Mobile Development CA2

Sean Stephens, Conor McGuire, Sean McAvoy, Robert McAteer

For CA2 we decided to build upon the 2D Animation Portfolio Application we built for CA1. We felt that this application was still appropriate as we could incorporate a variety of hardware features into the existing application.

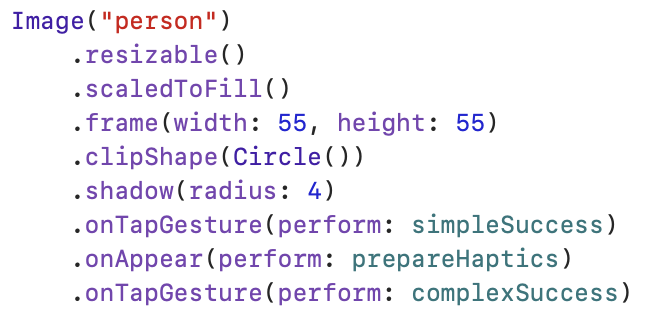
**Haptic Feedback**

Sean Stephens

In order to further incorporate the hardware of the device, I looked at adding haptic feedback to the user’s experience. Haptic feedback involves the device vibrating in the user’s hand. The code used to create the haptic feedback functions can be viewed below.



For the purpose of this CA, I decided to apply the Haptic functions to both the Profile Image and the Character Design heading of the Character Design page. The code used to assign the vibrate functions can be seen below.



When the user taps on the relevant areas the device will vibrate, below are shown the areas where the user can tap, which will cause the device to vibrate.

A close up of a logo

Description automatically generated with low confidence

**Push Notifications**

Sean Stephens

I then decided to incorporate push notifications for our application, the intention of these notifications is to make users aware of when an individual updates their portfolio. This will allow users to keep up with their favourite creators’ work. Below is the code shown to enable notifications.



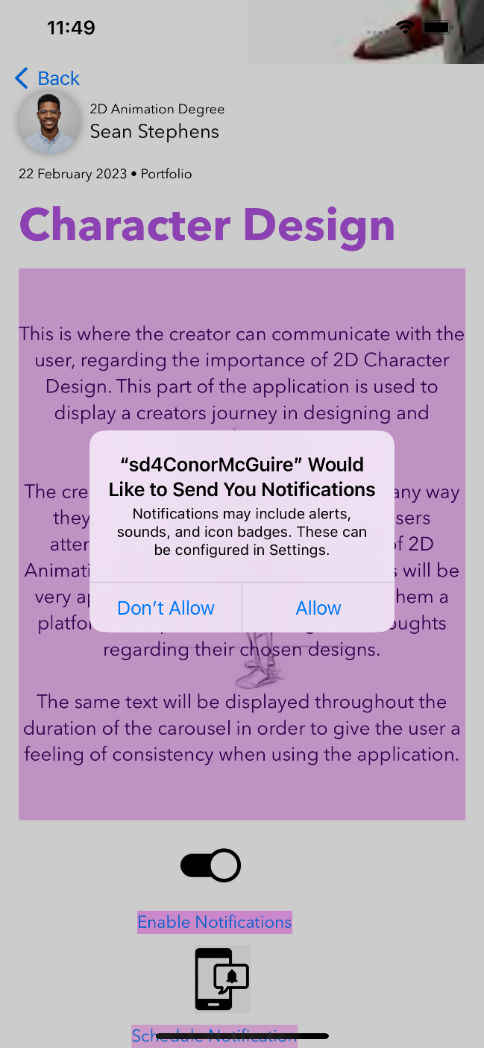
Once the User has enabled notifications, the application is then able to send notifications to the user’s device. The code to do this is shown below.



When the user presses the enable notification button, they will be prompted to enable the device notifications, the button and prompt are shown below.

A picture containing text, design

Description automatically generated



Once enable, the application can then send notifications, with them appearing on the Users lock screen etc. The button and notification are shown below.



A picture containing text, screenshot, graphic design, graphics

Description automatically generated

**Registration & Login / Gyro Navigation**

Conor McGuire

For my section of our app, I decided to create a **registration and login** page to demonstrate MVC functionality and Gyro **Navigation** to demonstrate a hardware feature. There is also a **push notification** sent to users on a successful login. MVC stands for Model, View, Controller which is the architecture preferred by Apple.

In my code, I use each of these parts:

* The GyroManager acts as the **model**. It handles the business logic related to the gyroscope.
* The SwiftUI View components act as the **views**. They display information to the user and handle user input.
* The LoginViewModel could be seen as a **controller**, as it manages the logic for user authentication and storage.

As we are not connecting these apps to a database, I decided to use local storage for the login and registration feature. This means that once the user registers their username and password, these are stored locally using **‘User Defaults’** and checked against the information they enter in the login screen.

A screen shot of a computer

Description automatically generated with medium confidence

For the gyro page, I made some simple code that checks the speed of the phone’s rotation on the y-axis to decide whether the user wants to navigate back or forward by one page. This is not a perfect way to write this as it does not register direction. A gesture recognition algorithm would be a solution to this but is not in scope for this project.

A picture containing text, menu, handwriting, whiteboard

Description automatically generated

**Background Idea – Voice Recording Feature**

Sean McAvoy

For CA2 I decided to create a voice recording featuring where the user could record an idea they get for a background design and later play the recording back to themselves so they don’t forget what they were thinking of doing, how they were planning it and the colours schemes decided.

Below is the code needed for this feature the UI design of this Page to include the Record and stop recording button. The play audio back button and to see the Audio that is

recorded. The screenshot on the right shows what this page looks like.

A screen shot of a computer code

Description automatically generated with low confidence A screenshot of a phone

Description automatically generated with medium confidence

Next in this feature I had to create an Audio Recorder: Observable Object. This allowed me create the Audio Record object which allows the users to start recording, stop recording, Play recording. It also has methods inside to allow the phone to save the recording, load the recording.

Below is the start of the Audio Recorder method. This is Setting up an object for the speaker and also creating an array to save the recording.

A picture containing text, font, screenshot, line

Description automatically generated

The next method is the start recording method.

A screenshot of a computer program

Description automatically generated with medium confidence

The next 4 methods are for Stopping recoding, Play recoding, Save recoding and Load recording.

A screenshot of a computer code

Description automatically generated with low confidenceA screenshot of a computer program

Description automatically generated with low confidence

I feel this feature will be very useful for the user with idea generation and coming up with his final designs in background design of 2D Animation.

Final product of Background Design Section of the App:

A screenshot of a cell phone

Description automatically generated with medium confidenceA screenshot of a phone

Description automatically generated with medium confidenceA screenshot of a phone

Description automatically generated with low confidence

A screenshot of a video game

Description automatically generated with medium confidenceA screenshot of a phone

Description automatically generated with medium confidence

**Storyboarding Voice Over Feature**

Robert McAteer

For CA2 I have utilised the device speaker hardware to improve the usability of the app. The new voice over feature is enabled using a button titled “Play Voice Over”. I wanted to introduce this feature to ensure that the app we have developed is as user friendly as possible. Once pressed, the button will activate a voice over feature that will read out the text on the screen. I took inspiration from art gallery tour guides I have been on recently, as I thought I this would add an extra element to the experience of the app.

A screenshot of a computer

Description automatically generated with medium confidence

The snippet below is the code I implemented into the app. This code shows me initialising a button element with the text “Play Voice Over”, this will be displayed to the user to interact with. I have then initialised a AV speech utterance with the string I want to read out to the user. I set the language to English and the read back rate to 0.5, to ensure the user can clear understand the text as it is translated to them. I then set the synthesizer to read the string/text once the button is clicked.

A picture containing text, font, screenshot

Description automatically generated

I have then styled the button to have a prominent border and the control size to be small, as I feel it incorporates well with the rest of the apps design.