

ARShips (Working Title) Games Design Document

Overview

ARShips will use a combination of augmented reality, Watson AI services and possibly some Internet of Things components to create a new and modern twist on the classic game. The idea is to create a portable and networked board game so two (or more) players can play the same game, even if they're not playing on the same board. I will be using the classic game of Battleships as a base for the board game which makes for a very familiar and simple game for people to pick up and play, and is also suitable for a large range of ages, attracting a large audience.

The addition of AR technology into this project will create a lot of different possibilities in terms of new gameplay features, such as upgradeable ships and powerups, and will also allow the game to become a lot more immersive.

The users will primarily play the game through a mobile device, as this will hold all the logic that controls the game, the networking capabilities and also display the augmented graphics.

Platform

The game will be created inside the Unity game engine, using the Watson SDK to handle the Watson service interaction. Android will be the primary platform I will publish to test on, however it will be possible to port it over to iOS with little to no hassle due to it being created with Unity.

Board Layout

A simplified version of the game board can be seen below. It is a grid with two sides, one for your ally ships and one to view your enemy's. The grid cells will each have a specific target image on them, however at the moment it will just be the grid cell name, with an 'a' or 'e' depending on whether it's an ally or enemy cell.

<i>Enemy</i>								
<i>A1e</i>	<i>A2e</i>							
<i>B1e</i>								
				<i>D5e</i>				
<i>A1a</i>	<i>A2a</i>	<i>A3a</i>						
<i>B1a</i>	<i>B2a</i>							
<i>You</i>								

Each cell can be augmented separately as well as the underlying board, meaning there will be a lot of flexibility in terms of visuals. As well as this, due to the game being primarily viewed through a mobile device, there will be the ability for each player to view the same board but see their own point of view.

After doing some research, one of the key features I would like to stick to is portability. This is very important in AR apps as you want to be able to play them anywhere, and especially with a board game, the users would not want to have to remember to carry a big board around whenever they want to play. To combat this, I have thought that instead of having electrical components in the board, just have the user print out a paper version that has all of the AR targets needed for the game to function. This way the user can carry the paper version wherever they go as it is easily foldable, or if they lose/break it just print another one out.

Player Input

Each player will control the game using the microphone on their mobile device, with Watson's Speech-To-Text service handling each voice command and converting it into readable data. An example of a voice command could be "Fire at A3" to execute an attack on cell A3. These commands will be available

for the player to see in an instructions screen at any time during gameplay, so there is never any confusion.

Voice will be the primary control method however an alternative will have to be created in case the user doesn't have access to a microphone on their device. Whenever a player enters an input, using either method, a dialog box will be displayed on the user's phone to make sure the correct move is being played.

Game World

For now, the graphics displayed on the mobile device will be very basic, with no animation, however the distinction between ally and enemy ships will be made obvious. Once all the necessary features are implemented and tested work will be done to make everything look a lot nicer. I would like the look and feel to move away from the classic battleships look to make this version stand out.

User Interface

There will be a number of different screens throughout the app including:

- Home screen
- Options screen
- Multiplayer connection screen
- Game screen

Each screen will be very basic in design for now, just displaying the necessary buttons and information for the player to get to the gameplay. Ideally, I would also like to add the ability to navigate the menu through voice commands as well.

Online functionality

For now, I am planning to use Unity's network manager to take care of the networking between two players as this is fairly trivial to set up. This will allow for a "room" to be set up by one player (the host) and the other player will be able to type in this room name and join.