2. METHODOLOGY

This section contains the choices we have made as a group to make this project to the best of our abilities. It will clearly articulate the reasons why we chose the particular procedures or techniques below.

2.1 APPROCH TO DEVELPOMENT (AGILE)

How did we choose this approach? How did we envision this agile approach?

*“Agile development methodology provides opportunities to assess the direction of a project throughout the development lifecycle. This is achieved through regular cadences of work, known as sprints or iterations, at the end of which members of team must present a potentially shippable product increment. By focusing on the repetition of abbreviated work cycles as well as the functional product they yield, agile methodology is described as ‘iterative’ and ‘incremental’. “*

Since this was an ongoing project from last year Alina had to sit down with the other team members John and Diarmuid and explain the previous approach taken to create the application. After this meeting we had come to the conclusion of dividing the project equally giving each team member a section of the project to work on. Here is where the agile approach kicked in. Each week we would go and work on the section we were responsible for and at the end of the week we would have meetings to show the progress we have done. By having these meetings as often as we could, we came up with solutions for problems arising.

Meetings with the supervisor and the client were done when we had something working to show. Meeting with our supervisor put things into perspective with us. The supervisor guided us to an idea we didn’t think about at the start of the project. Creating a database and an uploading software that the client can use to upload more data to the application. This idea was brought up in a later stage of the project therefore we had to meet again with the team and divide ourselves to create this backend section of the project.

With this methodology, the “Product Owner”, client worked closely with the team to identify and prioritize system functionality. The team met with the client whenever was needed to be done in order to successfully deliver a working software system. Agile emphasizes on the speed and efficiency of development workflow, and relies on rapid and reliable feedback between programmers (us) and customers (our client).

2.2 TESTING AND VALIDATION

For testing and validating our code we used two major ways.

1. **Using the Unity Editor**:Most of the code testing was done using Unity’s internal Editor. The editor automatically compiles all scripts attached to the active scene, and can be ran using the “Play” button. This allows simulation of most of the engine’s functionality, and means everything from simple logical and GUI scripts to database calls can be tested using it. However, the editor is not without limitations. While it allows for basic touch and drag gestures, multi-touch doesn’t work without building the project as an executable, (or APK file for Android). Another problem that arose was when we were building multiple iterations of the program to test it on Android devices, Unity handles iterative builds by using dll files to keep track of compiled scripts. However, this can cause certain scripts to not recompile properly. This lead rise to game-breaking bugs in the builds that didn’t occur when the program was run in the Unity Editor. This was rectified in the end by updating Unity to the newest version.
2. **Downloaded the android SDK**: The Android SDK is composed of modular packages that you can download separately using the Android SDK Manager. For example, when the SDK Tools are updated or a new version of the Android platform is released, you can use the SDK Manager to quickly download them to your environment. Simply follow the procedures described in Adding Platforms and Packages. Android studio provides the fastest tools for building apps on every type of Android device. The Android SDK provides all the necessary developer tools to build, test, and debug apps for Android in Windows, Mac or Linux.

We have tested our frontend and backend code on this. We downloaded the Android SKD and built the project onto it. This allowed us to see the application working on a phone, how it behaved and how it looked. Because the database was connected to one of the sections of the application at that stage we could also see the database in action. Few images were pulled from the database and were running as expected.

1. **Testing the database:** Connected the database to the application allowing sample images to show up in the section where this was connected. In order to ensure the images uploaded to the database will be added to their respective categories (or themes), the sample images in the database included themes both existing and new. When the images were downloaded, they were sorted into their respective themes. A new theme is made if it doesn’t currently exist.

****

This screenshot shows the application returned images from Amazon S3. The returned image (in this case a coconut tree) was saved under the theme “DLC”, which was not originally in the game. When downloaded, the application creates a new theme and sets the applicable image as the theme icon. The “DLC” theme shown here will not display if the app is not connected to the internet, or there is a problem connecting to the database. This way, the backend functionality is as unobtrusive as possible and users can play the game normally without needing to connect to the server. This means that the administrator can create new themes whenever they want and users can download them to their platform. An example would be a Halloween or other festive theme could be added to allow more replay-ability.

2.3 SELECTION OF PLATFORM, LANGUAGE, TECHNOLOGIES

Due to the clients’ requirements we had to create a cross platform application, so there was only a few software options to choose from.

*The process in choosing the right software:*

***IONIC:***

**Advantages:**

* Cross platform
* Main development in HTML, CSS and JavaScript.
* And being able to use great frameworks like angular, which is embedded in ionic by default.

**Disadvantages:**

* No experience with this piece of software
* It will take a bit of time to get used to using it
* It’s been described as being slow, tricky

***UNITY:***

**Advantages:**

* Easy to use
* We have all used it before
* Cross platform
* Has a free version
* Allows for rapid prototyping
* Asset store
* Collision detection without mathematics

**Disadvantages:**

* Using the engine requires you to agree with their policies 
* Expensive if you need all features.

In the end considering all of our options and all of our skill sets we decided to use Unity. Most recommended software for creating quality games. Also allows team work. Due to the software we decided to use we were limited in our language choice. C# is the language Unity works best with. This is also a language we have used over the years in this Course therefore it was a win for us.

2.4 USING GIT

In our journey to get the project finished we have adopted a smart way of keeping our code safe, while also allowing us to contribute to the project together. We have chosen the cloud software called GitHub.

GitHub can keep Open Source projects open, so anyone can view and contribute, while private projects get the same features but are only visible when you're signed in. Our team could use GitHub to manage the project, pull in customized versions of open source code, and share our own open source libraries and tools with the world all in one place.

In GitHub there is an option on GitHub called fork. A fork is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project. In our case each team member forked the repository that was created for the project. This repository in the beginning held the skeleton for our project. Each of us then worked individually and committed changes. All these changes didn’t affect the skeleton. In the end to pull the project together we merged the forked project and created a final complete project.