

EECS 495 Software for Accessibility Requirements Document Bubba-4

System Description

System Purpose:

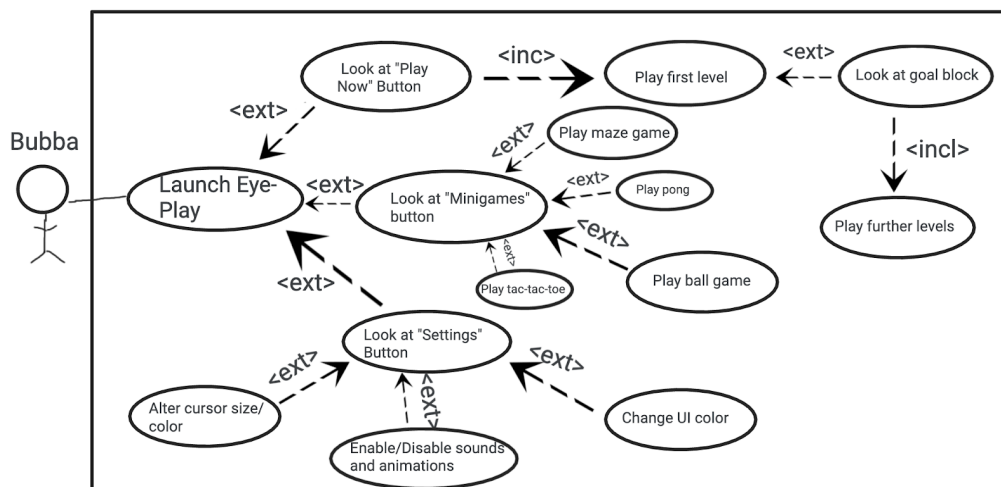
Eye-tracking is a novel technology that empowers individuals with motor disabilities to interact with software applications that would otherwise be inaccessible. However, eye-tracking can be difficult to learn, so an application is needed that will train users to master the precision of their eye movements.

Project Scope:

Fundamentally, this project is built teaching users the ability to use eye tracking. The base functionality encompasses the main game. The main game will consist of multiple blocks placed around the screen. Of these blocks, one block will be specially designed, perhaps with a symbol, and the user's goal will be to set their gaze on this block. This app provides eye tracking functionality to check if the user is looking at the goal block. The user will advance if they successfully set their gaze on the goal block. In order to help the user practice their eye movement, levels will be designed around different eye movements. For example, the first level may have the goal block on the right side of the screen so that the user can practice lateral eye movement. The second level may have the goal block on the bottom of the screen so the user can practice vertical eye movement.

In terms of extended functionality with additional time, the app will offer multiple minigames and a means to customize the UI to the user's liking. The minigames are designed with the intention of the user to make use of their eye tracking abilities after they have trained in the main game. For example, one minigame may be tic-tac-toe, where the user plays with a friend and uses their eyes to move their symbol onto the board. Another minigame may be a maze where the user uses their eyes to drag a ball to the center of the maze. Additionally, we hope to offer a settings menu where users can customize the app to their liking. For example, the user may have a preference for what color the UI is, black/white or yellow/green. Perhaps the user may want to make their cursor a different size. Ideally, this will make the app more personalized for each user.

Figure I: Eye-Play Use Case Diagram



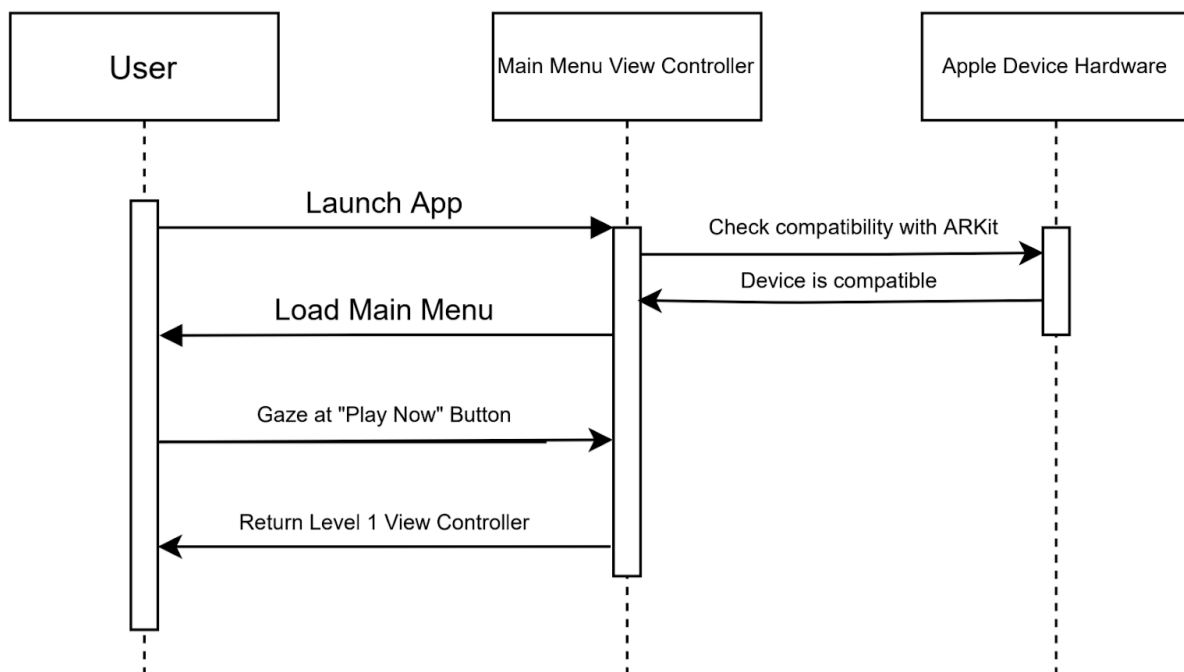
The use case diagram begins with the user, Bubba, launching the Eye-Play app. From here, he has three options. He can look at any of the “Settings”, “Play Now”, or “Minigames” buttons to bring up a new interface. If he gazes at the “Minigames” button, he will be transferred to another screen where he will have the option to pick from an assortment of minigames to play – The maze game, the ball game, pong, and tic-tac-toe. If he looks at the “Settings” button, he will have the option to change the UI color from the default black/white, enable or disable the sounds and animations, or alter the color or size of his cursor. Finally, Bubba may otherwise gaze at the “Play Now” button. At this point, he will be transitioned to the first level. At the first level, he will have the option to look at the goal block. If he looks at the goal block, he will proceed to further levels.

Project Flow:

The two scenarios presented below pertain to expected usage and unexpected usage. In both scenarios, we are assuming the user is someone with limited motor function that is interested in using the app for the first time to begin training their eye tracking abilities. For the expected usage scenario, we are assuming the user is using an Apple device that supports ARKit. For the unexpected usage scenario, we are assuming the user is using an Apple device that does not support ARKit.

The outcome that will be presented in each scenario differs. In the expected usage scenario, the user will be able to successfully proceed to the first level of the game. However, this will not be the outcome for the unexpected usage scenario. In the case of unexpected usage, the app will trigger an exception informing the user that their device does not support the eye tracking software necessary to use the app, and the app will forcefully shut down.

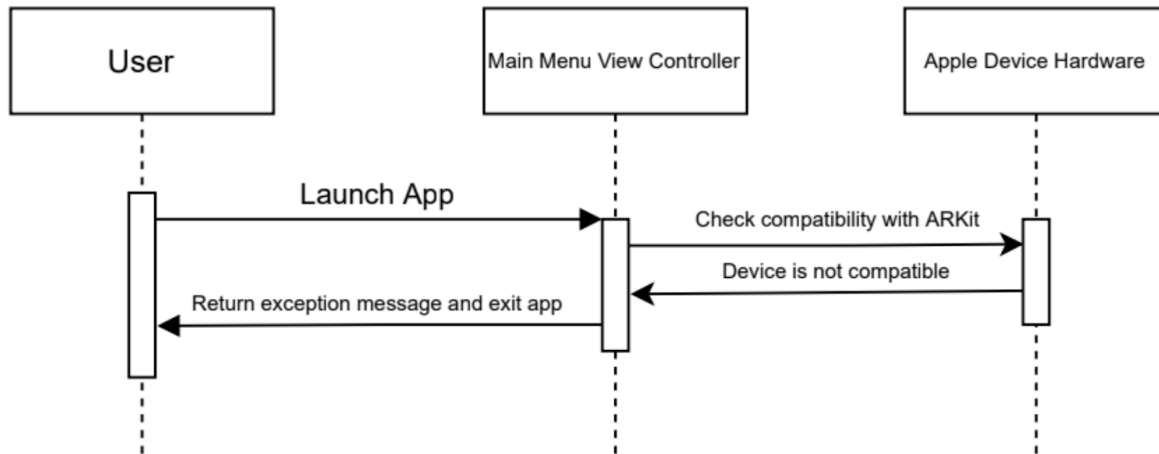
Figure II: Expected Usage Sequence Diagram



In this scenario, there are three different partitions for the interactions – the user, the main menu view controller, and the Apple device hardware. The sequence begins with the user launching the

app. When the app is launched, the main menu begins loading. While the main menu loads, the view controller performs a check with the Apple device hardware to find out whether or not the device is compatible with the ARKit software. Once the device verifies that it is compatible with ARKit, the main menu finishes loading and is available for the user to interact with. In this scenario, the user decides they would like to play from the beginning of the game, so they gaze at the “Play Now” button. Consequently, the main menu view controller returns the Level 1 view controller for the user to begin their training.

Figure III: Unexpected Usage Sequence Diagram



In this scenario, there are once again three different partitions for the interactions – the user, the main menu view controller, and the Apple device hardware. As with the expected usage, the scenario begins with the user launching the app and the main menu beginning to load. While the main menu loads, the view controller performs a check with the Apple device hardware to find out if the hardware is compatible with the ARKit. In this case, the device informs the view controller that it is not compatible. As a result, the main menu view controller returns an exception to the user informing them that this device is not compatible with the app. Then, the app will exit automatically.

Enumeration of Requirements:

number: LVL-1	name: Level 1	func or non: func	priority: α
description: The first level of the main game will consist of two blocks on opposite sides of the screen – one block to the left and one block to the right. The goal block will have a special symbol to indicate the user should gaze there.			
source: This was the original, main pitch of the app. This level will help Bubba practice lateral eye movement.			
notes: N/A			

number: LVL-2	name: Level Transition Animations	func or non: func	priority: β
description: After the user successfully gazes on the goal block, the level will end. Before moving onto the next level, a transition will occur. During this transition, fun and stimulating animations will play to provide positive feedback to the user and encourage them to continue playing.			
source: We really wanted a way for Bubba to feel encouraged and have fun as he played the game. His parents mentioned how much he loved fun animations in movies, so we felt it was important to incorporate this into the level transitions to keep him engaged and focused on advancing.			
notes: N/A			

number: LVL-3	name: Level Transition Sounds	func or non: func	priority: β
description: Upon completing a level or minigame, a transition sound will be played, in addition to a transition animation. Options will be available to toggle the sound animations.			
source: Bubba's parents emphasized how much Bubba enjoys the sounds of his elephant toy and the sounds in movies/shows he watches. Consequently, we felt that transition sound effects upon successful completion of a level would be a great positive feedback mechanism to encourage him. Additionally, The sounds will help us follow the principles of good game design as a good playable game should be an audio-visual experience. They will also help our user understand what is happening on the app in another way apart from the visual signaling.			
notes: N/A			

number: SET-1	name: Cursor Color and Size	func or non: func	priority: β
description: The ability to modify the size and color of the cursor that tracks eye movement on the settings page.			
source: Bubba's dad mentioned that Bubba is sensitive to only a subset of colors and would prefer a larger cursor size, so it's important that the size and color of the cursor can be altered if other users have similar preferences.			
notes: The default size will be large and the default color will be red, as those are Bubba's preferences.			

number: SET-2	name: Background and Block Aesthetic	func or non: func	priority: Ω
description: The ability to modify the color of the app background and blocks on the settings page.			
source: Bubba's dad mentioned that Bubba is sensitive to only a subset of colors. Particularly, he mentioned black and white make for good backgrounds. However, he mentioned that other children may have different needs, so it should be modifiable.			
notes: The default colors will be black, white, and gray, as those are Bubba's preferences.			

number: UX-1	name: 100% Hands Free	func or non: non	priority: α
description: The app needs to be completely hands free. That is, the user should be able to take full control and use it with just their eyes once it is launched.			
source: The whole point of this app is to train users in eye tracking so that they can use software that normally relies on motor function. In this spirit, it's important this app can be used solely with eye tracking without need for motor functionality.			
notes: N/A			

number: UX-2	name: Eye Tracking Should Never Fail	func or non: non	priority: α
description: The eye tracking needs to be as precise as possible and should never fail. If eye tracking is not supported on the hardware, the app must immediately exit.			
source: The whole point of this app is to train someone in eye tracking. If the eye tracking capability of this software is faulty or prone to failure, then that could have very adverse effects on our user's learning. Consequently, the eye tracking must be foolproof.			
notes: Extensive testing on multiple hardware devices must be conducted to ensure that the eye tracking is sound.			

number: UX-3	name: The App Should Be Replayable	func or non: non	priority: ∞
description: The app should be designed around frequent use by the user so that they may continue to improve the precision of their eye tracking skills. The user should feel compelled to return to the app and not feel bored or unengaged after their first use.			
source: We want this app to be a powerful resource for teaching users with disability how to leverage their eyes for accessibility. If this app does not capture the engagement of our users and they are not compelled to return, it is very unlikely they will successfully learn to improve their eye tracking abilities, which will defeat the purpose of this app.			
notes: N/A			

number: UX-4	name: App Should Not Induce Eye Strain or Nausea	func or non: non	priority: ∞
description: Our app will feature a harmonious color and layout scheme that will allow one to use it for a decent amount of time. Because the app may be used for a prolonged period of time, it will seek to avoid eye strain or nausea while the user keeps their eyes on it..			
source: A group member complained that a previous app they used was causing their eyes to water and giving them a headache due to a bug that was causing intermittent flashes to occur. We would like to avoid a similar experience when deploying this app.			
notes: N/A			

number: UX-5	name: The UI Should Be Intuitive	func or non: non	priority: ∞
description: Creating a good flow of visual elements from screen to screen is important for the overall functioning of the app. This means that no specific screen of the app should feel awkward or out of place. Intuitively without thinking our user should be able to locate everything and quickly get to the place they need to get too. Additionally colors and iconography should be consistent with other mobile applications so it is easy to navigate and simple to use.			
source: Many of our group members discussed how they've deleted app or programs that had a very difficult, unintuitive UI. We don't want to alienate our users with a poor UI, so following good design principles of UI design or interface design is important because it affects the overall usability and functioning of the final product. Keeping a good flow and harmony of visual elements will maintain commitment when the user is using the app.			
notes: N/A			

number: UX-6	name: The App Should Not Require a Specific Physical Orientation To Function Properly	func or non: non	priority: ∞
description: The app should not have to be mounted in a specific way to function properly. The user should have the freedom to orient their display in a manner that is most comfortable to them without compromising functionality.			
source: We wanted to design our app in such a way that it is physically comfortable to use. This means that users should have freedom over how they want to have their phone or tablet oriented when they use it. Whether that be flat on a table, mounted on a tripod, or angled on the palm of their hand, we want to ensure the user isn't restricted.			
notes: N/A			

number: UX-7	name: The App Should Be Fun	func or non: non	priority: ∞
description: The app needs to be an enjoyable experience from the first time the user launches it. Enjoyment should not be an "acquired taste". The games should be implemented in a way that they are fun to the users.			
source: One of our teams main objectives is to create a fun experience that can also encourage the learning process of using eye tracking technology. If our games are fun then that can encourage the user to better learn and interact with technology in general.			
notes: This is different from the replayable non-functional requirement. The idea that it is fun should apply from the very first time the user uses it, even if they never use it again.			

number: UX-8	name: The App Should be a Teaching Tool	func or non: non	priority: ∞
description: Our app will provide multiple levels and games that are targeted for a user who is a young learner. Ultimately, it will have to teach the user without them needing to be consciously aware that they are learning. Their eye movement precision should naturally develop and improve while they play. This can be achieved by training in the main game or even utilizing the minigames. Each minigame is targeted to a potential learning style for our user. If the user enjoys socializing, a social learner, the ping pong game would allow a 2 person interaction. If the user enjoys puzzles or games revolving around logic, the tic-tac-toe and maze minigames will be what they can use.			
source: This app was originally conceived under the idea to teach Bubba how to interact with eye tracking software, as his parents mentioned that was something that he was progressing with, but had to pause when Covid hit. Essentially, the whole premise of this app is to teach.			
notes: N/A			

number: UX-9	name: Multiple People Should Be Able to Play at Once	func or non: non	priority: ∞
description: The app needs to have the ability for the user to play with another person next to them. This other user should be able to play without needing to eye track.			
source: Some of the minigames we've come up with – tic-tac-toe, for example – are built on the idea of Bubba being able to play with his sister, Lea. Bubba's dad mentioned this was a fantastic idea and that Bubba would really enjoy it if other people could play with him. He added that it wouldn't be necessary for his sister to use eye tracking and it would be fine if she could use her fingers or tell Bubba where to place her mark for tic-tac-toe.			
notes: N/A			

number: MG-1	name: Follow the Ball Minigame	func or non: func	priority: α
description: A minigame in which the user moves the cursor on to a ball. Once the ball has been selected, it moves to a random spot and the user must continue moving their cursor on to the ball. The game can be played indefinitely.			
source: We realized that the main game of selecting blocks may get boring after a while, so adding these dynamic minigames is important for Bubba's long term engagement with the app.			

notes: This will be the first minigame incorporated into the app. Originally, all minigames were intended for the omega release, but this minigame was created during our integration of the ARKit API to test the eye tracking, so it makes sense to add it in during alpha.

number: MG-2	name: Maze Game	func or non: func	priority: Ω
description: A simple maze game, our player represented with a cursor, has to go through a maze to reach a destination. We aim to make it a fun logical puzzle that is also simple enough for a child to solve as our game is made for young kids learning and development.			
source: We wanted to provide a way for our user to carefully practice following a certain path, while also practicing the ability to dynamically change course in case they run into a dead end. We felt that a maze did a great job of achieving this. Additionally, we wanted to include a maze minigame because it is part of all of the classic game packs that were on the flip phones in the early 2000s. This nostalgia for simple fun games will inform our design and implementation of this minigame.			
notes: Currently, we are undecided on whether we will just have one maze or multiple.			

number: MG-3	name: Tic-Tac-Toe	func or non: func	priority: Ω
description: A simple tic-tac-toe game where player 1 can enter in their characters with their eyes. Player 2 can mark their characters manually or tell player 1 how they would like to execute their turn and have player 1 place their character for them. The game will reset once 9 marks have been placed and there is no winner or someone gets a 3 in a row.			
source: We wanted to include a simple multiplayer game appropriate for young children. Additionally this game allows Bubba to play with his sister Lea, which is something the Overby family responded very well to.			
notes: N/A			

number: MG-4	name: Pong game	func or non: func	priority: Ω
description: This minigame will be an incarnation of the classic pong game. It is played by trying to get the ball behind the other paddle. The left paddle will automatically be controlled by the user and the rightmost paddle will have the option of being either a computer player or a human one. The score will be kept in a toolbar above. The ball will randomly deploy to either side and make a sound as it collides with the paddle.			
source: We want to create a fun multiplayer game to help our user socialize while playing a computer/tablet game. Pong is another classic game to add to our edition of classic simple software from the early days of computing.			
notes: N/A			

number: PLTFRM -1	name: Needs to Run on as Many Apple Devices as Possible	func or non: non	priority: ∞
description: The app needs to run on as many Apple devices as possible. That is, we shouldn't develop with only one specific iPad or iPhone model in mind. Any device with the underlying ARKit compatible technology should be able to run this app without issue.			
source: Out of our group members, only 1 person has an iPad and everyone else has different iPhone models. In order for different group members to test the app, all of us need to run it on our different hardware. Consequently, this app needs to function on many different Apple devices.			
notes: Only Apple devices with FaceID functionality (iPad Pro, iPhone X and up) are compatible with the ARKit.			

number: MU-1	name: Background Music	func or non: func	priority: β
description: Our app will have music built in. The music selected should be brain-off video game music that encourages focus and meditation. The options menu on the title screen will control the			

music for the app. Music may be altered in certain minigames to convey a sense of increasing pace.
source: Music is a component of many games, so we thought it was appropriate to include it in the background of this app. In general, our group members agreed that background music in games helps with the immersion, particularly if a level is getting increasingly difficult and the music increases in pace to reflect that.
notes: N/A

Prototype User Interface (UI)

Figure IV: Main Menu



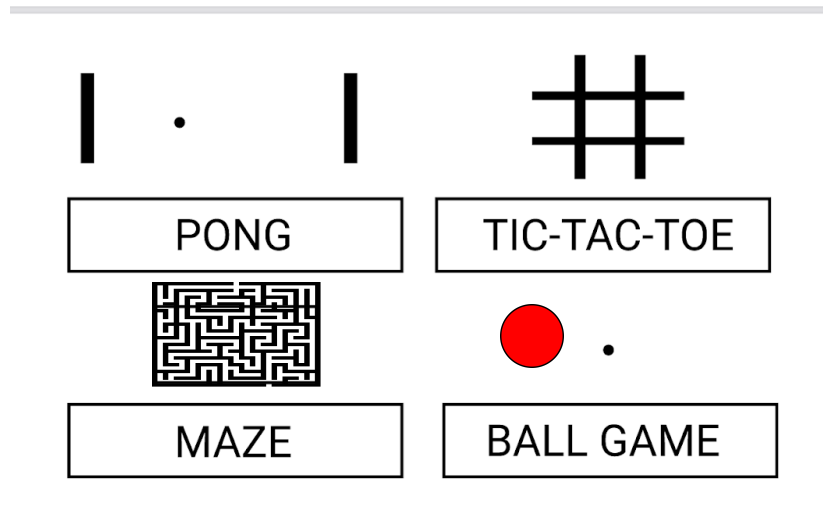
PLAY NOW

MINIGAMES

SETTINGS

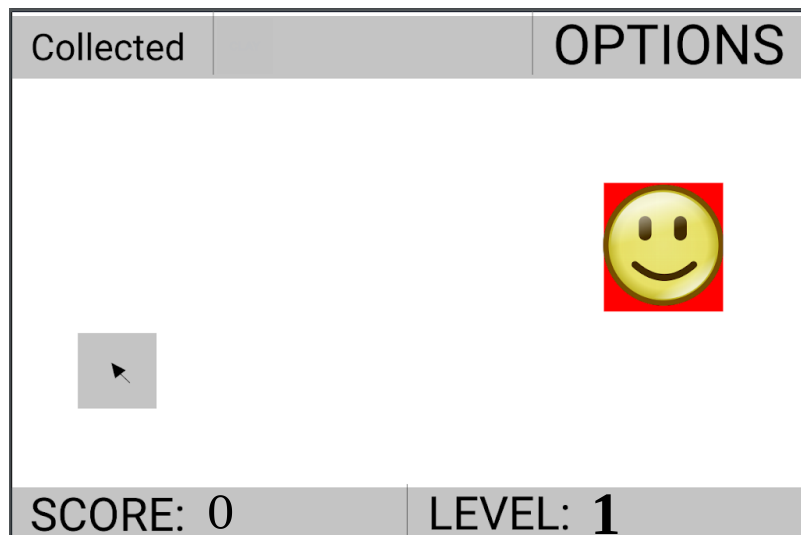
This view is for our main menu screen, it has 3 buttons that will take the user to other screens. The first button, the play now button, takes the user to the training screen where they can move a cursor and hit the blocks. The second button takes the user to the Minigame screens where they will select the minigame they want to play. The third button will take the user to the settings screen where they can alter configurations for the app, such as cursor color and size, music, etc.

Figure VI: Minigames Screen



This is the view for the minigame selection screen. The user will be brought to this screen from the Minigames button on the title screen. The various buttons will take the user to their respective minigames. The user may select a minigame by shifting their gaze to any of these buttons.

Figure VII: Main Game Level 1



This view is the first level for the main game. It is a simple view, where the user uses their eyes to move the cursor to the goal block (indicated with a smiley face). A gray distractor block is on the left. The view has bars that show what the user's current score is (based on how many blocks they have collected), the level they are currently on, and a button to open an options menu. There are two ways to go to another view. The first way is by gazing on the goal block, which triggers an animation on collision to transition to another level. The second is if the user gazes on the options button, which will pull up a menu.

Environment Description

Development Environment:

- XCode will be used to write all the source code files in Swift and used to make the basic UI components through its storyboard functionality.
- Figma will be used to design graphics.
- iOS will be used to build and test the app.

Target Environment:

- iOS will be the target operating system for the app.

Software Packages and Tools:

- Apple's ARKit API will be used to abstract the eye tracking.

References

<https://app.diagrams.net/>

<https://developer.apple.com/documentation/arkit/arfaceanchor>

<https://developer.apple.com/ios/>

<https://developer.apple.com/xcode/>

<https://www.figma.com/>

<https://freesvg.org/the-maze>

<https://github.com/chrishoste/eye-tracking>

<https://jamboard.google.com/>

<https://www.raywenderlich.com/5491-ar-face-tracking-tutorial-for-ios-getting-started#toc-anchor-010>