Project for Gesture Based UI Development

Soccer Shootout

By

Irakli Lomidze

G00275525

Git Repository

https://github.com/ika25/ARproject

Introduction

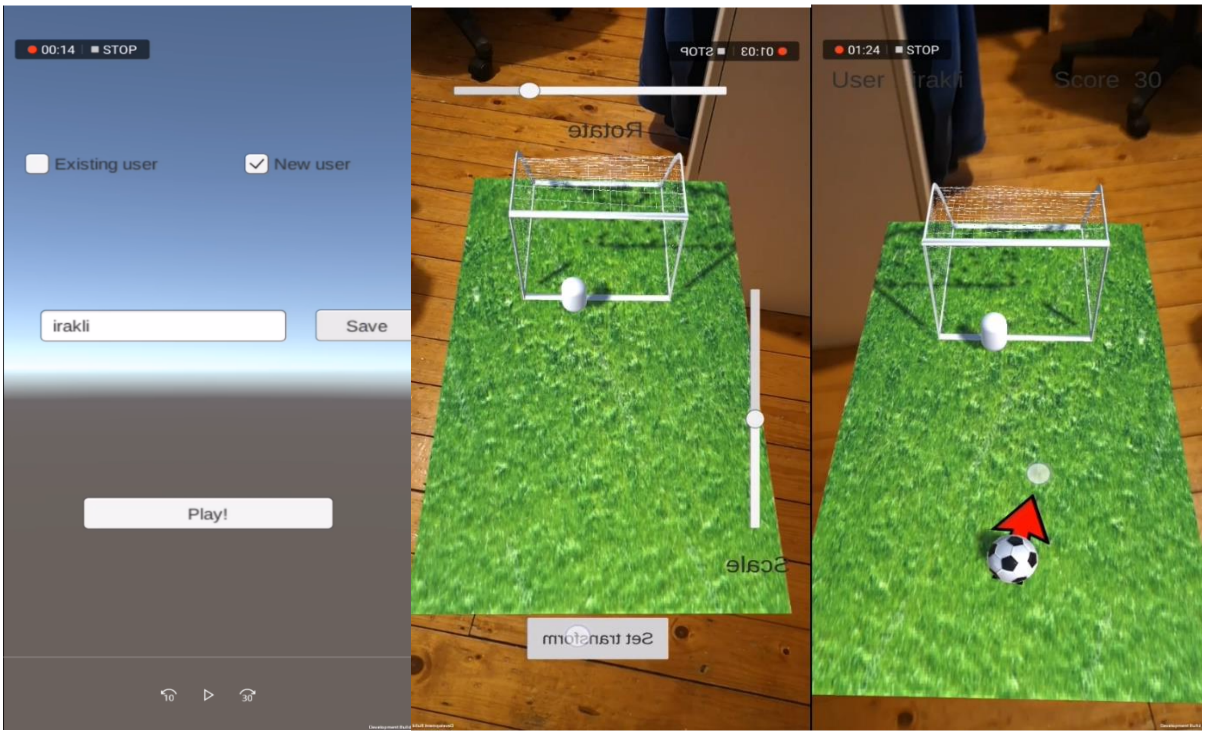
For my project development I chose to make application in Augmented Reality using ARcore in unity, for android devices.

After doing extended research on each available technology's that I was allowed to use for this project I decided to use AR technology as I already had access to hardware and software I needed to complete this project.

My other choice was Virtual Reality but I don't own hardware and due to circumstances we are In now I couldn't barrow hardware from college.

Soccer Shootout

The gesture-based games I chose to implement is small game using ARCore to detect ground, when ground is detected you can start interacting, you can change scale and rotate environment, when done click to play game, ball will drop and you are ready to kick the ball and try score past by keeper.



☐ Existing user ☑ New user

irakli

User irakli Score 30

Purpose of Application

The purpose of this application is to explore and experiment with the capabilities of the ARCore and the development components that come along with it such as the Visual Gesture Builder. At the end of development goal is to have an application that

- Naturally incorporates practical gestures into various components that contribute to the fluidity of the overall application
- Is fully context aware
- Provides a high rate of accuracy in relation to gesture recognition
- Uses those highly accurate gestures as a base for seamless navigation and to provide an additional layer of engagement and fun that wouldn’t otherwise be found in a traditional game-based application
- Provides continuous feedback to the user
- Contains input definitions that are simple to perform and non-cumbersome
- Has Interactions that are simple, easy to learn, recognize and master

Gesture Identification and Rationale

Augmented reality is the integration of game visual and audio content with the user's environment in real time. Unlike virtual reality gaming, which often requires a separate room or confined area to create an immersive environment, augmented reality gaming uses the existing environment and creates a playing field within it. While virtual reality games require specialized VR headsets, only some augmented reality systems use them. AR games are typically played on devices like smartphones, tablets and portable gaming systems.

An augmented reality game often superimposes a precreated environment on top of a user’s actual environment. The game itself can be as simple as a game of virtual checkers played on a table surface. More advanced AR games may actually build an environment from user surroundings. Such a game could involve, for example, in-game characters climbing from coffee tables to sofas on virtual bridges. Environment creation is a time-consuming task in game making and there is a constant demand for new scenery because once a user has explored an environment fully, they want to move on to a different one. AR gaming expands the playing field, taking advantage of the diversity of the real-world environment to keep the games interesting.

Gestures of the Application

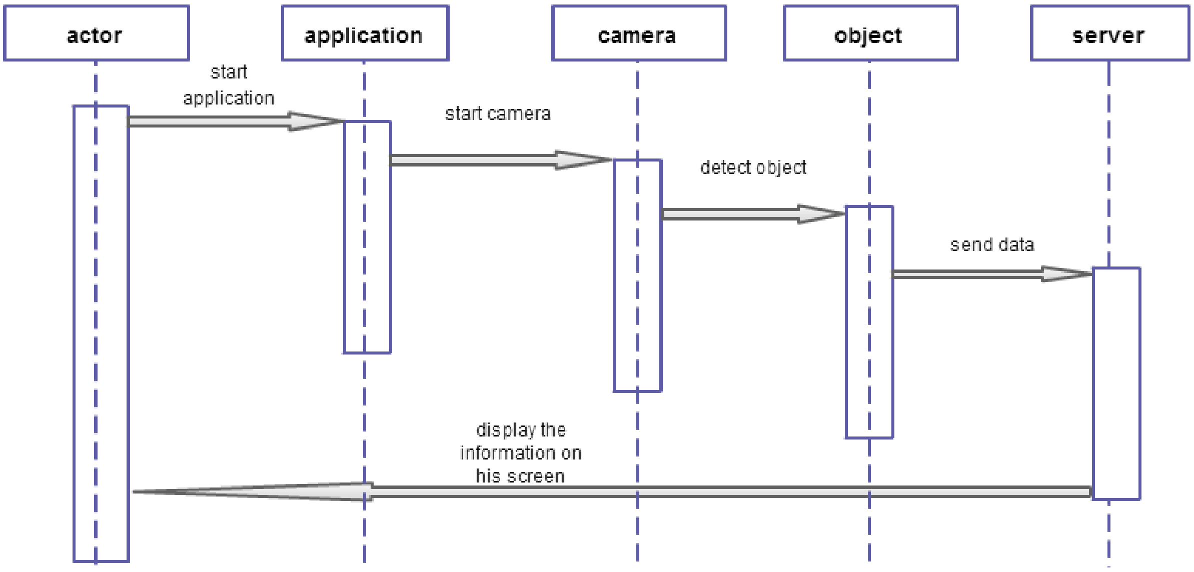
In this Game smartphones can be used to provide augmented reality experiences. augmented reality technology can be used through camera. The viewer can see the input from the camera being modified on the screen.

Augmented reality uses a camera to identify visual markers or objects, to showcase an overlay only when the marker is sensed by the device. Marker-based AR technology depends upon device camera to distinguish a marker from other real-world objects.

Not only the marker image but the position and orientation can also be calculated. Once recognized the marker on screen is replaced with a virtual 3D version of the corresponding object. This is done to permit the user to observe the object in more detail and from various angles. Rotating the marker would rotate the virtual replication as well.

The Game also uses touch gesture to kick the ball, way touch gesture works is that user touches the screen, leaving their finger motionless until the information is displayed of action occurs.

Solution Architecture



Core Technologies

- Unity – Version 2020.2.7f1
- ARcore sdk
- Android development in Unity

Core Libraries and Development Tools

- Arfoundation
- ARCore
- Scriptable object event system

Core Classes and Scripts

- Placement indicator
  - `Script for handling the placement indicator. Placement indicator is used to guide the user on where the instantiated AR objects lie in the real world.`
- Object Spawner
  - `Script for handling anything related to the instantiated AR objects in real world.`
- Swipe input manager
  - `Script for handling the user's touch input`

All the classes and scripts are well described and commented out in this project to further help you with how everything works.

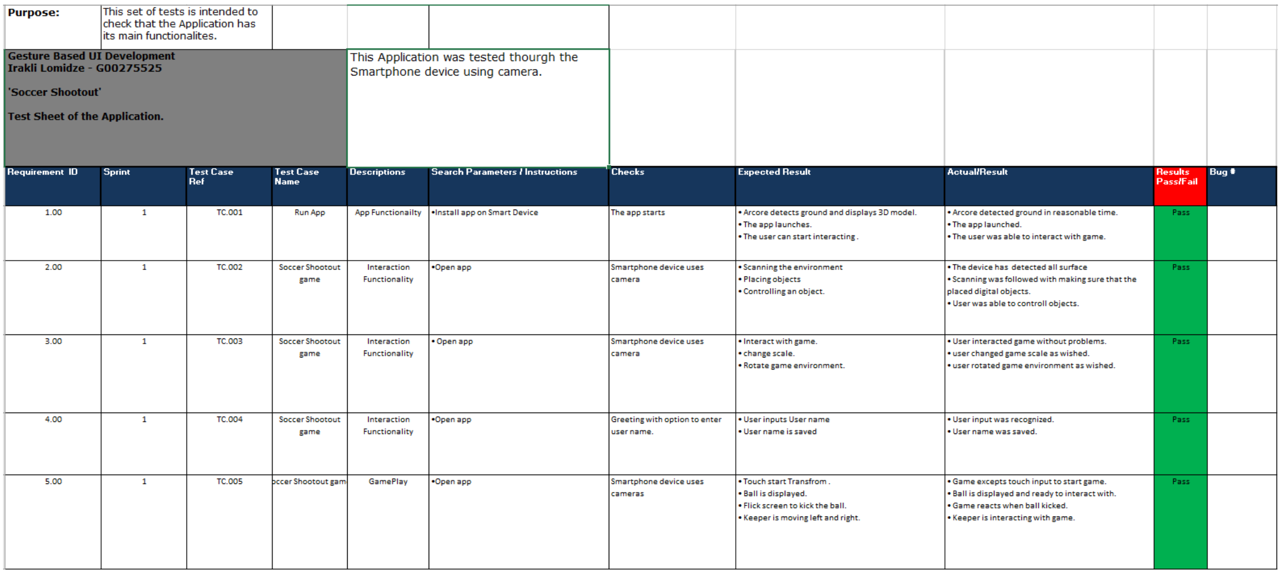
Testing

Tests ranged from looking at UI gesture interaction and game functions to fine tuning the overall performance of the application. Testing provided us with valuable insight and allowed us to continuously integrate new features and components we would otherwise overlook which I'll discuss further.

Testing Approach

I naturally adopted a Continuous Integration style of testing. While I can't automate tests, automation isn't strictly a requirement of CI testing. Each code integration would be manually tested for its intended functionality and verified by me (*Due to circumstances we are in right now because of COVID, college studies are done online therefor it's hard to find the non-submitting project member)*, I found this to be surprisingly yielding in terms of results since having someone else look at and test your implementation would generally lead to hidden pitfalls and bugs that the submitting party wouldn't initially identify

User Acceptance Testing Results



Purpose: This set of tests is intended to check that the Application has its main functionalites.

Gesture Based UI Development
Irakli Lomidze - G00275525

'Soccer Shootout'

Test Sheet of the Application.

This Application was tested thourgh the Smartphone device using camera.

Requirement ID	Sprint	Test Case Ref	Test Case Name	Descriptions	Search Parameters / Instructions	Checks	Expected Result	Actual/Result	Results Pass/Fail	Bug #
1.00	1	TC.001	Run App	App Functionailty	•Install app on Smart Device	The app starts	• Arcore detects ground and displays 3D model. • The app launches. • The user can start interacting .	• Arcore detected ground in reasonable time. • The app launched. • The user was able to interact with game.	Pass	
2.00	1	TC.002	Soccer Shootout game	Interaction Functionality	•Open app	Smartphone device uses camera	• Scanning the environment • Placing objects • Controlling an object.	• The device has detected all surface • Scanning was followed with making sure that the placed digital objects. • User was able to controll objects.	Pass	
3.00	1	TC.003	Soccer Shootout game	Interaction Functionality	• Open app	Smartphone device uses camera	• Interact with game. • change scale. • Rotate game environment.	• User interacted game without problems. • user changed game scale as wished. • user rotated game environment as wished.	Pass	
4.00	1	TC.004	Soccer Shootout game	Interaction Functionality	•Open app	Greeting with option to enter user name.	• User inputs User name • User name is saved	• User input was recognized. • User name was saved.	Pass	
5.00	1	TC.005	occer Shootout gam	GamePlay	•Open app	Smartphone device uses cameras	• Touch start Transfrom . • Ball is displayed. • Flick screen to kick the ball. • Keeper is moving left and right.	• Game excepts touch input to start game. • Ball is displayed and ready to interact with. • Game reacts when ball kicked. • Keeper is interacting with game.	Pass	

Excel test file will be included in project folder.

Post-Development Testing

To ensure our application was ready for deployment we thoroughly tested and compared actual outcomes to our desired outcomes. Unfortunately, due to the current international and national situation I couldn't be picky about my selection of testers which meant my team of testers consisted of me as well as family members.

Conclusion

The project was very intimidating initially and slow to pick up pace, with that said once we began our research phase and began to integrate components and technologies then saw how they clicked everything became very practical.

The hardest part was figuring out how to make augmented reality work smoothly, at the beginning I was going to use Vuforia Framework but after doing intensive research I realised that ARCore had better ground detection system.

If I had more time to work on this project, I would make few improvements like adding music, and better menus with settings but There was not enough time to acquire one before the project's deadline.

Overall, this was fun project to work with. Augment reality is cool technology and this project gave me chance to gain bit of skill in this new and growing technology.

References

Youtube tutorials for AR:

https://www.youtube.com/watch?v=FGh7f-PaGQc

https://www.youtube.com/watch?v=KqzlGApWPEA

Google ARCore official documentation:

https://developers.google.com/ar/develop/unity-arf/quickstart-android

Scriptable object event system free package form Unity asset store:

https://assetstore.unity.com/packages/tools/gui/scriptable-object-variable-and-event-system-138883

GitHub:

https://github.com/ika25/ARproject