



Augmented Reality Based Tabletop Gaming and Multiplayer Platform

¹Shubham N. Patil, ²Vivek M. Chaudhari, ³Saurabh B. More, ⁴Mahesh V. Korade

¹B.E student, ²B.E. student, ³B.E. student, ⁴Assistant Professor
Department of Computer Engineering,
Sandip Institute of Engineering and Management, Nashik, India.

Abstract: Even though Augmented Reality has been making progress in the field of game development, there isn't any Augmented Reality (AR) gaming platform that succeeds massively in providing AR experiences without the need of AR specific hardware. This is due to various constraints, such as AR headgear being too expensive for customers to afford, limited hardware capabilities, etc. This paper aims to build an AR gaming platform that hosts lightweight games which can be played directly without download, and explores the development of such games. This platform also aims to solve issues such as excessive battery consumption by these apps, lack of hardware resources. It creates opportunities for developers. Tabletop AR games are traditional board based games, which can be played into a shared 3d environment. The platform and its games are accessible without the need of any other specific hardware, by using the capabilities of our modern smartphones alone.

Index Terms – Augmented Reality, Gaming platform, tabletop games.

I. INTRODUCTION

Augmented Reality (AR) in gaming has been growing fast, with companies launching their own headsets, and developers releasing game experiences that support them. Though it has been said that in the coming decade these headsets would be as common as smartphones are in today's world, they remain expensive for now. Majority of the people do not have access to them. Our intent is to create a platform that will serve these users with unique AR experiences, and give developers the ability to publish their games on this platform. A lot of AR smartphone games already today use the technology very sparingly. Unlike other areas of software development, there are no proper design standards set for designing AR applications [1]. This often leads to improper app designs, and can also lead to overuse of resources, and faster drain of battery. We thus also intend to set design guidelines for developers to follow when making apps for this platform. These guidelines will ensure that these apps are both battery and resource efficient and offer good performance. They will make sure these apps are secure. They will also bring developers together on the same page and make development easier and faster.

II. LITERATURE REVIEW

A group of researchers from University Teknologi, Malaysia [2] explored the creation of such games using collaborative augmented reality. This study provides an in depth insight into development of collaborative interfaces we want to create. Unity3D offers proper development framework to build these games [3], and AR development software kits such as Google ARCore, ARKit are supported by Unity [4]. We require collaboration between devices to provide a shared gameplay experience. From this study, we have learned the architecture of the two methods of collaboration between devices, collocated and remote collaboration. In collocated collaboration, users are present at the same place. In remote collaboration users are distant. To provide collocated collaboration, we utilize the Wi-Fi connection. One device acting as the master, hosts the game for all the players, and the rest connect to that hotspot. To provide remote collaboration, we use Photon Unity Networking (PUN) [5]. PUN provides a dedicated server that players across the globe can connect to, and is very efficient in performance. PUN gives us real time synchronization between devices using event callbacks via its server. Building the platform app and its interface in Unity gives us ability to use Unity native services. It also speeds up the production process with its cross platform development capabilities.

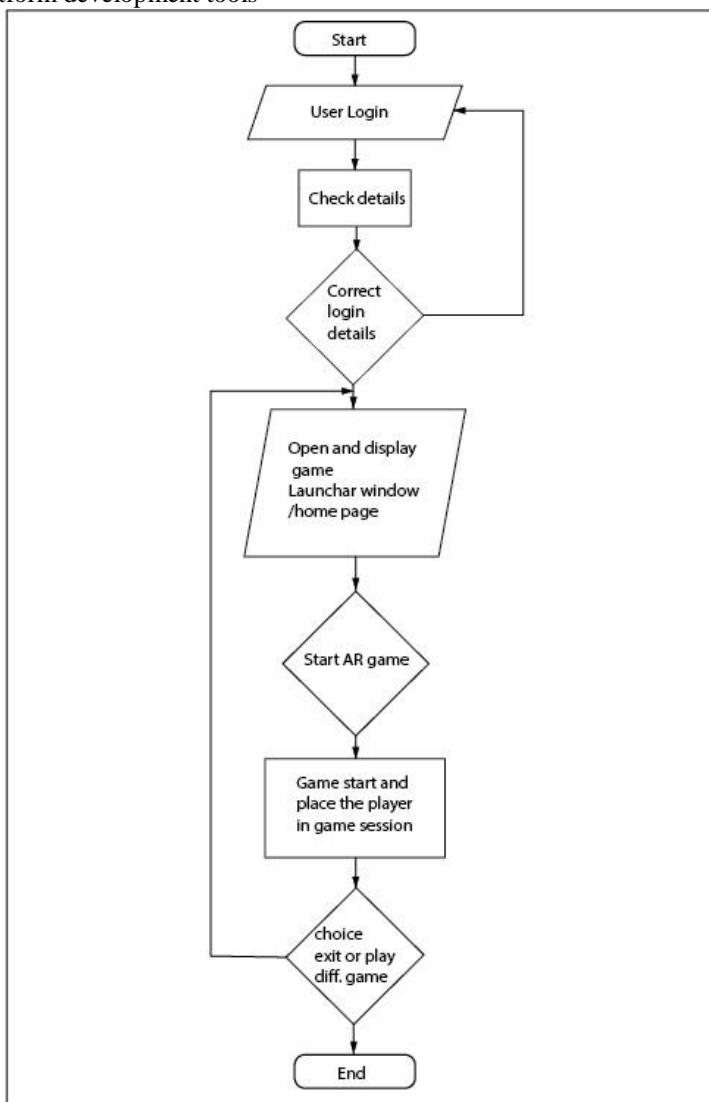
Firebase provides services such as analytics, cloud messaging, authentication, real time database, storage, crash reports, etc. Firebase is faster than traditional RDBMS to handle large amounts of data [6]. These services are highly useful in development of a gaming platform. Analytics allows us to monitor user behaviour in our app. Use of the free multi-platform firebase authentication decreases security concerns, and is easy to integrate. Its NoSQL cloud storage makes it easy for developers to store and transfer data for frontend and backend development. As the platform grows, scale-ups can be done as per the rising needs.

III. METHODOLOGY

The methodology in this paper includes using AR development software kits (ARCore, ARKit) with Unity 3D game engine. The prototype of the platform and games are built in Unity. Our platform uses PUN to provide both local and remote collaborative gaming experiences. PUN allows us to run multiple rooms at same server simultaneously. A web application is built and hosted on firebase services for the needs of the user classes below. Game developers here are developers who design games and want to publish on the platform. Platform developers are the platform makers. A conceptual architecture is explained below.

Users are categorized into four classes, by functions and privilege levels:

- **Players:**
 - Players can play the game locally / globally
 - Can access all player class functions
- **Moderators:**
 - Can moderate and maintain the platform,
 - Can approve games uploaded by developers after checking if the content meets the rules and requirements of the platform.
 - Reply to support tickets and forward to developer if necessary.
- **Game Developers:**
 - Can submit their games for publish approval by moderator
 - Can access all the developer tools/services provided by the platform for building the games
- **Administrators/Platform Developers:**
 - Can access the platforms functionality that is limited to the administrators such as testing, debugging and other platform development tools



Player functionality:

Login / Register: Upon start of the app, the first view visible is the user Login / Register page. If the user is new to the platform, he/she can choose register. After filling the form, the user id and password will be saved in our firebase database. If one is already registered, one can simply choose the login option and enter their id and password. After matching it against the saved id and password in firebase, if it is correct, one will be logged in.

Homepage: The next screen after login visible to the user is the home screen, which displays all the games available to play for the user, organized in different categories of gaming. There is a sidebar for user settings that includes user account functions such as edit profile picture, name, and whether to share the game players' data with developers to improve the platform.

Click to play: Players can click on any game to open a game window. Selecting play in this window downloads the game files and scripts required from the server and load the game. Then for single player game, the user is directly starts playing the

game. In multiplayer games, player has to choose between local or remote gameplay. Player has to either host the game or connect to a host in local environments, and when playing in global environments, player can join any game lobby being hosted in the list of lobbies online. Each lobby has a certain limit of how many players it can take maximum, which varies as per the game. At the start of each game player's game-data such as scores, stats, etc. is loaded from the firebase database, and it is updated after each game round.

Players can rate a game, and provide feedback for the developers on games on the game window page.

Moderator and Developer functionality:

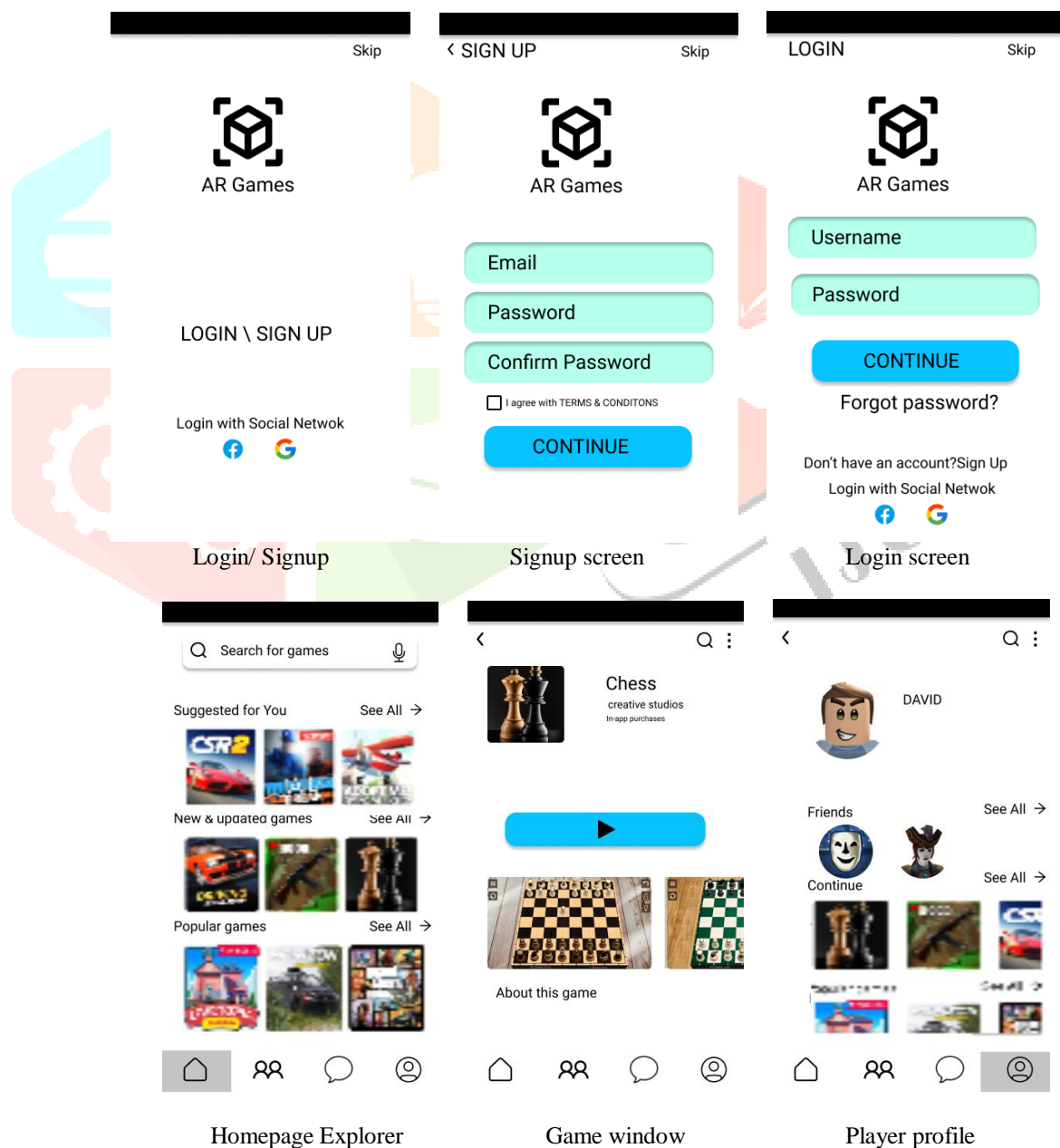
Moderators and Developers have access to the respective accounts on the web application. Developers can use this to submit their applications for publishing, access rules and regulations, design guidelines of the platform here, and also both of these can access the community forums.

Ad monetization:

Ads are shown to players at various times in the app, although not frequently. This generates revenue for the platform, out of which 85 percent share goes to the developers of the game. Players will have the option to become a donator to disable ads.

IV. RESULT

Some of the user interface of the platform is shown below:



V. CONCLUSION

Thus, the platform gives opportunity to developers to explore AR game development and earn for their games, while also giving the players unique experiences of AR. It motivates more developers to explore AR gaming by providing easy development scripting tools and support. It creates a community through forums, which is very helpful for new developers to learn fast. Tabletop games like Chess, Tic Tac Toe, dice games, snakes and ladders, monopoly, etc. are implemented and tested. The gameplay experience is very much enriched via use of AR, as opposed to playing computer games in a screen. The development of such apps brings us closer to making our vision of the metaverse a reality.

REFERENCES

- [1] C. Rolim, D. Schmalstieg, D. Kalkofen and V. Teichrieb, "[POSTER] Design Guidelines for Generating Augmented Reality Instructions," 2015 IEEE International Symposium on Mixed and Augmented Reality, 2015, pp. 120-123, doi: 10.1109/ISMAR.2015.36.
- [2] Yusof, Cik Suhaimi, Tian Sheng Low, Ajune Wanis Ismail and Mohd Shahrizal Sunar. "Collaborative Augmented Reality for Chess Game in Handheld Devices." 2019 IEEE Conference on Graphics and Media (GAME) (2019): 32-37.
- [3] Hussain, Afzal & Shakeel, Haad & Hussain, Faizan & Uddin, Nasir & Ghouri, Turab. (2020). Unity Game Development Engine: A Technical Survey. University of Sindh. Journal of Information and Communication Technology. 4.
- [4] "Getting started with AR development" - <https://docs.unity3d.com/2022.1/Documentation/Manual/AROverview.html>
- [5] "Photon Unity Networking (PUN)" - <https://doc.photonengine.com/en-us/pun/current/getting-started/pun-intro>
- [6] Khawas, Chunnu & Shah, Pritam. (2018). Application of Firebase in Android App Development-A Study. International Journal of Computer Applications. 179. 49-53. 10.5120/ijca2018917200.

