3D Voronoi Portfolio B

Angus Robertson, Moise Guran, Nathaniel Petrache, Samuel Balson

1. Overview

The Client

The client is the University of Bristol's School of Mathematics. The School's Fry Building was recently refurbished, and part of this refurbishment involved the erection of a large Voronoi pattern sculpture opposite the entrance.

Dr Marton Balazs and Professor Peter Green represent the client. Dr Balazs is the IT-representative within the School and the main point of contact on the project. Professor Green prepared the initial calculations for the Voronoi sculpture and provided his expertise on Voronoi patterns.

Domain

The domain of the project is an educational mobile application, using augmented reality (AR) technology. The app was developed as an open-source solution and is available for download by visitors of the Fry Building.

AR apps have been developed for similar purposes such as in museums to bring an interactive element to their otherwise static or 2-dimensional displays.

Project

The sculpture on the Fry Building is a 2D slice of a 3-dimensional Voronoi tessellation, mathematically defined as a set of regions whose points are all closest to one of a set of points within the Fry Building.
The project's aim is to display the entire tessellation, along with the set of points, in augmented reality. The purpose of the solution is for the educational interest of all visitors of the Fry Building, and thus ease of use had to be a priority.

Vision for Project Solution

The solution is a mobile app that visitors can download by scanning a QR code on a pair of posters located in the Fry Building atrium and courtyard. These QR codes also double up as images to anchor the 3D model to so as to understand the relative position of the user to the sculpture. When the user opens the app, they will see the view from their device's camera on-screen. The app will detect the sculpture's position and orientation and display a 3D image of the Voronoi tessellation overlaid on the real view of the sculpture. As the user moves around the space, the 3D image will adjust its position and orientation accordingly. The app also has educational links and information regarding Voronoi tessellations and the sculpture.

2. Requirements

User Stakeholders

Visitors
A visitor is anyone who passes through the Fry Building and has the intention of using the app to view the 3D tessellation. This includes students, staff and open day visitors of the University.

Maintenance Team
This is anyone who is involved with maintaining the app after its completion. It could potentially be one or several of those team members currently involved in the development of the app. Alternatively a third party or even the client itself may be responsible for keeping the app functional and up-to-date. The maintenance team will be interested in a well-documented app that is clearly constructed. This will allow easier updates and fixes. During development the requirements evolved to include core functionality within the app, specifically to help the maintenance team add anchors.

Other Stakeholders

Passers-by
These are also visitors such as students, staff and open day visitors who are passing through the Fry Building. However, they do not have the intention of using the app and may be indirectly affected by its use. This could be due to app users unknowingly blocking their path while distracted by the app. The use of the app may cause bystanders concern or confusion if they feel that they are being filmed.

School of Mathematics/University of Bristol
The School of Mathematics is the client. The School is interested in the success of the app as a way of enhancing the experience of viewing the Fry Building's Voronoi sculpture. The app should help to showcase a part of the School and its success would reflect well on the client.

Development Team
The team is comprised of the five members stated at the top of this portfolio. Their involvement with the app includes development, testing, documentation and potentially maintenance. The development team intends to create an app that meets the client's needs and is effective for end users. This will reflect well on them for their future careers and success in the project is desirable since it contributes to their academic grade.

User Stories

Visitor
1. "As a visitor I want to be able to view the 3D Voronoi tessellation to better visualise and understand how the sculpture was designed."
2. "As a visitor I want to be able to learn more about Voronoi patterns, the mathematical relevance and the Fry Building's sculpture in particular."
3. "As a non-mathematical/computer science visitor I want to be able to learn about the Voronoi sculpture in an intuitive way and use the app without difficulty."

Maintenance team

4. "As a maintainer I want to be able to input the initial offset values of the CG model from each anchor within the app."
5. "As a maintainer I want to fine tune the offset values manually, using the augmented reality view to see updates in real time."
6. "As a maintainer I want to save the offset values so that I can refer to them after leaving the augmented reality view."

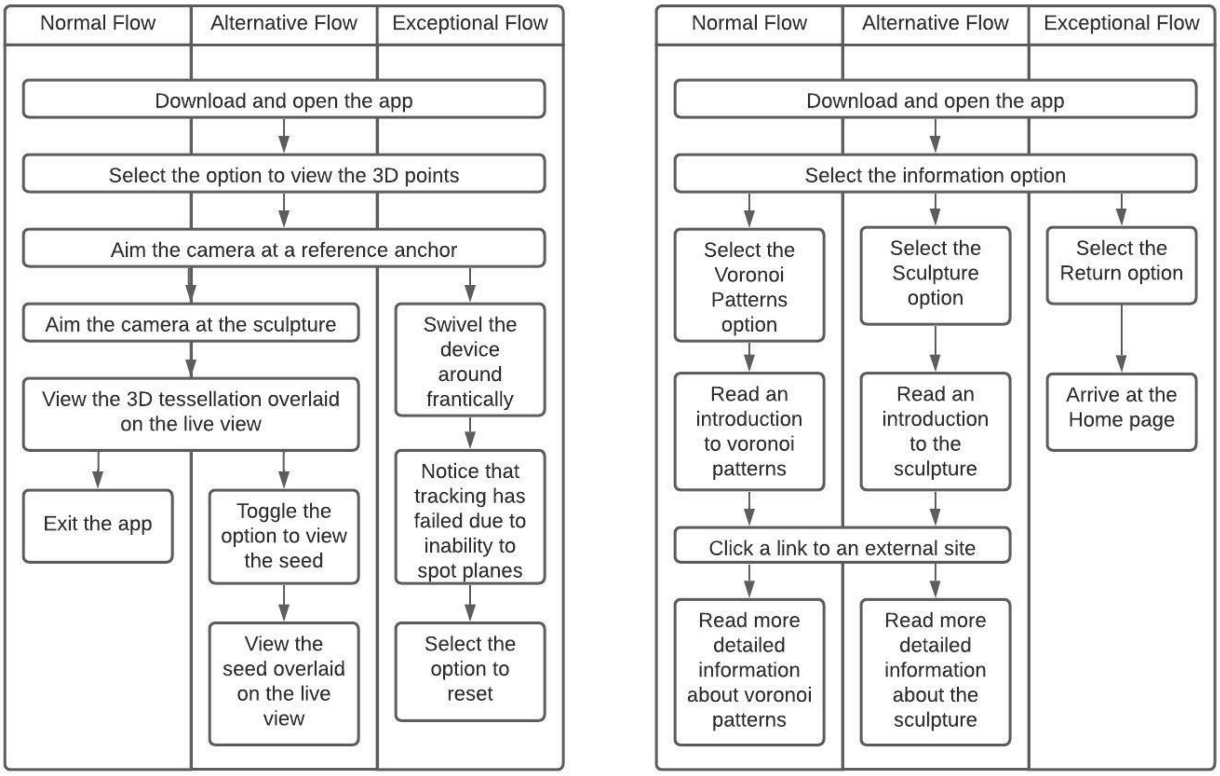

Figure 1 – flow diagrams for user story 1 (left) and 2 (right)

Requirements - User Story 1

This user story reflects the main purpose of the app. The flow steps have been broken down into atomic requirements below. Additional key features have also been included for completeness.

Index	Requirement	Additional Info
1	Accessibility	
1.1	Be publicly available in the Google play app store	App should be searchable in play store and have a listing page with screenshots
1.2	A download QR code should be made	A QR code should be visible near the sculpture as a link to the Google play listing
1.3	Able to run on most/all android phones	Phones will require AR capabilities but app should run on all android phones that fulfil this criteria

2	Functionality	
2.1	Display information about the sculpture	App should contain a page with sculpture images and information
2.2	Educate users on voronoi tessellations	App should contain a page with sculpture images and information
2.3	Display the view through the user's camera	
2.4	Allow user to scan QR code to activate model	QR codes for alignment will be visible near the sculpture
2.5	Visualise the 3D model of the sculpture	Model should be visible as a wireframe
2.6	Align 3D model of sculpture over 2D sculpture	Model points should line up with existing sculpture
2.7	Allow user to realign if model moves too far	Method for changing x y z coordinates/ rotation/ size
2.8	Allow user to toggle seed points in their view	Display the set of generating points that created the sculpture
3	Maintainability	
3.1	All program should be well documented/commented	All source code should be completely explained
3.2	Source files should be provided with specific instructions on how to change assets	There should be a collated list of important file paths e.g. the location of the target anchor image
3.3	Client should be given instructions on deployment to Google play	Client should be aware of the steps taken to update the app listing updates/deprecation
4	App permissions	
4.1	Ask user permission to use phone camera	App should make a popup requesting camera permission before it is used

Table 1 – Requirements

3. Personal Data, Privacy, Security and Ethics Management

Personal Data

The application measures the user's position within the Fry building using the anchor points from the QR codes. The use of the camera may show personal information, including faces and other potentially private images to the app. However none of this data will be stored, the location and camera will only be used for immediate calculations and to display to the device screen and as such will not be recorded for any reason.

Privacy and Security

The app will have access to the user's camera, once permission is given. With applications released for android devices it is impossible to use the camera without the user having given consent. The application includes a statement outlining the fact that any imagery will not be stored or used for any purpose except from the application's intended purposes. The application will only use the device's rear camera and as such is unlikely to capture the primary user's face, and does not include any form of facial recognition anyway.

A password is required to access the developer mode. The developer mode only provides access to additional in-app features concerning manually aligning the model. It does not provide access to personal data outside the app or have any effect on the app in the Play Store. Therefore we decided it was not necessary to include a sophisticated password system.

Ethics

Despite no current laws prohibiting it, we have the permission of the university's School of Mathematics to incorporate Augmented Reality on their private property and on the sculpture itself, to avoid any ethical dilemma preventing the use of AR in the Fry building. The application includes a statement making it clear to the user that the application uses AR technology and so any images shown though appearing through their device's camera may not accurately portray the environment in front of them. This avoids any incidents of user's being misled by the application's display and potentially injuring themselves or others.
This application may also result in some ethical mental problems, for example young children or those suffering from mental health problems may be scared or confused by some features of the application, but this can easily be resolved by implementing an age restriction upon users of the app, and including in the previously mentioned statement a section outlining the fact that the app should only be used by those comfortable with the use of AR above a certain age.

4. Development Testing

Strategy

Throughout the development cycle we have employed two different testing strategies: one for the usual features and functions found in most applications and another for the very specific task our project had to accomplish: presenting in real-time a simulation that matches up with a real-world sculpture. The former strategy consisted of a mix of simple automated testing and some by-hand functionality testing done either on our own phones or digital emulators. The ladder had to be accomplished entirely through by-hand calibration. We did set up a digital replica of the real-life object in order to expand our weekly windows of opportunity by allowing us to test different configurations without visiting the Fry Building.

Example Component: Scan the reference anchor

One such component that had to undergo careful by-hand testing is our anchor detection system. As mentioned in the Requirements section for spatial calibration purposes, our application uses the phone's camera to scan for a certain picture which needs to be displayed in a known position in the vicinity of the Voronoi screen. From there it will be able to calculate the device's position and rotation relative to the sculpture.

This is an essential process as it 'anchors' the virtual representation to the real world object. As such, extensive care and thought have been put into crafting the most precise experience possible. To ensure this, over a large number of weeks we have each individually subjected this system to a multitude of different possible conditions to test the full extent of its predictability. Each time we checked what orientation and distance it calculated for the device and compared it to what we measured in the real world (The system is supposed to only work a fixed distance from the picture).

Test	**Description**	**Expected Result**
Normal vertical position	Simulating the normal situation in which a user attempts to scan the reference: · The device is held vertically with the screen parallel to the reference. · The device is held 1m from the reference (distance measured manually).	Orientation = 0.0 Distance = 1.0 Reference = ref_id
Alternative horizontal position	Simulating an alternative situation in which: · The device is held horizontally with the screen parallel to the reference. · The device is held 1m from the reference.	Orientation = 90.0 Distance = 1.0 Reference = ref_id
Alternative Angled position	Simulating an alternative situation in which: · The device is held at a 45-degree angle to the reference (orientation measured manually). · The device is held 1m from the reference.	Orientation = 45.0 Distance = 1.0 Reference = ref_id
Reference partially obscured	The reference is partially covered up. The device is held in the 'normal vertical' position'.	Error: no reference detected
Reference obscured	The reference is completely covered up. The device is held in the 'normal vertical position'.	Error: no reference detected

Table 2 – Development testing table for scanning the reference anchor

5. Release Testing

Strategy

For the purposes of integration testing each of the different components' tests were run before and after integration to ensure that they all still worked as intended on their own. Tests were then performed on the transition between usage of the different modules to ensure that the system functioned correctly.

Example User Story: User Story 1 (Viewing the 3D Model in the Fry Building)

The chosen example is the process of displaying the points in 3D space. As stated before, these are imaginary pre-calculated points floating inside the Fry Building. They represent the backbone of the whole tessellation the slice represented by the sculpture is based on. For our app to be able to accurately portray the tessellation the points had to be tracked perfectly.

Test	**Description**	**Expected Result**
Normal slow movement	Simulating the normal situation where a user is gently moving around the designated area in order to admire the 3D points	A perfect match to our perception of reality
Faster movement	Simulating a more frantic movement around the designated area. The speed may be increased for testing the limits of our app	A perfect match to our perception of reality
Temporarily obstructed	During some normal slow movement test we will deliberately cover one or more usually visible points for a short duration. The number of obstructed points may be increased later for further limit testing	Perfect tracking, despite interruptions in displaying
Poor lighting conditions	Simulating the rare situation in which a user will attempt using the app in the designated area at night or during poor weather or other natural conditions that lead to poor light levels. Initially the torch will be light on, but upon further testing it will be toggled off	A good match to our perception of reality while the torch is on and a potentially acceptable match otherwise (the expected use does involve turning the torch on)

Table 3 – Release testing table for viewing the model in the Fry Building

6. Architecture

The system of interaction between the user and the app is as follows:
Upon entering the fry building the user will be able to download the app by following the QR code link posted near the sculpture. The user will require an AR enabled android phone in order to use the app's capabilities.

From here the user's interactions with the app are documented in the flowchart below.
The user interacts with the app, moving between different screens and moving the phone around, and the app displays the AR view to the user.
The app will be downloaded on their phone, and will request permissions for the camera access, and also process the camera view passed to it from the phone.
The phone receives information from the camera that is used by the app to process the AR sculpture, and the camera receives information from the user's surroundings as they point the phone at the sculpture and the QR code.
The QR code is posted near the sculpture and will not move with relation to the sculpture meaning the app uses the QR code to place the sculpture in the correct position with respect to the augmented reality space.

The app itself also uses methods from the ARCore and OpenGL libraries, which allow the app to track the environment through the camera and render the 3D model over the camera view, respectively.
The app contains pre-made rendering files of the tessellation; the model is not calculated at runtime in order to make the app more efficient. This means that the OBJ asset files need to be provided along with the app data when the user downloads the app from the Google play store.

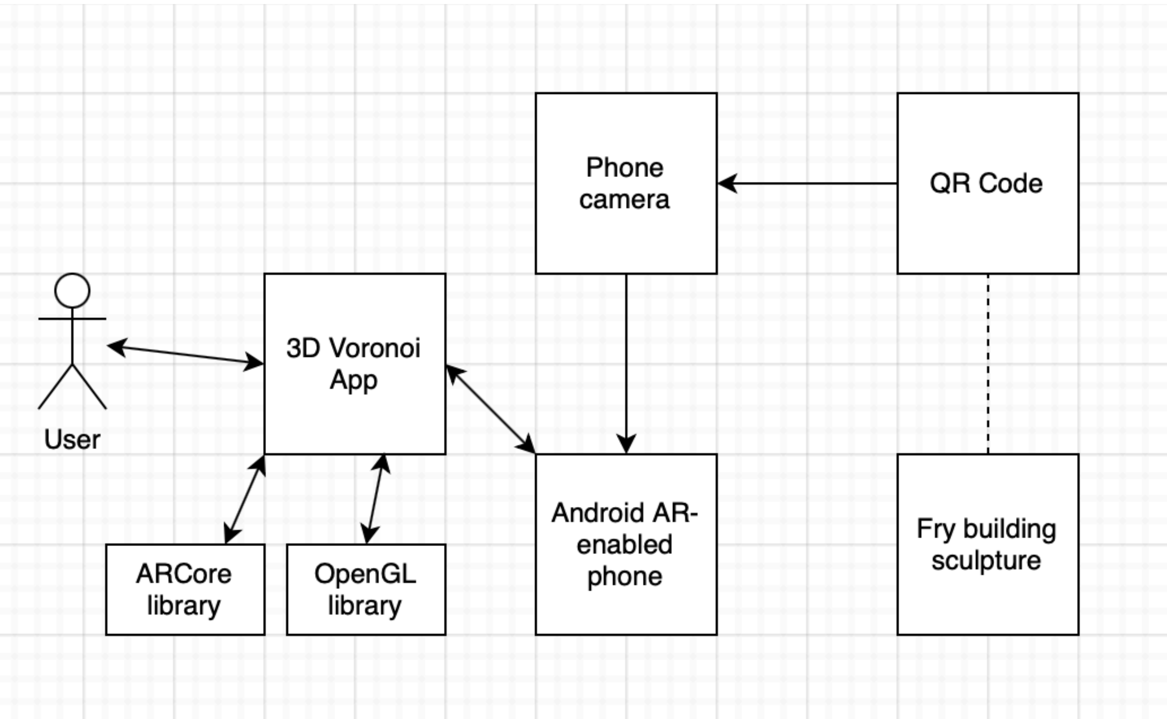

Figure 2 – A flowchart describing the use of the app with the available hardware and environment

7. Object Oriented Design

Static Modelling Aspect

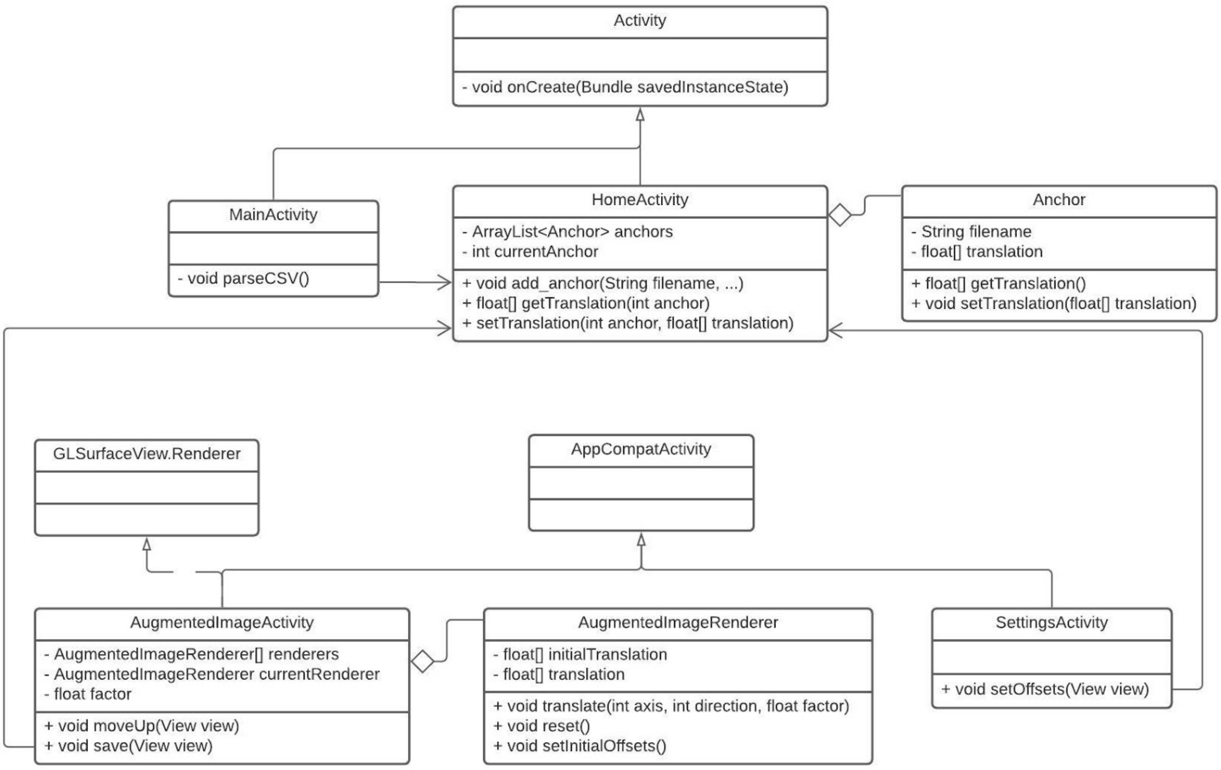

Figure 3 – UML class diagram of the components involved in registering an anchor image.

The Anchor Image Component

The most important feature of the app is displaying the full 3D model of the tessellation aligned to the sculpture in the Fry Building. From early on in the design process we settled on the idea of using a QR code as a detectable image from which to anchor the CG tessellation model. As development progressed, it became apparent that ARCore was ineffective at detecting the position of the anchor image as consistently as desired. The requirements therefore changed to put more emphasis on the alignment tools used to change the offset of the CG model from the anchor image. By the final release, this had become one of the core features of the app. The chosen modelling aspect to discuss is therefore the storage and manipulation of anchor image data as illustrated by the UML diagram above.

Driving Factor: Use Requirements

Firstly, the client needs to be able to add new QR codes to stick around the Fry Building. This prompts the need for a simple method of adding this data to the project to enable the app to adapt to the new QR poster(s). Another requirement is for the client to be able to determine good initial offsets of the CG model from the anchor in the first place. This means that we needed to develop a system for them to be able to manually align the CG model and view the new offsets.

We decided to store the initial anchor data in a CSV file. It includes a list of anchors, each with their image file name and width as well as initial translation, rotation and scale offsets of the CG model from this image. We opted to use a manual entry CSV format (with accompanying documentation for the client) because the data is very simple and specific to this particular case. Alternative options such as using a database or GUI entry system would require additional storage, code and dependencies, resulting in unnecessary complexity.

The MainActivity class is the entry-point to our app and contains the CSV parser. Once the data has been extracted from the CSV file it is stored in static Anchor objects. The code responsible for holding and manipulating these Anchors is entirely static. This is so that these initial values can be accessed at any point following app initialisation, regardless of the current page.

There are two methods of manipulating the offset values of the anchors at runtime: via text entry in the Settings page or using the alignment tool suite in the augmented reality view. This process is described in more detail in the dynamic section below. The SettingsActivity takes values entered by the user and updates the static anchors by making a call to the HomeActivity's setter methods. The use of this settings page with text input enables the client to have a clear view of the offsets as required for modifying the CSV file.

The AugmentedImageRenderer keeps its own copies of the initial anchor offsets. When the user interacts with the alignment tool suite in AugmentedImageActivity, it only modifies these variables. This allows for save and reset functionality by updating and fetching from the static anchor data respectively. This completes the triangle, giving the client fine-tuned control over alignment when in the augmented reality view, but also allowing them to save their values so that they can switch to the Settings page and view them more clearly.

Driving Factor: Android UI and ARCore

Another major factor that influenced the design of the solution was the interfaces that needed implementing in order to make use of the UI elements provided by Android and the graphics module by ARCore. As shown in the diagram above, all app pages required an 'Activity' class. For example, HomeActivity displays the home page with buttons to navigate between the other pages in the app. It therefore extends Android's Activity class, giving it access to the onCreate method that sets up a page in an Android app.

Similarly, other classes such as SettingsActivity extend AppCompatActivity. This class allows older versions of Android to make use of newer features, extending the range of devices that are able to make use of our app. This relates to a key requirement set out by the client in making the app accessible to a wide range of users.

The AugmentedImageActivity class implements GLSurfaceView.Renderer in order to access methods that interact with the OpenGL rendering tools. This is essential in order to render the CG model of the sculpture/tessellation overlaid on the camera view of the real world. The client's fundamental requirement of the augmented reality feature made this implementation a necessity.

Dynamic Modelling Aspect

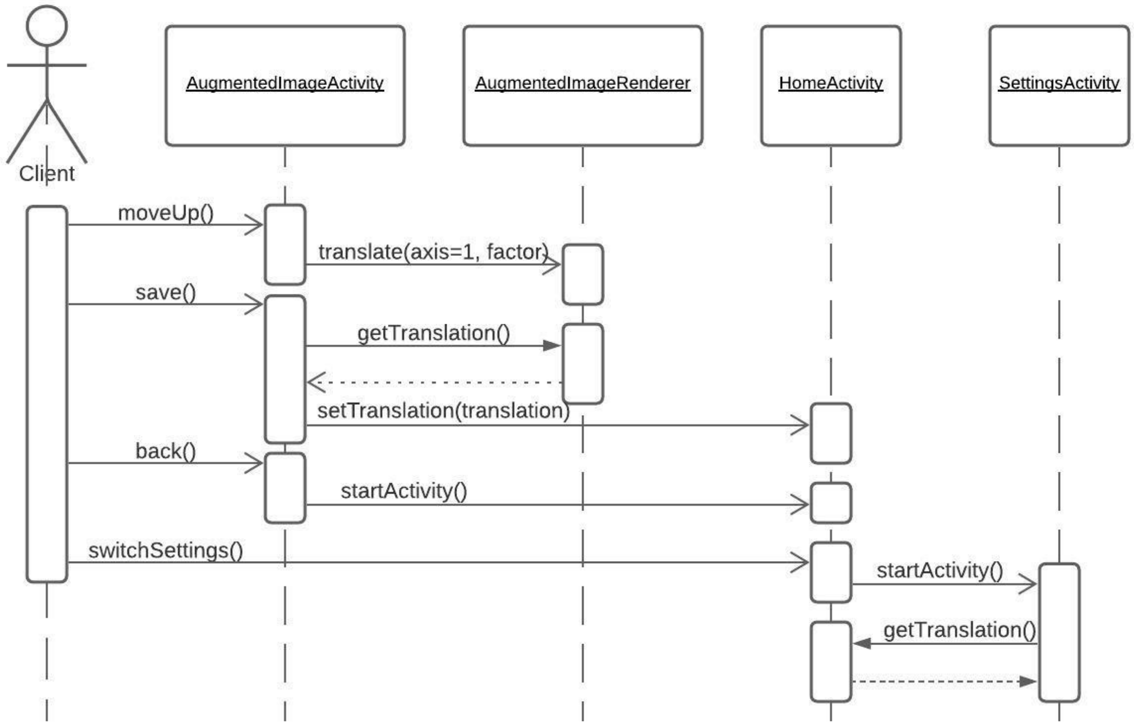

Figure 4 – UML sequence diagram of the process involved in modifying offset values

The sequence diagram above illustrates an ordinary use case where the client is using the app to align the model as best they can to the sculpture. They would then record the offset values for input to the CSV file when registering a new QR code in the Fry Building. For the sake of simplicity and clarity in the diagram, only the manipulation of translation has been shown. Rotation and scale are also updated in an identical manner.

The user interacts with alignment buttons (such as 'move up') in AugmentedImageActivity to fine-tune offset values. These are updated in AugmentedImageRenderer (by translating along the y=1 axis in figure 4). When the user is ready to save changes, the global initial state is updated in HomeActivity by fetching the current state of offsets from AugmentedImageRenderer and calling HomeActivity's setter methods. The user can then navigate to the settings page, which on start-up accesses and displays the newly updated initial offsets in their entirety.

Splitting functionality and display in this way allows each component to specialise in a particular function and improves user experience. AugmentedImageActivity provides the user with a suite of buttons to move the CG model around the screen. In order to de-clutter this screen and make the experience more user friendly, a separate display of all offsets with a text input system is provided in Settings. The flow of data as demonstrated by the example above is therefore the minimum necessary for saving current and initial offset values.

8. Acceptance Testing

During the development process a total of one user testing sessions were conducted.

The Approach To Validation

In order to obtain data for improving the application, a user testing strategy was designed that outlined the process involved with collecting the data from users. The motive behind user testing is that the subsequent iteration of development would then be informed by the results of the test session. For the single session conducted during the project, a questionnaire was designed.

An interview-styled procedure guided by the pre-made questionnaire was the original choice of method for obtaining user data. A demonstration of the current version of the application at the test site would be performed for participants, and they would be walked through individually by the interviewee through the set of questions in the questionnaire, which addressed different aspects of the application. The use of a questionnaire meant that a focused set of points relating to individual aspects of the system could be addressed separately. The interview approach to conducting the session would allow a less restricted form of obtaining data if needed by deviating from the planned structure and introducing new questions if the interviewer so wished or found an opportunity to do so.

Context And Environment

At the time, due to the nature of the worldwide pandemic situation, in-person interviews could not actually be conducted at the test location. Instead the questionnaire was converted into a *Google Forms* document which could then be shared to prospective participants using an online link. Using *Google Forms* meant that all participants' results could easily be collated for analysis. The interview aspect of the testing session could not be easily replicated; an alternative format similar to an in-person interview was done instead; a video demonstration of the application being used was shown prior to conducting the questionnaire, and the questionnaire was extended to collect further comments about the application. Hence ultimately an alternative form of acquiring data was performed; participants involved could partake wherever they were at the time due to the online nature of the study.

The Users

The users who volunteered to participate in the study were mainly first hand contacts of the interviewee, all of whom were students of roughly the same age. This sample was not representative of the overall target group for the final product.

Insights And Feedback

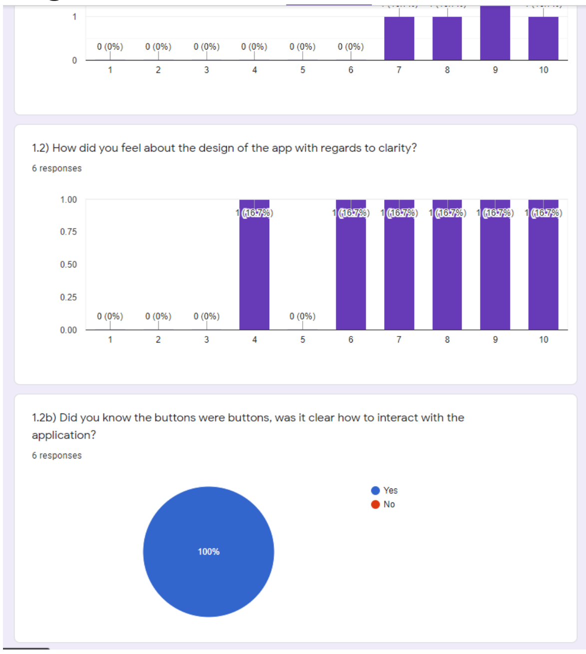

Figure 5 – Summary of user feedback

The responses from the *Google Forms* version of the questionnaire were automatically collated into a summary page by the *Google Forms* interface (*see Figure 5*). From the summary, statistics for questions eliciting numerical ratings from participants had to be manually calculated, and general opinions and feedback were condensed into smaller text summaries. These were copied into a document report that mirrored the structure of the questionnaire. The report itself highlighted key findings, particularly those that suggested areas of improvement for future versions of the application.

Using Evaluation Findings For Refining The Final Iterations Of The System

The following features were introduced/improved as a result of the user testing session (*taken from https://github.com/spe-uob/3DVoronoiRepo/issues/61*):

1. Increase font size on information page
2. Add a help popup window to explain how to use the AR feature
3. Add a warning popup before using the app/AR feature saying that it uses AR and may not be suitable to kids, etc.
4. Add links in the information page to extra info regarding voronoi tessellations, the Fry Building, interactive Voronoi apps, etc.
5. Replace the images on the information page with our own to avoid copyright infringement

The issues raised were consequently resolved - thus if the summarised participant responses can be considered as criteria for achievement, then the development iteration following the test session turned the application into something more valid for prospective users - so the implementation of the testing strategy succeeded in informing the development of the application.

9. Reflection

Success

The customer expressed their high level of satisfaction with the final version of our application. We created a simple intuitive design that helps illustrate in an immersive manner the 3 dimensional voronoi tessellation that inspired Fry building's shading screen. During our collaboration we have also put in place an easy to use system for our client to be able to adjust the overall experience of the app after release. This has been accompanied by a comprehensive written list of instructions and a few in person demonstrations. All this factors combine to make a system successful in its original scope and future-proofing

Effects of our application

Now any visitor of the University of Bristol can try, upon entering the Fry Building, a short educational demonstration of not only the fascinating intricate world of mathematics that is studied and understood here but also the scale of opportunity offered by our institution.

From now on any mathematics student struggling to picture how a 3 dimensional voronoi pattern may take shape can easily visualise it after just a few taps right in their school's lobby. Any curious mind that observes the interesting shape keeping the foyer cool has access to a quick summary of its origin alongside an interactive representation. And any possible future computer science students interested in the kind of project they might participate in at the University of Bristol will be able to just take a look at our application and its results.