

Table League AR Report

Lukas Bonauer, Ahnaf Munir

February 2020

Abstract

This report discusses the work done for the development of the mobile application Table League AR. This app is an augmented reality based Android game which is a combination of traditional tower defense and football video games. In this competitive multiplayer game two players try to use multiple balls to score goals on the opponent's side while defending their own goal by building towers and by collecting unique powers. The entire game environment is projected onto the real world through the two mobile phones of the players.

Contents

1	Introduction	3
2	Gameplay Overview	3
2.1	Main menu	3
2.2	Setting up the game	3
2.3	Basic Controls	4
2.4	Towers	4
2.4.1	Push Tower	5
2.4.2	Pull Tower	5
2.4.3	Barrier	6
2.4.4	Capture Tower	6
2.5	Powerups	6
2.5.1	Freeze	6
2.5.2	Tower Jam	6
2.5.3	Grappling Hook	7
2.6	Game Over	7
3	Technical implementation	7
3.1	Networking	8
3.2	Tracking	9
4	Player feedback and changes over time	9
5	Conclusion	10
A	User Feedback	11

1 Introduction

Augmented Reality (AR) has become one of the biggest technology trends in the last few years and its popularity is continuously growing. AR provides an interactive experience for the users by enhancing the real-world environment with digital augmentations overlaid on top of it. This technology is applied in diverse range of fields including tourism, navigation, archaeology, architecture, military, commerce etc. In recent times it has also entered the video game industry. Especially since the release of the mobile game Pokémon GO in 2016, AR-based video games have become quite popular.

Table League AR uses latest AR technologies to develop a fun, interactive multiplayer video game that encourages players to move around while playing the game. The game takes inspirations from classic tower defense and football games. In tower defense games players are usually responsible of a territory or some possessions which they must defend from enemy attacks. This is usually accomplished by building defensive structures such as walls and towers. The towers typically have some special feature like shooting cannonball, lasers, arrows etc. The players are usually also given unique powers which can be freezing time, disabling enemies attack etc. Like traditional football game, our game has two players trying to score goals on the opponent's side. They also must defend their own goal by building defensive towers and using different types of powers. Our project was developed using Unity. The AR aspect of the game is implemented using a combination of Vuforia and ARCore while multiplayer is integrated into the project using Mirror Networking, an open source Networking API for Unity.

The rest of the report is structured as follows: Section 2 gives an overview of the gameplay. The technical aspects of the project are discussed in section 3. Section 4 provides a detailed look into the incremental updates the project went through using the user feedback we collected. Section 5 adds some concluding remarks and discusses future direction of the project.

2 Gameplay Overview

2.1 Main menu

Upon launching the app, the players are greeted by the main menu. From here they can choose to play either a singleplayer or a multiplayer match. If the players wish to play a multiplayer game and they are not the host, then they must first insert the IP address of the host in the options menu. For both type of games, the player (only the host player in multiplayer matches) must choose the number of the balls and duration of the game before starting the match.

2.2 Setting up the game

Before beginning the match, the player (only the host player in multiplayer matches) must set up the position and dimension of the field to play in. This is

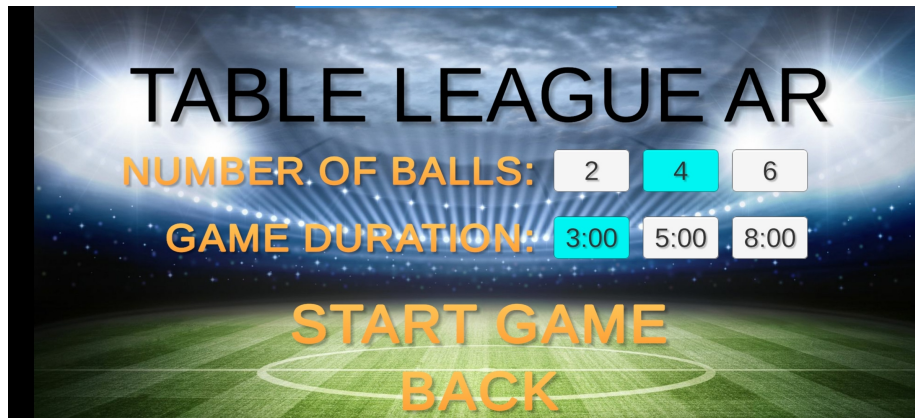


Figure 1: Game Configuration Menu

done by first placing a physical marker where the player wants the center of the field to be at and focusing on the marker through the device camera. When the marker is focused properly the app will prompt the player to set the width of the playing field. This can be done by moving the device horizontally to make sure that the white lines visible on the screen are aligned with the edges of the surface that is used as the playing field. After this step is finished, the player will be able to see the playing field through their device.

In case of multiplayer games, the non-host player only needs to focus on the marker and the field will already be aligned to the surface based on the specifications set by the host player.

2.3 Basic Controls

After the game has been set up, each player is responsible for two separate goal posts on their side. A certain number of balls spawn in front of them based on the specifications set before the game. The main objective of the game is to shoot these balls into the opponent's goal post to score points. The players can shoot the balls by touching them on their device screen. The direction and power of the shot will depend on the angle and distance of the device from the ball. The closer the player's device is to the ball, the faster the shot will be. This encourages players to physically move around the table in order to take proper aim and shoot balls with more force.

Each player can only interact with balls on their own side of the field. This stops players from interfering with each other in the real world.

2.4 Towers

The players can make it easier to defend their goalposts by building four different types of towers. Towers can only be built on a player's own side of the field and

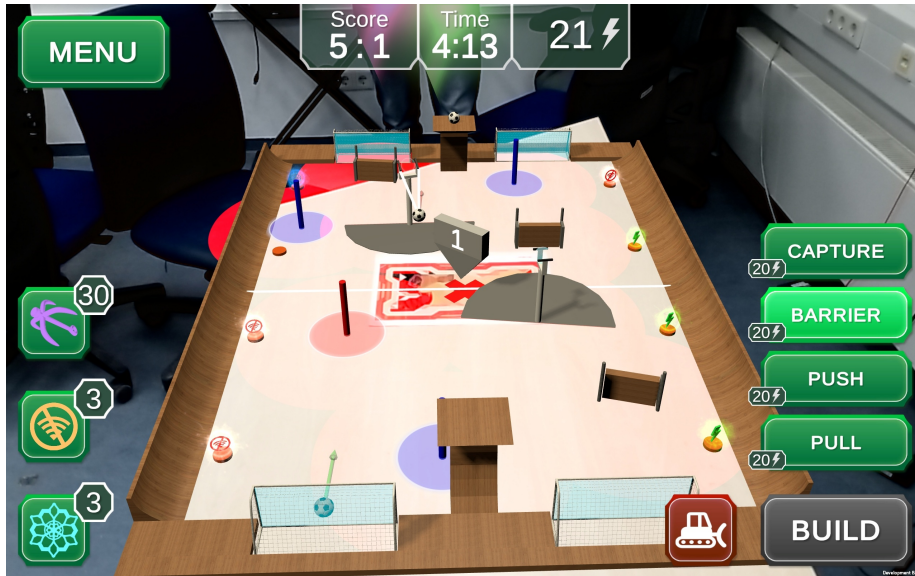


Figure 2: In-game view from a player's perspective

have to have a minimum distance to other towers on the field.

Each tower requires a certain amount of resources to build. These resources spawn on both sides of the field at specific spawn locations. Collecting one of these resources give the player 10 resource points, which they can then spend to build towers.

Players are also able to remove any tower they have previously built if they change their mind about how to best use their limited space. A tower can be destroyed by selecting the destroy tool and then pressing on the tower for a short amount of time. Both the tower building and destroying functionalities can be accessed from the 'BUILD' button at the bottom right corner of the screen. The effects of the four towers are briefly explained below.

2.4.1 Push Tower

The push tower pushes any ball in its range away from its center. It costs 20 resource points to build. The tower also has a charge. Each ball interaction reduces the charge of the tower. The tower cannot interact with any ball if its charge is empty. It starts to recharge by itself as soon as it is not fully charged, but only if there is no ball in its range.

2.4.2 Pull Tower

The pull tower also requires 20 resource points to build. The tower pulls any ball in its range towards its center. Similar to the push tower, the pull tower has a charge which is reduced at each ball interaction.

2.4.3 Barrier

The barrier can be built with 30 resource points. It stops any ball coming towards it. However, when a ball hits the barrier it breaks down and any subsequent ball approaching the barrier is not stopped anymore. The barrier becomes active again after a few seconds.

2.4.4 Capture Tower

The capture tower can capture a ball if it stays within the tower's range for a specific amount of time. When a ball is captured it is dragged to the top of the tower and does not move anymore. Only the owner of the tower is now able to shoot the ball captured by the tower. However, until the captured ball is removed, additional balls can move by the tower unhindered. Building this tower costs 30 resource points.

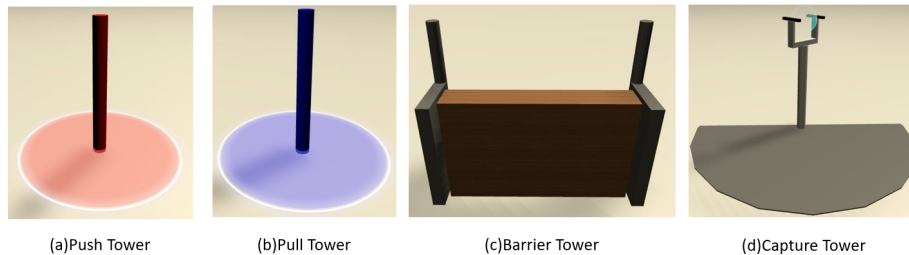


Figure 3: The four different types of tower

2.5 Powerups

There are three types of powerups that spawn at the same spawn locations as the resources. Collecting the powers rewards the player with unique abilities which can be used strategically during the match. The powerups can be activated by pressing the corresponding button on the bottom left corner of the screen. The three different powerups are discussed below:

2.5.1 Freeze

The freeze powerup freezes all balls at their current locations for 5 seconds. It prevents both players from interacting with them for a certain period, giving time to improve their defenses and prepare their ultimate strike.

2.5.2 Tower Jam

This powerup jams all of the opponent's towers for 10 seconds. While a tower is jammed it cannot interact with balls at all, allowing players to break through even the best opponent's defenses.

2.5.3 Grappling Hook

When the grappling hook is used it grabs all balls in the field and brings them to the location where the player's device is located in the field. This allows setting up a targeted attack from a strategic position. It is also an option to regain control of forgotten balls which have been lying on the opponent's side for a while.



Figure 4: The three different power ups

The grappling hook can be countered by the freeze powerup.

2.6 Game Over

At the beginning of the game the player specifies the duration of the match (3, 5 or 8 minutes). When this time runs out, the match ends. After the end of the match the players can see the end game menu, where the result of the match is shown. Similar to a typical football game, the players scoring the most goal wins. Draws are possible. The end game menu also provides the players with some in-game statistics about both players. This includes the number of times the ball was hit, number of towers built, number of powerups used, distance travelled by the player and also the number of saves. Whenever a player is able to save a sure goal scoring opportunity of the opponent it is counted as a save. When the players are done reviewing the end game screen they can return back to main menu by pressing the 'END GAME' button.

3 Technical implementation

The game was built with Unity version 2019.2. It uses the Lightweight Render Pipeline in order to maximize performance on mobile devices with limited hardware.



Figure 5: Players statistics shown at the end of the game.

3.1 Networking

The game requires the two mobile devices to be connected through a local WiFi connection, and it works without any additional servers or infrastructure. One of the player’s devices acts as the host, while the second player connects by entering the host’s local IP address, which is displayed in the options menu. Additionally, any number of extra devices can be connected in spectator mode, allowing them to observe the game without participating. This can for example be used at DemoDay to show the ongoing game to spectators on a separate screen, without requiring them to look over the players’ shoulders to see what is going on.

In version 2019.1, Unity removed its high level networking API, leaving no working built-in replacement to build networked games at small scale. Fortunately, the open source networking library Mirror reimplements much of the old functionality to make sharing GameObject state between devices easy. It supports multiple low level transport layers, but for our small number of players and local connections the default transport based on TCP was sufficient.

The host player acts as both server and local client, while the non-host player is connected only as a client. All player input, such as shooting a ball or placing a tower is sent to the server as a command and handled there. All physics updates are handled by the server, only the balls’ positions and discrete events (such as ”the barrier has been knocked over”) have to be sent back to the client.

Game sounds are also triggered exclusively by the server. Since players are standing close together and can hear the sounds from each other’s devices, most sounds are only played on the device that is closer to the source. For example, the audience cheers when a goal is scored, but only on the defending player’s de-

vice. This makes it easier to distinguish where on the field something happened, since true spatial audio is not possible through only a phone’s speaker.

3.2 Tracking

AR games require accurate tracking of the camera’s pose in the real world and a way to align virtual content to real geometry. Table League AR uses a flat surface, ideally a table, as its playing area and a single printed marker placed on the table to define the center of the playing field. This marker also serves as the zero point in the global coordinate system. For multiplayer, this choice allows all devices to project the game to the same real position without having to explicitly share their tracking state. Only the desired playing field’s size – which is calibrated manually by the host player at the beginning of a game – is transmitted over the network when the other device connects.

To implement tracking we use Vuforia, a commercial library that is now integrated into Unity. It can recognize and track objects of various types. This game uses a ”VUMark”, a more advanced version of an Aruco tracker with customizable design. The Vuforia API provides callbacks when it detects and loses a marker and then automatically calculates the camera’s position relative to the marker. However, this only works while the marker is visible to the camera. This game’s fast-paced mechanics frequently lead to player’s not looking at the center of the table or from a too low angle for the marker tracking alone to work reliably.

To mitigate this, we enabled Vuforia’s extended device tracker, which applies sensor fusion with the device’s IMU sensor and other visual information from the camera. This works either through Google’s ARCore system – if supported on the device – or by visual SLAM algorithms. With this feature enabled, tracking is very reliable over the entire duration of the game, even during fast camera movements, provided that ARCore is available and lighting is sufficient. On older devices without ARCore or if the room is very dark, the projected playing field jumps around a lot when the tracker is not in good view.

4 Player feedback and changes over time

We conducted several tests with different players over the semester to gather feedback, which helped us discover and address usability problems, bugs and potential new features. For example, users commented that they did not have enough overview over the field and often forgot about balls lying in a corner of the field on their side — out of view. So we added indicators to the edge of the screen so players can always be aware where balls are, even while not looking at them.

There was also an issue where players joined the game before walking to their end of the table, so the game was unable to properly assign them to the correct side of the field. So players would sometimes end up standing on their opponent’s side, neither of them able to interact with anything. Now there is

a new step in the game setup where each player needs to press an additional "READY" button before being assigned their side.

Initially, we considered adding more complex features, such as having to manually recharge your towers after their energy was used up or making powerups focus only a particular ball which you have to click on. However, based on feedback and observing new players trying the game, we found that there is already enough going on and players had a hard time switching between shooting balls, collecting resources and building towers all at the same time. Thus we focused on relaxing the player experience a bit, for example with the "freeze" powerup, by reducing the timers for collecting and building and by increasing the range at which players can interact with the world.

A more detailed collection of feedback and observations from various playtests is given in appendix A.

5 Conclusion

We developed an Android based multiplayer game that takes inspiration from classic tower defense and football games. The inclusion of augmented reality creates an engaging and immersive experience for the players. The gameplay was designed in such a way that encourages the players to move around the playing field and also develop strategies over multiples plays of the game. While the current build of our game is quite stable, we believe it can be improved further. The feedback of our games suggests that some of the players find the learning curve of the game to be a bit steep. This issue might be solved by adding a tutorial to the game to familiarize the players with the controls before playing a competitive match. We also plan on improving the networking aspect of the game: Players should be able to automatically discover open games in the local network instead of having to manually input the IP of the host.

A User Feedback

2020-02-04 DemoDay

- Overall quite positive feedback!
- People covered tablet camera a lot with their fingers
- Red tower can be confusing looks like it cannot be built
- Inputs and features not self-explanatory
- Tracking issues (probably due to low lighting)
- Tablet crashes sometimes, random reboots
- Non-gamers: overwhelmed with fast pace learning curve

2020-01-18 playtest with Felix and Timo

- The playing area is too large
- Maybe make side wall a round “half-pipe” instead of a straight wall for better bounce physics
- Visual lines in front of goal could make it easier to locate
- Suggestion to freeze balls only on own side
- Powerup cooldowns seem useful

2020-01-16 playtest with Peter

- Request for display where you can(not) build towers
- Close menu when tapping empty part of screen
- Entering build mode by accident too much / without noticing
- Unbalanced / too random spawning of powerups
- Single-touch vs multi-touch on tablet vs phone
- Powerup collection time was perceived as too high
- Need sound when freeze is over, hard to notice
- Move score (rest probably) to top of the screen
- Bug: trying to build tower by clicking on other tower is broken

2020-01-14 playtest with Vivi friend

- Not enough “green” (tower resources) in the beginning, too much in the end

- “Tower jammer” disables towers for too short
- Barrier + capture tower overpowered compared to magnetic ones, suggestion to adjust build costs accordingly
- Easier to play with multi-touch on phone than on tablet
- Tower destruction is not very intuitive (button toggle state not obvious enough, having to hold down is annoying)