MAIN BODY

ARe Framework and Interaction

Conceived as a real-life iteration of an interactive hologram system, my Project ARe drew inspiration from the visual aesthetics of Tony Stark's (Iron Man) holographic interface. While influenced by the intricate interaction depicted, it is crucial to acknowledge the existing technological limitations and my programming constraints in achieving comparable sophistication.

Leveraging existing technology and my knowledge, I aimed to design my project for users to interact with virtual content via finger gestures on a mobile screen. Consequently, I chose to incorporate the following finger gestures as the primary mode of interaction.

- Pinch scale – using the Thumb and Index finger to scale the virtual content on the screen

- Rotate – sliding the Index finger on the screen, either left to right to rotate the content in either direction

- Translate – move the content through three-dimensional space by placing the index finger on the virtual content and dragging it across the screen

- Display 1 – hide/unhide portions of the content or display hidden contents by double tapping on the screen

- Display 2 – Show annotations and additional information through touch-hold of the screen

By using a user-centred design principle as touched on by (Ref #17), the purpose was to eliminate any steep learning curve that would result from having first grasp how to navigate in Are by using gesture interactions that can be found on modern touch screen mobile devices.

Next, I wanted ARe to be a complementary learning tool to be used alongside physical resources like textbooks instead of a complete replacement for it. I opted to proceed with a marker-based AR system, where the digital content would be generated and placed on a marker – which would be found in the textbook. This would make the digital content placed alongside the content being studied to demonstrate its usefulness towards an enhanced user learning experience.

ARe: Planning, and Design

Planning

As part of my planning and design phase, I created a storyboard that would help portray the exact operation of my prototype. This would help me to visualize the key functionality at an early stage, enabling me to make any changes where necessary. From the layout, I went with an illustration of a single learner in their own learning space, to show that my project is designed to suit a self-learning environment.

A timeline was created breaking down key tasks into milestones on a per-week basis, with each week ending with a test build to evaluate prototype functionality and development progress.

This approach worked perfectly for me since I approached my project as a software development task. Employing the agile design method enabled me to proceed with development while testing new visuals or features after each weekly sprint (Ref #7). Furthermore, I needed a testing model that fed back into development for new iterations and further testing during production, which I achieved using Test-driven development (Ref #webpage here)

Design

Keeping user engagement and interest at the heart of my design, I decided to investigate appealing visuals for my digital content. Bearing in mind that I am developing for mobile, I needed a way to present my models at the best visual quality without compromising on device performance. I decided to go with low-poly models to limit the polycount and gradient texturing approach to aid performance while offering stunning visuals. Further optimization techniques (retopology through decimation) to maintain reasonable and functional polycount on downloaded models with high resolution.

Next, to give the users a real learning experience, I decided to choose a subject area that was quite a challenge while studying mathematics while offering a certain level of familiarity to anyone who would see it. For this, I chose trigonometry, mainly because most trigonometry problems would have an accompanying 2D image to visualise the problem. Feedback from informal peer review also enabled me to decide to proceed with a beating heart for the second subject. While the purpose of the visuals of the mathematics was to demonstrate effectiveness in visualising abstract concepts, the biology on the other hand was to test on user engagement and exploration, features I would also test in the mathematical problem as well.

To create a realistic learning experience for users, I adapted my marker designs. Instead of conventional 2D images on paddles as exemplified by (Ref #4), ARe marker was designed to be a textbook page with actual learning content combined with unique patterned images to trigger the digital content. Here, the user would enhance the learning experience complimented by ARe alongside the physical learning resource. The layout design would be done in Adobe Photoshop. More details will be covered in subsequent sections.

ARe: Development, Challenges and Solutions.

I began looking at the best tools I would employ to develop Are. While many online tools enable beginner developers and enthusiasts to quickly build an AR experience, I sought a system that would give me more control over building features as I wanted them. For this, I began exploring Unity as my development tool of choice. My next choice was then deciding on the suitable development framework that would best suit my project. I began experimenting with both AR foundation – Unity’s custom AR development SDK, and Vuforia AR SDK – an external AR development tool. I began experimenting to see what was available to me to achieve my goals. Both tools had their pros and cons. Firstly, setting up the AR foundation proved straightforward compared to the Vuforia which required initial setup on an external site before proceeding to Unity for another setup and integration. For image tracking, this was more direct in Vuforia which will rely on a database set of embedded images for tracking (part of the initial setup). XR interaction toolkit was the additional plugin that worked together with AR foundation – which provided me with robust user interaction, Vuforia on the other hand had no inbuilt interaction system. My only limitation with the AR foundation was figuring out how to implement the image tracker. With limited time on my hands, I began researching how I could combine both frameworks, taking key features from them to suit my needs. After a long time of trial and testing, I was able to get a working system. However, combining both Vuforia AR SDK and AR foundation generated a lot of errors and when the system seemed to work to a certain point, I noticed the system lagged a lot during development and testing, limiting the speed of my progress. With this occurring, I was forced to revise my methods and find a more suitable working alternative. Further research led me to a free Unity plugin that allowed user interactions that I needed called Lean Touch. On inspection, the plugin offered the necessary on-screen interactions. Having discovered this, I decided to work with Vuforia AR SDK, since it offered a straightforward approach to image tracking.

However, as development proceeded, I noticed that the ‘Lean touch Rotate Axis’ component required a two-finger interaction instead of one. Part of my design principles was to create a prototype that was easy to use and control. Seeing as the user would have to hold the device with one hand and interact with the other, I want an interaction that would require the user's index finger, except for the scaling functionality. To implement this exact function, I found tutorials that demonstrated how to achieve this using custom-written C# scripts. Combining this with the Leantouch Scale component I was able to achieve a rotation and scale operation. However, when applying the lean touch translate component a script conflict occurred. Both the translate and rotation functionality required a single finger interaction, this caused the 3D model to rotate and move at the same time. I aimed to make both operations independent of each other while maintaining both functions. To resolve this, I began exploring creative ways to design a solution. A solution I came up with was to initialise the movement of the object only when the user touched and moved it while assigning rotation sliding to any location on the screen. I also added a constraint in the translation C# script that would block any rotation once the translation action was happening.

To keep up with my proposed timeline, I started working with Cubes and a test marker in Vuforia to build the interactions and on-screen features and functionalities. The purpose of this was to ensure the main interactions were created and resolved while proceeding with the 3D production pipeline simultaneously. This approach allowed me to progress with both the key functions and the creation of 3D models for both mathematics and biology. Towards the end of my project, once the models were complete, I swapped out the cubes with the actual models.

Colour plays an important role in creating engaging visuals. When deciding on the visualisations for my chosen subject, I opted for an art style that bordered on the lines of play without being silly. My goal was to make learning fun and to achieve this by using appealing and playful styles in my design. By keeping the colours vibrant, I intended to immediately grab the users' attention once the application was used. By using a gradient texture approach, I would be able to swamp and flow between colour gradients without increasing file size.

To include the learning enhancement component to my design, I included an information board which held the trigonometry details and clues, formulas for solving the trigonometry problem. This would be accessed by placing the hands on the screen for some time to reveal this function. Also, I included a triangle image to show the unique different angles from the problem. This would be revealed by double tapping on the screen.

For Biology, I included animated arrows that showed the flow of blood from the left ventricle to the right ventricle. Annotations were also included to enable each user to know which component of the heart they were looking at. The heart was also animated to beat so that users could have a look at a real animated heart compared to the flat 2D image of the textbook.

For my presentation, I wanted the triggers to be unique. Vuforia image tracking system works by recognising unique contrasting patterns and assigning dots to them to make them recognisable to the system. I wanted my design to not just be geometric patterns without any connections to my work. So, I generated the default markers from (website comes here) and combined them with images that tied back to the content. For biology, I used an edited image of the human heart and for mathematics a 2D representation of the trigonometry problem. When combined with the final textbook design for both subjects, it produced a unique design that would work for my purposes.