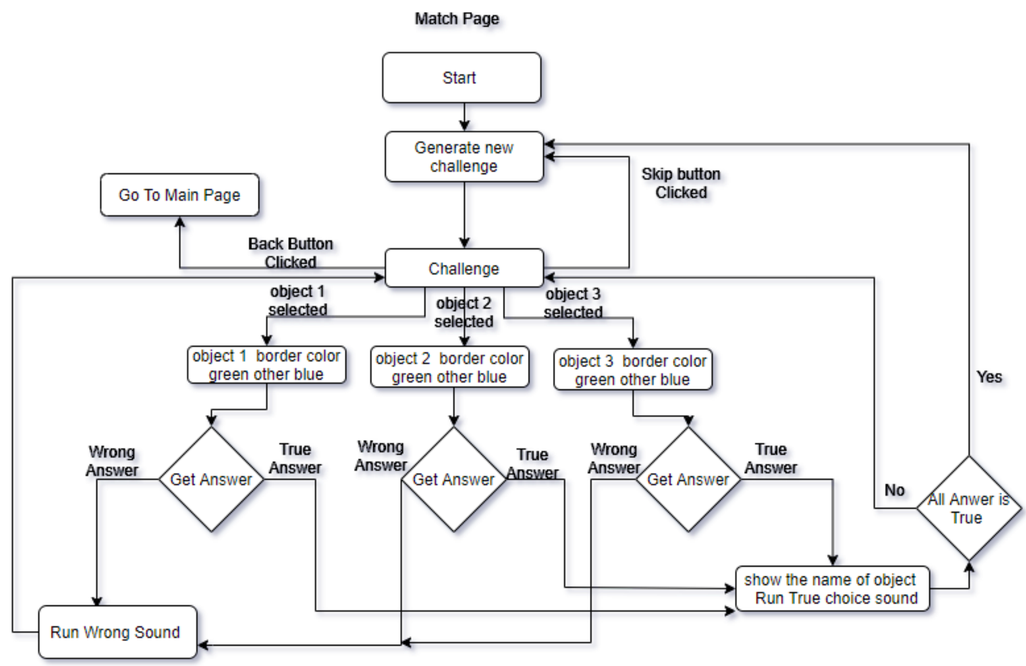

Figure 20: TV Match Page FlowChart

- **Guess Page**:

  In the Guess Stage , the TV generate Random challenge , and the player will determine the name of this object ,if the answer is true the name of object will appear on TV with some animation with green background for object image and run winner sound then generate new challenge, if not correct the background of object become red and wrong answer sound will play .

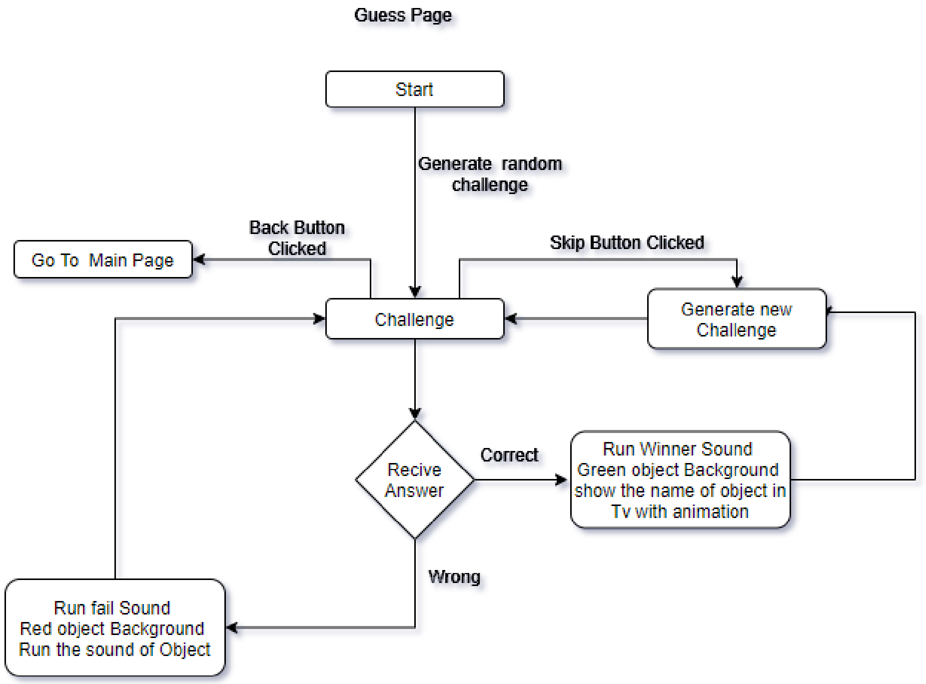

Figure 21: TV Guess Page FlowChart

4.7 Game Engine (Wizard)

this part of the project is built using the java, it's output is a JSON file for the controller and the TV game

- Home Page: in this page we asked for the information for the game such as the name, the number of rows and elements, the background sound and button sound.

- Information Page: in this page we asked for the information for the objects to used as data for both the controller and the TV game.

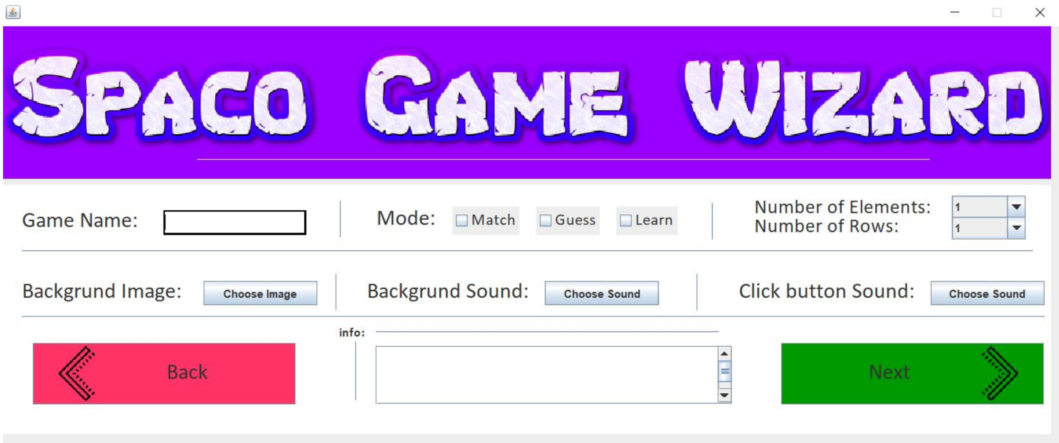

Figure 22: Home Page for Wizard

Figure 23: Information Page for Wizard

4.8 Communication between Android Controller, TV Game and the Wizard

The communication and the project is connected over the JSON file, the wizard output is a JSON file for both controller and TV game and the data for it [JSON, Photos, Sounds buttons and information], after that the controller and is communicate over the TCP.
As we are using Unity for our game project , we export the Unity project as android project then we add the NSD classes and used the "UnitySendMessage()" function to send the data from android to Unity engine as follows (11)

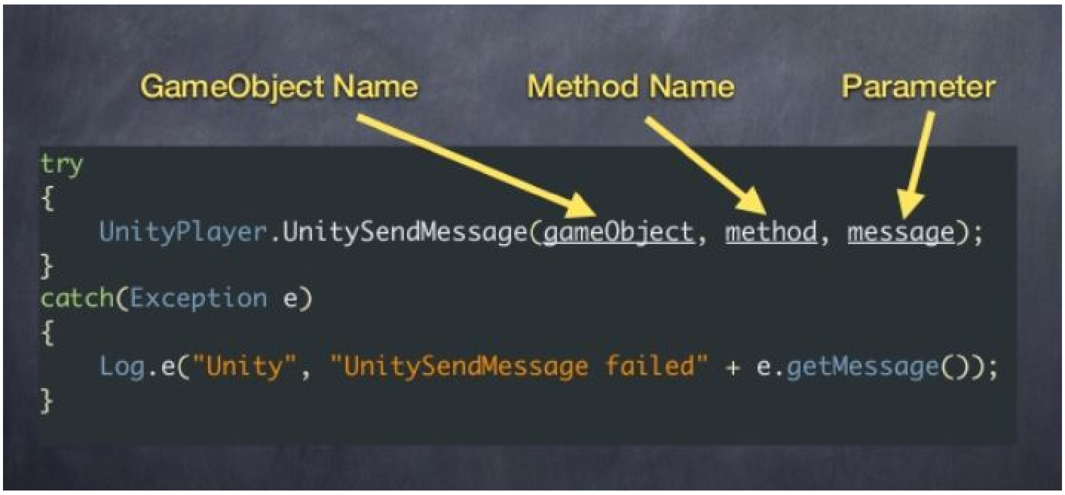

Figure 24: UnitySendMessage Function

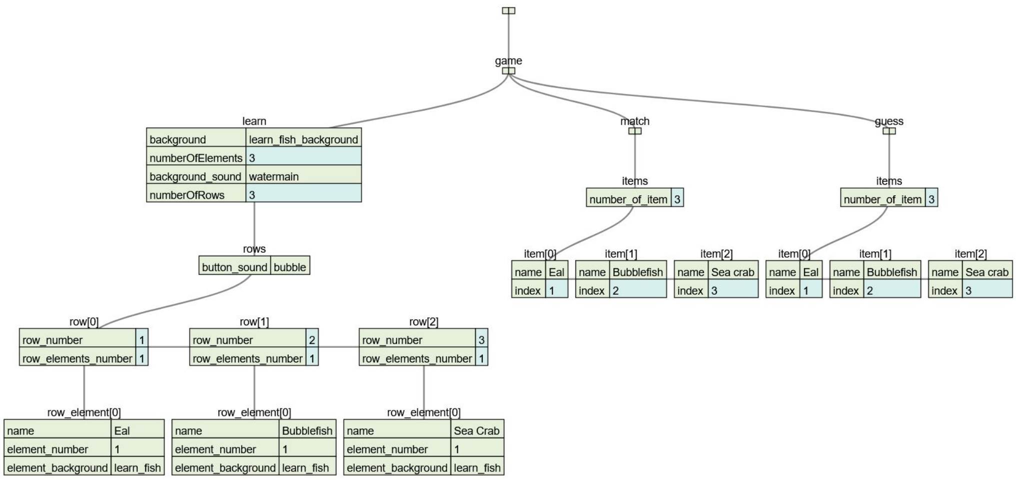

Figure 25: JSON diagram Example

5 Results and Discussion

This project is complete TV-centric educational game , which improve the abilities of kids in specific areas ,also the content of information can added easily by using wizard .

5.1 Outcome of Project

- Complete Educational video Game .
- Change the content of objects by using wizard .
- each type of challenge develops a certain ability for child .
- Using Network service Discovery(NSD) as the Backbone of communication Over TCP which support reliable communication .

6 Conclusion & Discussion

6.1 Conclusion

After this Project we earned new skills :

- Learn C programming Language , and Java For Android.
- Working with JSON Files by using LitJson Library .
- learning Unity , and gain some experience in Game Development .
- Learning Network service (NSD) to exchange data between TV and Controller .

6.2 Recommendation

Unity Editor is full IDE , but not all properties and assets are free , also there is sound extension cant be uploaded at run times , and it needs to license , and the advantage of Unity is support most used platforms such as Android , IOS , Windows .
we recommended to learn it before one month from semester begin to be familiar with.

6.3 Future Work

- Support different platform for controller .
- Support 3D object in TV .
- Enhance the User Interface(UI) for TV and Controller .
- Adding new types of challenges in the game .
- Adding new sections for Stories and children movies .

7 Reference

References

[1] P. Cesar, D. Geerts, "Past,Present,and Future of social TV: A Categorization", Proceedings of the IEEE Consumer Communications and Networking Conference, pp.347-351, 2011.

[2] J. Baek,B. Yun "A sequence-action recognition applying state machine for user interface",IEEE Transactions on Consumer Electronics, pp.719-726, 2008.

[3] N. Kosutic, M. Mitrovic, M. Z. Bjelica, V. Jelovac, "The Concept and Implementation of a Game Development Platform Based on the Integration of Consumer Electronic Devices with Android OS", Proceedings of the ETRAN 2012, pp. 1-4, 2012.

[4] Net Educational System Ltd., Nessy Learning Programme site: https://www.nessy.com/us/ , [Accessed on May 24th,2020]

[5] Paraskeva, F., Mysirlaki, S., & Papagianni, A. (2010). Multiplayer online games as educational tools: Facing new challenges in learning. Computers & Education, 54(2), 498–505. doi:10.1016/j.compedu.2009.09.001

[6] https://www.ietf.org/rfc/rfc6762.txt [Accessed on May 24th, 2020]

[7] http://www.dns-sd.org/ [Accessed on May 24, 2020]

[8] https://developer.android.com/training/connect-devices-wirelessly/nsd [Accessed on May 24th, 2020]

[9] Din, F., Calao, J. (2001). The effects of playing educational video games on kindergarten achievementChild Study Journal, 31(2), 95–103.

[10] Rosas, R., Nussbaum, M., Cumsille, P., Marianov, V., Correa, M., Flores, P., Grau, V., Lagos, F., López, X., López, V., others (2003). Beyond Nintendo: design and assessment of educational video games for first and second grade studentsComputers Education, 40(1), 71–94.

[11] https://docs.unity3d.com/530/Documentation/Manual/PluginsForAndroid.html [accessed on May 25th, 2020]