Developing a mobile game app themed about Libyan culture using Unity engine

Khaled M. Ben Hamed[1*], Nahed F. Farah[2], Mohanned Binmiskeen [3],

Mohamed Bashagha[4]

[1*]k.benhamed@uot.edu.ly, [2]n.farah@uot.edu.ly, [3]m.binmiskeen@uot.edu.ly,
[4]m.bashagha@uot.edu.ly,

[1,2,4] Department of Mobile Computing, Faculty of IT, University of Tripoli, Libya
[3] Department Internet Technologies, Faculty of IT, University of Tripoli, Libya
*Corresponding author email:

A B S T R A C T

This paper presents the design and implementation of an educational game App using Unity engine. The game aims to provide informative experience of Libyan traditions while keeping players entertained. Also, the game attempts to document Libyan fading traditions while being amusing and enjoyable. This game will be very first Libyan games to be launched into Google Play Store.

Keywords: Mobile Computing, Mobile Games, Mobile App Development, Unity Engine.

1 Introduction

With each passing day, the mobile game industry is witnessing alarming growth as the technology it runs on, evolves and matures[1]. In 2019, global mobile game revenue grew to $68.5B and this number is expected to hit $76.7B by the end of 2020[2]. This wide spread of mobile games provides an opportunity to spread information with it. Thus, making it one of the most efficient way for future education to help students to improve their technological, mathematical, scientific and engineering abilities[3].

Libya has a distinctive **history, traditions, and culture**[4]. A major problem facing Libya is the lack of cultural documentation and preservation of its historical treasures[5]. As games start to migrate more quickly and rapidly onto mobile devices, the culture and traditions of our ancestors face the risk of being forgotten and lost between the historical archives in museums forever. Of course, one mobile game app will not solve this problem. Nevertheless, it can only serve as a reminder of the impact these lost traditions have on our daily life style. In addition, to hopefully triggering the right people into taking care of this issue. This paper introduces the first mobile game themed about Libyan culture and traditions into the Google Play Store. The game integrates Iconic Libyan themes with game rules to create an educational and entertaining environment for the young youth and children to learn some of the oldest

traditions, words and cultures, while also filling the space represented by Libya in the global game market.

2 Research, analysis and backgrounds

The very first stage of the project was to research and analyze the basic concepts and ideas that should be implemented within the game. This included researching successful mobile games apps, as well as different development processes and comparing them. The advantages and disadvantages of each game category should be taken into consideration. Additionally, software development tools benefit and constrains are very important to determine the approach of the development process. Also, the research included Libyan history, culture and traditions, while focusing on the old city of Tripoli, and the traditional Libyan games, which are commonly played by the children in the streets of Libya. This allowed us to integrate a similar traditional vibe into the game.

From game comparisons research prospective, we choose the following games based on their popularity, success and creativity, these three attributes which we are looking forward to learn from and hopefully pass them into our game.

Super Mario Brothers [6]**:** it is the best-selling video game franchise of all time.

Created by: Shigeru Miyamoto

Considered an Ideal Model for good games design for many reasons such as:

Importance of the name: Miyamoto commented that if he had named Mario "Mr. Video", Mario likely would have "disappeared off the face of the Earth".

Importance of cultural reference: attributing Mario's nationality to his mustache.

Importance of colors: Due to the graphical limitations of arcade hardware at the time, Miyamoto clothed the character in red overalls and a blue shirt to contrast against each other and the background.

Importance of simplicity: red cap was added to avoid drawing the character's hairstyle. To make him appear human onscreen despite his small size.

Angry Birds: it became a very popular[7].

Created by Finnish company Rovio Entertainment

What makes it successful?

- Original story idea, comical style, and simplicity.
- Good advertising, easy to share, and low price.
- Small psychological details (large impact).
- Charming art style and cute characters.

From Libyan culture research prospective**,** before creating a game that represent a certain culture, we must have a deeper understanding of that culture, in this case Libyan games. Most commonly Libyan boys like to play games like *football*, *al-batch*, and *sab3a rashadat*. Libyan girls on the other hand play *niggiza*, *fathy ya warda-sakri ya warda*, and *rope jumping*[8].

A Closer look at – ***Battesh***. This game consists of small glass balls, also known as marbles, which come in different colors and variety of sizes. *"Battesh"* is one of the traditional Libyan games played by the children mostly around street alleys and schoolyards. Figure-1 shows the main character in our mobile game wearing traditional Libyan cloths and holding batch.

Figure 1: *Batch*

This traditional Libyan game seen less and less every day, as the word is shifting towards smart technology. Traditional Libyan food is one of the main aspects of culture[13], some of the most common Libyan food used in our mobile game are shown in Figure-2.

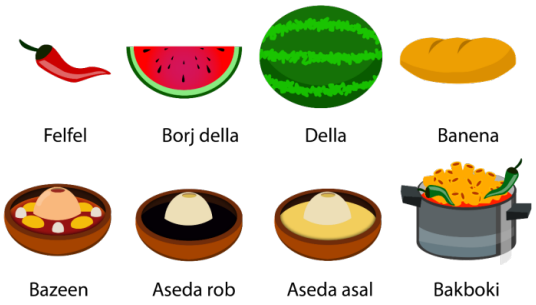

Figure 2: *Food items design.*

From implementation research prospective, there are several constrains to our project. All of which force us to make decisions and must be dealt with to succeed. These decisions are listed below:

- **Graphics 2d vs. 3d** [9]: It's not difficult to produce a simple 3d game, however, that will also defeat the objectives of this project. By doing 3d games, the quality of the work is decreased. 2d graphics allow greater focus on the design, the mechanics and the rules of the game, while also improving the quality of the work.

- **Vector vs. pixel** [10]: The pixel Art process is very difficult and time consuming as it deals with each and every pixel independently, forcing us to minimize the size of the picture and simplifying the details as much as possible, while trying to get our idea across to the user at the same time. Vector art clearly became the best choice.
- **Time constrains** are handled by efficient team work, organizational skills, planning ahead, making the right decisions and deciding priorities[11].
- **Budget constrains** limit the project team from the ability to obtain resources and might potentially limit the project scope.
- **Quality is restricted** by the specification of the product. The quality of the project is increased significantly by limiting the artwork themes, setting priorities and targeting specific devices and operating systems (in this case android devices).
- **Designers rules:** From the design prospective, the game integrates Iconic Libyan themes with game rules. there are a number of rules designers must follow to capture the attention and approval of a larger number of users. One of these rules is called "KISS" which stand for "keep it simple stupid"[12] by keeping the interface design simple, functional and understandable it covers a wider range of population around the world.

3 Requirements and planning

In Software engineering, a functional requirement defines a function of a system or its component. A non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system[13]. As shown in Table 1, a numeric rating (in descending order of importance) to be assigned to all functional requirements of the project.

Table 1: *priority scale*

priority	**1** very high	**2** high	**3** normal	**4** low	**5** optional

Let's start with the functional requirement, we set for the our mobile game:

- **Functional requirements**

Table 2 shows sample of game app functional requirements. Because games can be very demanding when it comes to functional requirements details, the functional requirements are split into three conceptual sections based on app system architecture.

1- **Game app section**, which is responsible for the app main menu and basic app navigations.
2- **Game world basics section**, it is an artificial universe, an imaginary place in which the events of the game occur. It represents the actual game play itself.
3- **The player basics section**, which may also be referred to as three basics, User, Player, Hero. **User:** the person who is using the game app. **Player**: is the user while in the state of interacting with the game world. Usually by playing in the

role of a game character. In this particular case "Batcha". **Hero:** is another name for the main character in this particular case, its (Batcha).

Table 2: *Game app functional requirements*

Game app basics			
ID	**requirement**	**Testing method**	**priority**
FRGB01	The system should allow user to become a player	Black Box	1
FRGB02	The system should allow easy access to game world	Black Box	1
FRGB03	The system should allow user to choose game level	Black Box	1
FRGB04	The system should allow user to choose language	Black Box	1
FRGB05	The system should allow user to choose language	Black Box	1
...	..	..	.

Once the Mobile game app functional requirements have been sorted and prioritized, Next, we list some of the non-functional requirements:

- **Non–functional requirements**
  - **Response time:** The average response time between click and reaction must be less than 0.5 seconds.
  - **Required resources:** The game should be able to run with minimum of 512 MB of RAM. And shouldn't exceed 100 MB of total size.
  - **Platform:** The game must run on Android mobile devices.
  - **Frames rate:** The minimum frame rate should be more than 15 frames per second.

Use case diagrams

As a sample one of the use case diagrams, Figure-3 represents the interaction between the user and main menu.

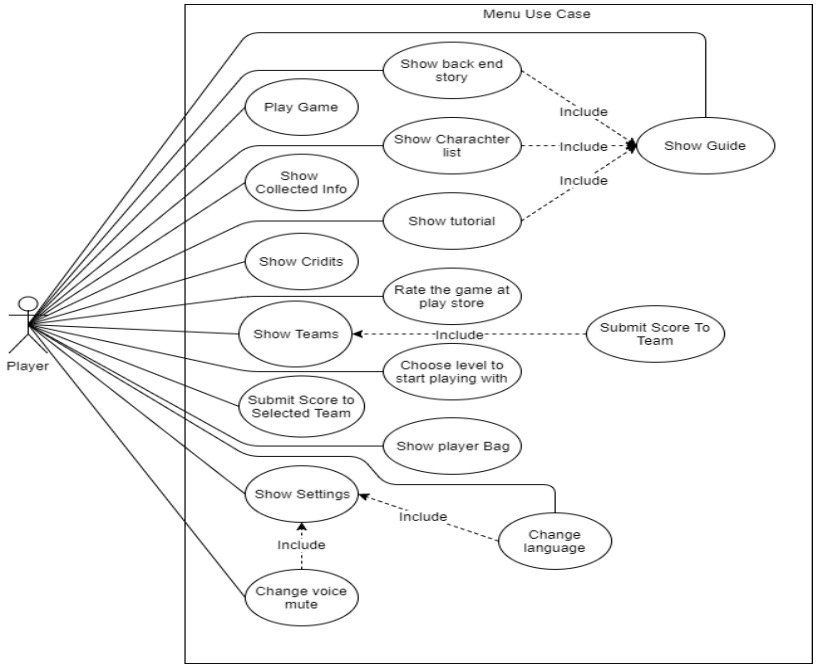

Figure 3: *Main menu use case diagram*

Activity diagram

Our mobile game play activity is shown in Figure-4, and it's showing the different activities that the user can perform in the game play menu.

Die: refers to the event in which the player dies, from any cause,

Win the Game: refers to the action of the player winning the game and this happens when the player goes to the end of the level without getting die.

Pause Menu: refers to the action of pausing the game when clicking on the pause button.

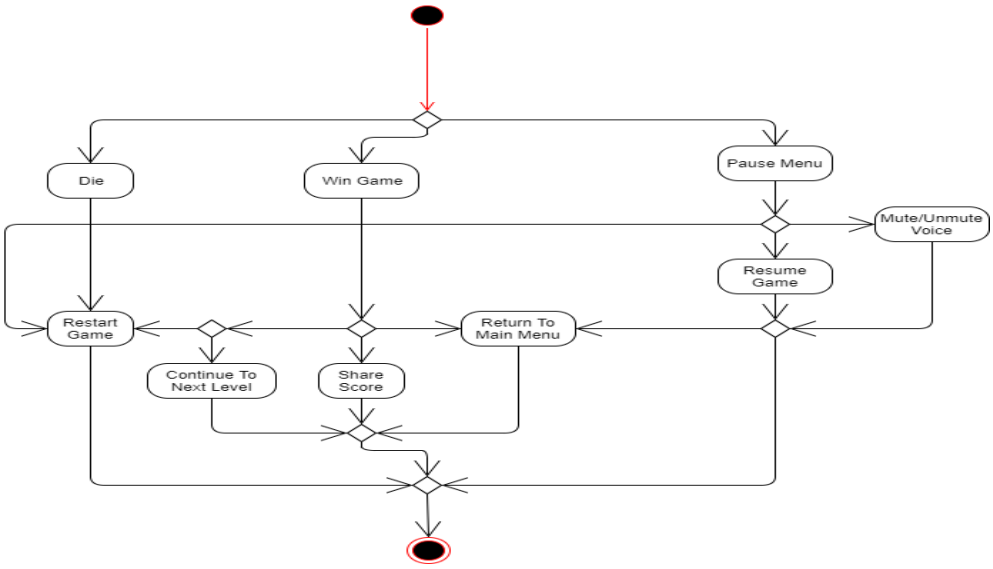

Figure 4: Game play activity diagram

4 Art and software design

Game design is split to art design and software design. First we must understand the relation.

4.1 Art design

The art design process is a very intense part of any game project and so it was for our mobile game too. Usually it's handled by a professional team, or by using premade assets which can be bought online. Thankfully we have managed to complete all the necessary elements of the game art in time. We made sure ALL elements are original in design to our mobile game only. The design of the five main characters of the our mobile game are shown in figure-5. Shown in figure-6, Azooz algayla – a famous evil character which is known through old Libyan stories, the old lady is known to kidnap kids who misbehave in the afternoon. Civilian and evil theme, in figure-7, are characters that are casually spread across game levels, their main function is to stop player from throwing batch randomly within the game, as they decrease the score with each hit they receive. Background scenes are a selection of blocks that represent the old city of Tripoli and some of its most important sights. Selected scenes of our mobile game are shown in figure-8.

In

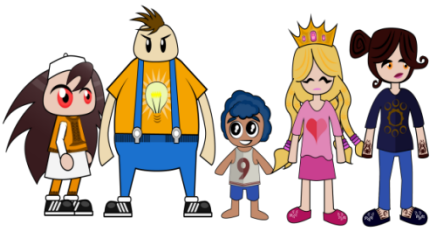

Figure 5: *Five main characters design*

Figure 6: *3zooz algayla design*

figure-8, the Evil theme of the game, shows that the dark side of the game which the player is fighting against. The sky background image changes randomly at the start of each game session.

Figure 7: *Civilians and evil theme design*

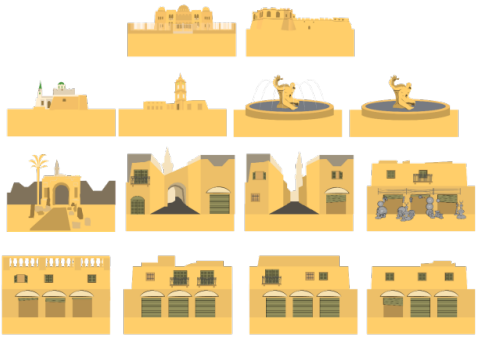

Figure 8: *Background compatible blocks design*

The figure-9 represent the variety of different animations that have been implemented into the hero character (Batcha) while jumping animation of the Our mobile game to represent different game states.

Figure 9: *Hero jumping animation*

Educational aspects are designed to help users to learn about Libya (the past, present and future….), culture and traditions, Libyan people and words (Libyan phrases), Geography (Map), and monuments and sightseeing.

4.2 Software design

As a data solution, Class serialization is chosen instead of relational databases so the data model implementation will be classes and class diagrams will be drawn to illustrate the data in them, but first there's explanation and figures of the Entity Relation Diagram.
An entity-relationship model (ERM) is an abstract conceptual representation of structured data. Entity-relationship modelling is a relational schema database modelling method, used in software engineering to produce a type of conceptual data model (or semantic data model) of a system, often a relational database, and its requirements in a top-down fashion.
The figure-10 illustrate the whole system Entity relationship diagram.
In the next section, we discuss the tools used in the implementations and testing conducted.

5 Implementation

In the implementation, we favoured Unity vs. libGDX. The problem we had with libGDX was that we were unable to display the Arabic letter on the screen[14].

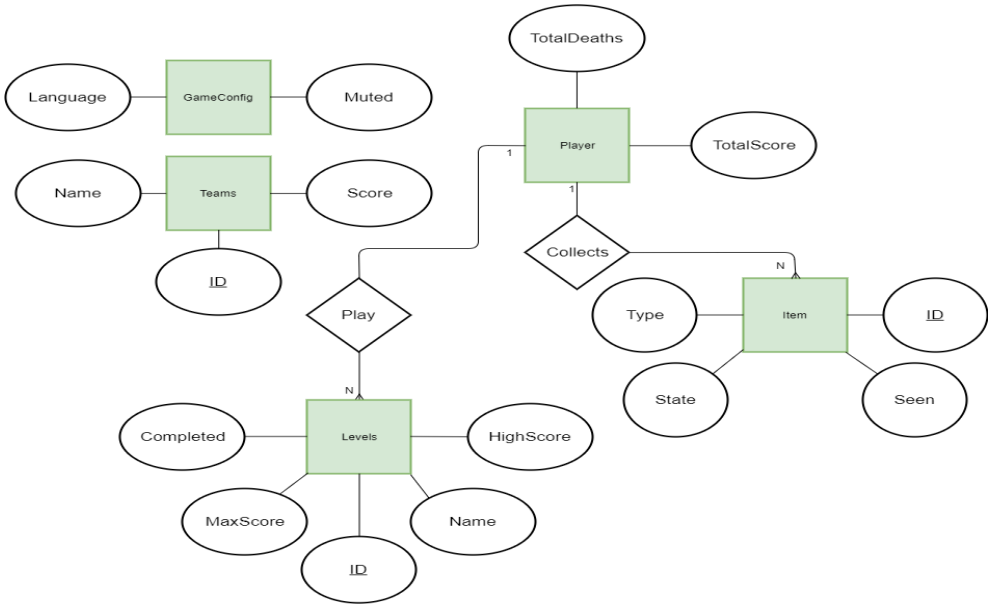

Figure 10: *game ERD*

5.1 Developments tools

C# programming language: We choose it for writing scripts in Unity, we choose it because its modern language packed with a lot of intensive features that will help us to get the job done in less time and in a more efficient way [15].

Localization Editor(LE): It gave us the ability to manage the variation of languages in our mobile game app and separate the texts of the UI from the code to make the editing process of the string more effective and less time consuming[16].

Spriter2UnityDX: We used it to translate the animations created by Spriter into language that can be understandable by Unity[17].

SpriterDotNet: It's scripts that gave us the ability to run the imported Spriter project in the .NET framework[18].

Firebase Realtime Database: W used it as a cloud-hosted database [19]. Our data is stored as JSON and synchronized in Realtime to every connected client.

5.2 Testing and debugging

We used white box testing and black box testing. White box testing is the testing when the developer should know the inner workings and start test based on that. We used Path Testing technique to calculate the number of possible paths and the complexity of the app which can be defined as:

$$V(G) = E - N + 2$$

Where, E – No. of edges and N – No. of Nodes

$$V(G) = P + 1$$

Where P = No. of predicate nodes (node that contains condition)

Example : Get Teams Data Tests

- **Step 1:** Construction of graph with nodes and edges from the code, figure-11 Shows the Get Teams Data code with numbered paths, and figure-12 Shows all paths for this code.

- **Step 2:** Identification of independent paths.

Path 1: 1 – 2 – 12 – 13
Path 2: 1 – 2 – 3 – 4 – 5 – 13
Path 3: 1 – 2 – 3 – 4 – 6 – 7 – 8 – 9 – 13
Path 4: 1 – 2 – 3 – 4 – 6 – 7 – 8 – 10 – 11 – 13

- **Step 3:** Cyclomatic Complexity Calculation[20]

$E = 15, N = 13, P = 3$
$V(G) = E - N + 2$ $\quad V(G) = 15 - 13 + 2 = 4$
$V(G) = P + 1$ $\quad V(G) = 4$

- **Step 4**:- Design of Test Cases

The Table 3 shows test cases for selected path we created.

Table 3: *Path 2: 1 – 2 – 3 – 4 – 5 – 13*

Inputs	**Expected Results**
FindObjectOfType<GameManager>().Connected == false && Application.internetReachability == NetworkReachability.NotReachable	Hide the please wait text and show internet error message and hide the error text after 3 Seconds

Black Box Testing is the method that let the tester test the system without knowing the specifics of the code itself, so it gives the ability for everyone to try it. Here we conducted tests on several of the parts of the system against the project requirements. Table 4 Shows a selected test case of change language of our mobile game.

Table 4: *Test Case - 001*

TC-001	
Purpose	Change the Language
Setup	Lunch the game
Steps	Select the settings menu Choose language
Expected Reponses	The language of the entire app should be changed
Result	Pass
Requirement covered	FRGB05

```
{
1   DatabaseReference reference = FirebaseDatabase.DefaultInstance.RootReference;
    DatabaseReference teamsRef = reference.Child("Database").Child("Teams");
    GameManager gm = FindObjectOfType<GameManager>();

    GameObject.Find("PleaseWait").GetComponent<Animator>().SetTrigger("Show");
2   gm.Get("https://www.google.com", () =>
3   {
        Debug.Log("Completed");
4       if (!FindObjectOfType<GameManager>().Connected &&
            Application.internetReachability == NetworkReachability.NotReachable)
5       {
            GameObject errorText = GameObject.Find("ErrorText");
            errorText.GetComponent<Text>().enabled = true;
            errorText.GetComponent<LELocalize>().localized_string_key = "no_internet";
            errorText.GetComponent<LELocalize>().UpdateLocString();
            GameObject.Find("PleaseWait").GetComponent<Animator>().SetTrigger("Hide");
            Invoke("HideErrorText", 3);
        }
6       if (FindObjectOfType<GameManager>().Connected ||
            Application.internetReachability != NetworkReachability.NotReachable)
7       {
            teamsRef.GetValueAsync().ContinueWith((task1) =>
            {
                GameObject.Find("PleaseWait").GetComponent<Animator>().SetTrigger("Hide");
8               if (task1.IsFaulted)
9               {
                    // Handle the error...
                }
10              else if (task1.IsCompleted)
                {
11                  DataSnapshot snapshot = task1.Result;

                    Debug.Log("Done: " + snapshot.Children);
                    Reafresh(snapshot.Children);
                }
            });
        }
    }, () =>
12  {
        GameObject errorText = GameObject.Find("ErrorText");
        errorText.GetComponent<Text>().enabled = true;
        errorText.GetComponent<LELocalize>().localized_string_key = "no_internet";
        errorText.GetComponent<LELocalize>().UpdateLocString();
        GameObject.Find("PleaseWait").GetComponent<Animator>().SetTrigger("Hide");
        Invoke("HideErrorText", 3);

        Debug.Log("error");
    });
}13
```

Figure 11: *Get teams data logic*

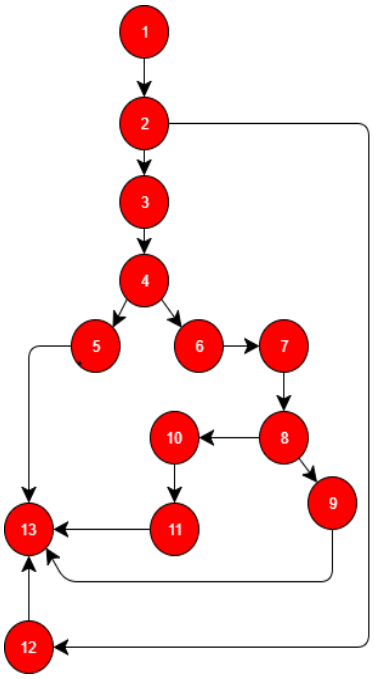

Figure 12: *Get teams data logic graph paths*

6 Conclusions:

In summary, we have created an educational Libyan game using Unity engine. From the design prospective, the game integrates Iconic Libyan themes with game rules. At different levels, various Libyan sights are displayed. Icons are presented in different shapes such as puzzles, tokens, or treasures for the player to collect. Also, obstacles, backgrounds and all game images have a sense of familiarity to Libyan life style and culture. From the implementation

prospective, the game was developed using Unity game engine which allowed us to create a game for cross platforms, combined with the art work made using Adobe Illustrator to create vector arts and Spriter to create character animations for game play. Other programs and development tools are used as needed. Furthermore, we adhere to Agile approach in the game development process.

Future work is mainly perfecting the game design, focusing on the user feedback and experience, we'll continue to fix any issues with the game play. There are a few options that can be added to receive a decent income if the project is to be successful, however these options need to be studied sensibly and put into further careful consideration as they may hinder the overall performance and popularity of the game.

References

[1] K. Alomari, T. Soomro, K. Shaalan, "Mobile Gaming Trends and Revenue Models", International Conference of Industrial Engineering and Other Applications of Applied Intelligent Systems, pp. 671-683, August, 2016. DOI:10.1007/978-3-319-42007-3_58

[2] Market research firm Newzoo, "*Global mobile market report*", September, 2019. Access online on 10 September 2020 at Available: https://venturebeat.com/2019/09/17/newzoo-global-game-revenues-will-hit-68-5-billion-in-2019/

[3] C. Chang, G. Hwang, "Trends in digital game-based learning in the mobile era: a systematic review of journal publications from 2007 to 2016", International Journal of Mobile Learning and Organisation, vol.13 no.1, pp.68 – 90, 2019. DOI: 10.1504/IJMLO.2019.096468

[4] D. Mattingly, S. Mclaren, E. Savage, Y. Al-Fasatwi, and K. Gadgood, *"The Libyan Desert, natural resources and cultural heritage"*, Society for Libyan Studies, *Cambridge University Press*, vol. 37, No.6, pp. 165-167, March, 2015. DOI: https://doi.org/10.1017/S0263718900004118

[5] N. Brodie, "Why Is No One Talking about Libya's Cultural Destruction?", A journal of the American Schools of Oriental Research, vol. 78, no. 3,pp. 212-217, September, 2015. DOI: 10.5615/neareastarch.78.3.0212

[6] M. Garin, "Super Mario, the new silent clown: Video game parodies as transformative comedy tools", International Journal of Cultural Studies, vol. 18, no 3, pp. 305-309, May 2015. https://doi.org/10.1177/1367877913513688

[7] M. Böhmer, B. Hecht, J. Schöning, A. Krüger, G. Bauer, "Falling asleep with Angry Birds, Facebook and Kindle: A large scale study on mobile application usage", Proceedings of the 13th Conference on Human-Computer Interaction with Mobile Devices and Services, August, 2011. DOI: 10.1145/2037373.2037383

[8] H. Biuk, "Libyan Child Play," Access online on 10 September 2020 at Available: http://www.tripolipost.com/articledetail.asp?c=17&i=9459.

[9] A. Sinicki, "Learn Unity for *Android Game Development* A Guide to Game Design, Development, and Marketing*", APress, 2017.* DOI 10.1007/978-1-4842-2704-6_2

[10] R. Rosenbaum, C. Tominski, "published Pixels vs. Vectors: Presentation of Large Images on Mobile Devices", Conference: International Workshop on Mobile Computing (IMC), Packi Publishing Ltd, Jan, 2003. Access online on 10 September 2020 at https://www.researchgate.net/publication/247160237

[11] S. B. Kaleel and S. Harishankar. "*Applying agile methodology in mobile software engineering: Android application development and its challenges", Technical report*, Department of Computer Science, Ryerson University, 2013.

[12] K. Bjornard , *K. principle*, *the Interaction Design Foundation.* Access online on 10 September 2020 at Available: https://www.interaction-design.org/literature/article/kiss-keep-it-simple-stupid-a-design-principle.

[13] F. Tsui, O. Karam, B. Bernal, "Essentials of Software Engineering 4th", Jones and Bartlett Learning, 2018.

[14] M. Zechner, "*libgdx*", 2013. Access online on 10 September 2020 at Available: *https://libgdx.badlogicgames.com/*

[15] A. Troelsen, P. Japikse, " Pro C# 8 with .NET Core 3 Foundational Principles and Practices in Programming" , Apress, 2020. DOI:10.1007/978-1-4842-5756-2

[16] L. Editior. [Online]. Access online on 10 September 2020 at Available: http://localizationeditor.com/

[17] Spriter2UnityDX, Spriter2UnityDX. Access online on 10 September 2020 at Available: https://github.com/Dharengo/Spriter2UnityDX

[18] SpriterDotNet, Access online on 10 September 2020 at Available: https://github.com/loodakrawa/SpriterDotNet.

[19] M. Ohyver, J. Moniaga, I. Sungkawa, B. Subagyo, I. Chandra, "*The Comparison Firebase Realtime Database and MySQL Database Performance using Wilcoxon Signed-Rank Test*", *the 4th international conference on computer science and computational intelligence*, vol.157, , pp. 396-405, September, 2019. https://doi.org/10.1016/j.procs.2019.08.231

[20] G. Jay, J. Hale, R. Smith, D. Hale, "Cyclomatic Complexity and Lines of Code: Empirical Evidence of a Stable Linear Relationship", Journal of Software Engineering and Applications, vol.2, no. 3, pp. 137-143, January, 2009. DOI: 10.4236/jsea.2009.23020