Captain Cillian

Alina Danci

Diarmuid Byrne

John Lavin

B.Sc.(Hons) in Software Development

April 26, 2016

Final Year Project

Advised by: Dr Damien Costello

Department of Computer Science and Applied Physics
Galway-Mayo Institute of Technology (GMIT)

Contents

About this project

Abstract "Captain Cillian" is a standalone mobile application based on a series of children's books of the same name. The application is comprised of interactive, educational pages for children to enjoy. The software is designed to be used with both touch devices and keyboard/mouse to maximize user base and ease-of-use for its audience. The application carried on from last year to this year because it was unable to reach the clients expectations at that stage. The project was completely remade since the software used last year wasn't cross platform. This new application will be free to download from the corresponding Application Store. It will be cross platform. It will have a number of engaging games for children to play but also learn at the same time. The application will advertise the books which the client is trying to sell. To achieve this application Unity software was used. Unity is an amazing piece of software that allows any game or application to achieve it highest potential. Due to the use of correct software, great team work and sample code from the Unity asset store we have achieved to complete most of our tasks and objectives. An important factor that we have added to our application is the connection of a database. This will allow the application to be updatable in the future. We persisted to get the best results even though we had time limitations, software limitations and problems along the way.

Authors Alina Danci, Diarmuid Byrne, John Lavin.

Chapter 1

Introduction

As it was declared in the statement above, "Captain Cillian" is a standalone mobile application based on a series of children's books of the same name. The application is comprised of interactive, educational pages for children to enjoy. The software is designed to be used with both touch devices and keyboard/mouse to maximize user base and ease-of-use for its audience. Captain Cillian is an "Edutainment" application therefore it will have some learning features as well as fun features. If you are not familiar with the term "Edutainment" it will explained in a few words. The American Heritage Dictionary defines edutainment as "

the act of learning through a medium that both educates and entertains." If we break up this definition we will find that the most important words would be education and entertainment. These two words combined create a perfect learning environment for children in our century. This was the challenge we took on for our final project this year. Creating this application to be fun, educational as well as cross platform, high quality and at the same time advertise our clients books. This wasn't the first attempt at creating this application. This application is ongoing from last year. The team last year have not met all the clients requirements that's why it was continued this year. The team last year was made up of Alina Danci and Dennis O'Neill. Due to Dennis switching courses in the final year the project was continued by Alina and two new team members. Their names are John Lavin and Diarmuid Byrne. Because the new team members didn't know anything about this application and its history there was a meeting at the start of the year with the client. Since the client was busy at the time with advertising the series of books she left Alina in charge of explaining the concept to the new team members John and Diarmuid. This process took a few weeks. Then the application from last year had to be assessed to see

what we could keep and what there was to throw out. Unfortunately the application was previously done in Visual Studio. Visual studio did allow the application to be cross platform and it would have taken too long to keep the code convert it to each platform and fix the problems it had. After a few more weeks of brainstorming the team decided that it was better to star the application again and use a special cross platform software. Taking into consideration advantages and disadvantages the team choose Unity to remake the Captain Cillian application. The next few weeks the team began on actually creating and coding the application. The application was divided into sections and all the team members got to choose a section to work on. The following information will explain all these processes in detail giving a complete view of how the product was created and what it was like working on this project. It will consist of objectives, problems faced, the outcomes and an explanation on the experience of working in a team to create this application.

Chapter 2

Context

2.1 Objectives

- The project was put back into production due to some requests the client had for the application and were not met:
- The previous application was created on a Windows phone platform. Therefore not reaching the clients needs.
- The client wants an application that can be cross platform that would be able to reach more audience.
- The client also wants more quality and less quantity.
- The client changed her mind about displaying content of the books in the app and now wants the image of the books displayed with a link to her website where they can be purchased.
- The client changed the main headings of the application. These have to be updated from Create to Create and Play, from Fun to Ocean Facts, from Learn to Learn Irish and Explore remains at the same heading.

2.2 Sections of the project

2.2.1 Explore

The explore section will draw heavily from content within the books, as selecting explore from the main menu will provide another menu displaying the different book covers. Clicking each cover will bring you to the website where you can buy the books online.

2.2.2 Learn Irish

This section will focus on familiarizing the users with common Irish words and phrases. It will consist of a "match-up" game where multiple Irish words or short phrases are given on a grid layout, and must be matched to a corresponding pictures/icons. The grids are 4x4, or 4x5 but are shown below as 3x3 for easier illustrative purposes.

2.2.3 Create and Play

Write a story allows the user to create their own storyboards, by supplying them with a background scene and sprites on a navigation bar at the side of the screen. You can drag and drop icons, zoom them in or out as you please, you can also type in a little story about what you have created.

2.2.4 Ocean Facts

The Ocean Facts section draws from the theme of sailing with Captain Cillian. The scenes contains a puzzle game Pick and place the puzzle piece. On the side of the puzzle there will be a fact displayed about puzzle you just created. This is a new way of learning facts while also having fun.

2.3 URL- Link to GitHub

1 https://github.com/AliLigi/ProjectDocumentation - This link will bring you to the place where our documentation is stored. We decided to create two different repositories because the amount of information was too large to fit into one github repository. This repository holds the Specification that we did at the start of the year for the client, it also holds the previous documentation from 3rd year and other planning documents and information that was important to add to the project.

2 https://github.com/DiarmuidByrne/Captain-Cillian - This repository stores the code that we put together, this will display the final code. Each team member forked this project and updated on the section they were working on.

Chapter 3

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.

3.1 Approach (AGILE)

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

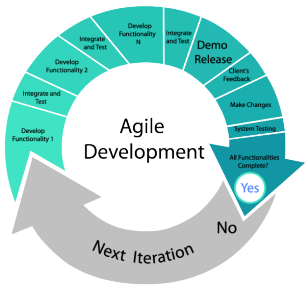

Figure 3.1: Agile process diagram.

"Agile development methodology provides opportunities to assess the direction of a project throughout the development life-cycle. 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 back-end 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 work-flow, and relies on rapid and reliable feedback between programmers (us) and customers (our client).

3.2 Testing and validating

For testing and validating our code we used two major ways. First placing the application on the android phone, then testing the database and the uploading software.

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.
3. 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.

Figure 3.2: Create

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.

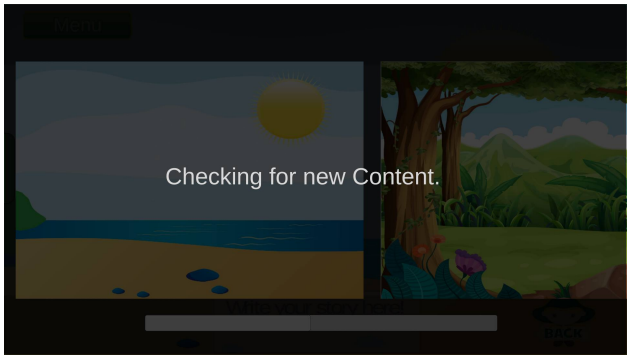

Figure 3.3: This is the application running on an android phone. Showing how its searching the database.

3.2.1 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 applications on every type of Android device. The Android SDK provides all the necessary developer tools to build, test, and debug applications for Android in Windows, Mac or Linux. We have tested our front-end and back-end 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.

3.2.2 Testing the database

Connected the database to the application allowing sample images to show up in the section where this was connected. TO BE CONTINUED. ADD IMAGES OF IT WORKING

3.3 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. We came up with advantages and disadvantages to each of the software so it will make the process of choosing much easier. The process in choosing the right software:

3.3.1 Ionic

Advantages:

- Cross platform
- Main development in html, css and js
- 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

3.3.2 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 sharp 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.

3.4 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.

Chapter 4

Technology Review

4.1 Technologies used

- Unity - is a cross-platform game engine developed by Unity Technologies and used to develop video games for PC, consoles, mobile devices and websites. First announced only for OS X, at Apple's Worldwide Developers Conference in 2005, it has since been extended to target more than fifteen platforms. It is the default software development kit (SDK) for the Wii U. Unity's graphics engine's platform diversity can provide a shader with multiple variants and a declarative fall-back specification, allowing Unity to detect the best variant for the current video hardware; and if none are compatible, fall back to an alternative shader that may sacrifice features for performance. Unity is notable for its ability to target games to multiple platforms. Within a project, developers have control over delivery to mobile devices, web browsers, desktops, and consoles [9].

  Unity also has an Asset store.The Unity Asset Store is home to a growing library of free and commercial assets created both by Unity Technologies and also members of the community. A wide variety of assets is available, covering everything from textures, models and animations to whole project examples, tutorials and Editor extensions. The assets are accessed from a simple interface built into the Unity Editor and are downloaded and imported directly into your project [10].

  Why choose Unity? Besides all these amazing features that are free to use including the asset store. Unity is cross platform and specially created for making games. The quality of the games is at its highest

level. We were asked to create less games and more quality. With this thought in mind Unity seemed the best choice. We researched other software but by comparing the advantages and disadvantages of each resulted in us using Unity. Unity's games are developed mainly in C-sharp therefore perfect for us since we have a great knowledge of this language. The asset store was a very attractive feature for us. Even though we had the graphics made for us by the clients husband there was code that we found useful on the asset store. Choosing this particular software allowed us to meet most of the clients objectives.

- Amazon DynamoDB - is a fully managed NoSQL database service that provides fast and predictable performance with seamless scalability. DynamoDB lets you offload the administrative burdens of operating and scaling a distributed database, so that you don't have to worry about hardware provisioning, setup and configuration, replication, software patching, or cluster scaling. With DynamoDB, you can create database tables that can store and retrieve any amount of data, and serve any level of request traffic. You can scale up or scale down your tables' throughput capacity without downtime or performance degradation, and use the AWS Management Console to monitor resource utilization and performance metrics. DynamoDB automatically spreads the data and traffic for your tables over a sufficient number of servers to handle your throughput and storage requirements, while maintaining consistent and fast performance. All of your data is stored on solid state disks (SSDs) and automatically replicated across multiple Availability Zones in an AWS region, providing built-in high availability and data durability [11].

  We chose this particular Database called DynamoDB because it is a NoSQL database. It is free to use for a year which allows our client to see if she wants to update her application or if she just leave it as is.

  It is always good to give clients the option to choose . Our supervisor Damien advised us to give the client choices, to make the client think at the future of the application. Also creating tables and adding to them is not very complicated which made the job of creating it very smooth. We were worried about creating the back-end of the project since we didn't have much experience but the DynamoDB made this a very present experience.

- Node JS - In software development, Node.js is an open-source, cross-platform runtime environment for developing server-side Web applications. Although Node.js is not a JavaScript framework, many of its basic modules are written in JavaScript, and developers can write new modules in JavaScript. The runtime environment interprets JavaScript using Google's V8 JavaScript engine. Node.js has an event-driven architecture capable of asynchronous I/O. These design choices aim to optimize throughput and scalability in Web applications with many input/output operations, as well as for real-time Web applications (e.g., real-time communication programs and browser games)[12].

  The Node.js in our case was chosen to create an uploading software for the Client. This was the easiest way since the language used is JavaScript. We have learned a little JavaScript last semester in one of our modules. Node.js was chosen over other options because we have used it last year and due to time limitations it was the best approach.

- Languages used - C-sharp is an elegant and type-safe object-oriented language that enables developers to build a variety of secure and robust applications. It is also a modern and innovative programming language that carefully incorporates features found in the most common industry and research languages. In keeping with the design philosophy of C-sharp, Microsoft has introduced several potential new features to the C-sharp language that increase developer productivity with language constructs. Most of our application was done using c-sharp, this is because Unity works best with c-sharp. It saved a lot of time using this language since all of us knew how to use it [13].

  JavaScript- is a cross-platform, object-oriented scripting language. It is a small and lightweight language. Inside a host environment (for example, a web browser), JavaScript can be connected to the objects of its environment to provide programmatic control over them. JavaScript contains a standard library of objects, such as Array, Date, and Math, and a core set of language elements such as operators, control structures, and statements. Core JavaScript can be extended for a variety of purposes by supplementing it with additional objects. JavaScript isn't a language we have practiced a lot. We have used this language only in one module last semester but since we only used it to make the uploading software there wasn't a lot we had to learn [14].

4.2 Research for project: Children and Education Based Games

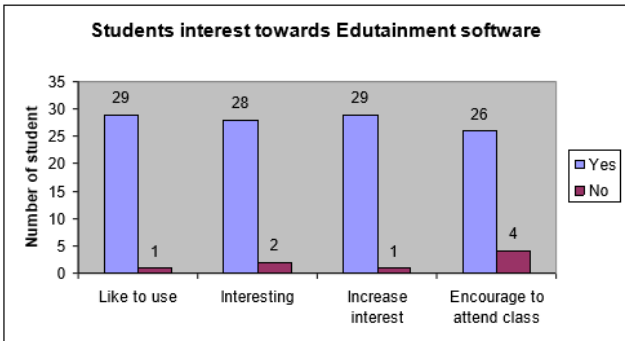

Figure 4.1: Is edutainment a Yes or a No for children..

We live in a very visual world, where imagery is extremely powerful. The old saying a picture speaks a thousand words comes to mind. Imagery can be an extremely effective educational resource. Certain forms of technology can be utilized to portray imagery in an educational manner, computers in particular. Computers are capable of taking this idea a step further. The sounds and graphics gain children's attention faster than any text book could. For this reason young children often find computers very interesting and exciting.

No matter how interesting the features are, if they are not well presented, they will seem less attractive and will not hold the child's interest (Ibrahim, 2004) [3] . Using this research to create our product led us to having a visually attractive application. The use of vibrant colors will definitely attract the children we are targeting. The icons and images were provided by our client who created these graphics specially for this application and her series of books. Normally, the quality of the software can be surmised by allowing the target audience to test it out during development. But since we cannot test on a target audience it was more difficult to achieve this application. Our testing was actually going to the client allowing her to take a look test and tell us what she would like to see different.This is not the ideal solution since Adults and children have a totally different perspective on design and level of difficulty. Therefore another obstacle in the design of edutainment software is the difference in the understanding of heuristics between adults and children.

Heuristics are essentially generic rules or design guidelines or principles for good interaction design (MacFarlane; Pasiali, 2005)[6]. Heuristics in software design have been around for a long time, and for the most part, what works and doesn't work when considering adult-computer interaction has already been established. The problem becomes apparent when considering the design of a piece of software designed to be used primarily by children.

This is backed up by Alsumait, who poses the need to create specific heuristics based on needs of children as well as the requirements of e-learning applications (Alsumait, 2010) [5]. This application is designed to reach to children that need a push in the right direction. For children that are not reached by the present educational materials.

As stated by Walldén (2004), edutainment material is easier to approach than regular educational material. Walldén surmises that extra emphasis should be put on edutainment applications for groups that are not reached by the present educational material [4]. The general consensus is that edutainment, or more specifically e-learning, when done right proves beneficial for a wide range of users. These users range from children with learning difficulties, adults that missed out on an education, or want to learn a new language or skill, or immigrants who need help in getting up to speed in language skills or schoolwork (Walldén, 2004). We believe that our application will motivate children that are not motivated anymore by just reading books and writing stories on paper. Virvou designed a case study based on a large-scale virtual reality educational game, using a significant number of students to partake in the study. Virvou concluded that the "good students" will do well in their studies regardless, whereas the rest of the students would perform poorly in subjects of little interest to them.

"Thus students of little interest in their courses may benefit from extra motivating environments such as those of VR-educational games [1]." Additionally, students prone to cheating or losing interest found themselves absorbed in the game, due to its compelling nature (Virvou, 2005). Another interesting feature that our application holds is in the Create and play section. This section encourages children to create their own little story, to use there creativity and apply it to the game. This great idea is also backed by Robertson. Robertson reinforces this ideal, by describing a workshop in which kids can create their own games instead of

just playing them. This idea brought to fruition when considering students with difficulties expressing themselves in narrative form due to undeveloped or sub-par written skills. Enabling children to express their ideas in a non-textual medium would give them the opportunity to exercise their imaginations and produce an artefact which can be enjoyed by an audience (Robertson, 2005). At the same time it gives them the opportunity to interact and co-operate to overcome problems like solving these educational games. It promotes positive social behavior in the form of good communication skills and team work. It all depends on how the person designing the application. They have the power to use there knowledge in creating technologies and applications that bring people together or alienate them.

"Technology can either alienate us or it can connect us and that is a design decision." JC Hertz (Author, Joystick Nation). In conclusion in order to progress and utilize computers to their full potential there must be some restructuring of the curriculum. The way in which educators view technology as a learning resource may have to be examined. Clough suggests:

"Meaningful technology education is far more than learning how to use technology. It includes an understanding of what technology is, how and why technology is developed, how individuals and society direct, react to, and are sometimes unwittingly changed by technology"(Clough 2013) [8].

Chapter 5

System Design

5.1 Start Menu

Figure 5.1

- The first thing you see when you open the application. The Opening screen page.
- Boasts a bright, colorful theme that repeats throughout the game to make the design more appealing to the target audience.
- Displays the company logo and previews different scenes from the game.

Figure 5.2

- Tapping anywhere on the screen brings you to the Main Menu
- Designed well, shows all the different sections of the application.
- Attractive to the eye. Captures children's attention.

5.2 Create and Play

Figure 5.3

- This is the first thing you see when you choose the "Create and Play section"
- Choosing create a story will bring you to the scene in the next screenshot.

Figure 5.1: Logo

Figure 5.2: Main Menu

- You can view previously made stories from the "View your Stories" selection.

Figure 5.4

- This is in the Create and play section.
- First, the user sets a name for their story
- While the user is naming the story, all required database calls are ran asynchronously to reduce waiting times.

Figure 5.5

- In this screen-shot you can see a few images.
- These are the scenes you can choose to create your story.
- We are using many new features that we haven't used before.

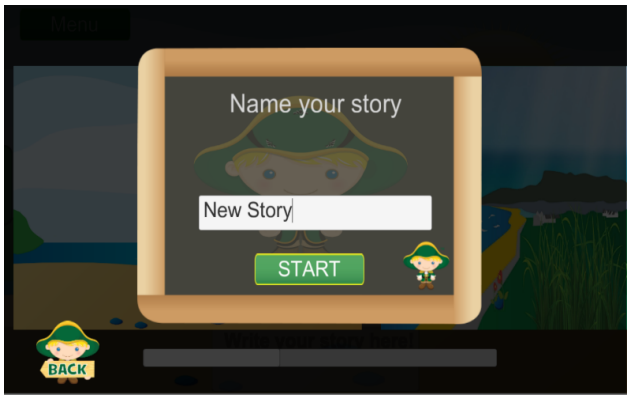

Figure 5.3: Create Name

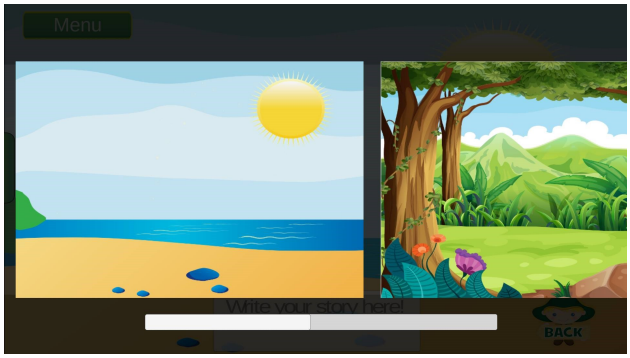

Figure 5.4: Choose scene

Figure 5.6

- This screen-shot shows the game in action.
- A few of the icons have been added to the scene using the menu sidebar
- Every time a scene is saved, a snapshot is saved of the scene (ignoring UI elements to prevent clutter) and saved locally.

Figure 5.7

- The user is brought to this scene after finishing a story or by selecting "View your stories"
- Selecting a story allows the user to scroll through each scene in the story.

Figure 5.5: Create/fill scene

Figure 5.8

- This shows how the user can cycle through the different scenes on a selected story.
- The green arrow buttons allow navigation through each scene.
- The story can be emailed to a friend if the user has their email address saved in the main menu settings

Figure 5.6: Email scene

Figure 5.9

- This is an example of an email sent using the application. The user is able to write a short message and the story scenes are sent as email attachments.

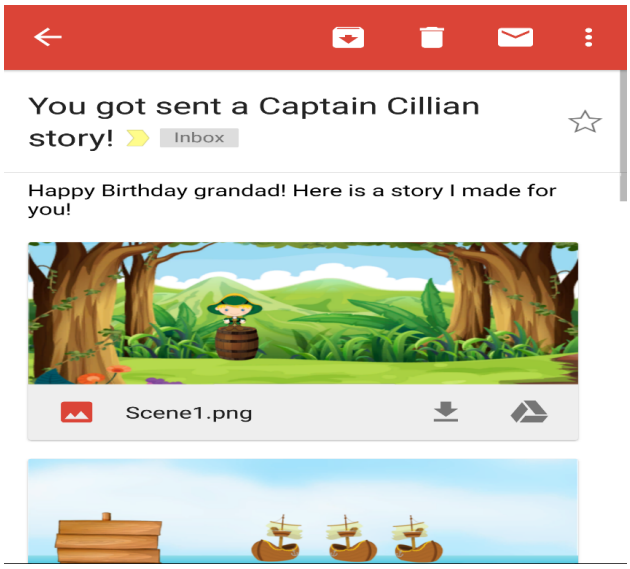

Figure 5.7: View Email scene

5.3 Learn Irish

Figure 5.10

- In this section we added a match the word and picture game.
- After researching this we found that it would be the most effective way for children to visualize and also learn Irish words.
- It will be useful for using in the classrooms as our client is visualizing this to be used in schools.

5.4 Ocean Facts

Figure 5.11

- This is the Pick and place puzzle game which displays one fact about one of the books.
- Smart approach in getting children to have fun and enjoy playing a game while also learning.
- This section implements sound .When the puzzle is completed.

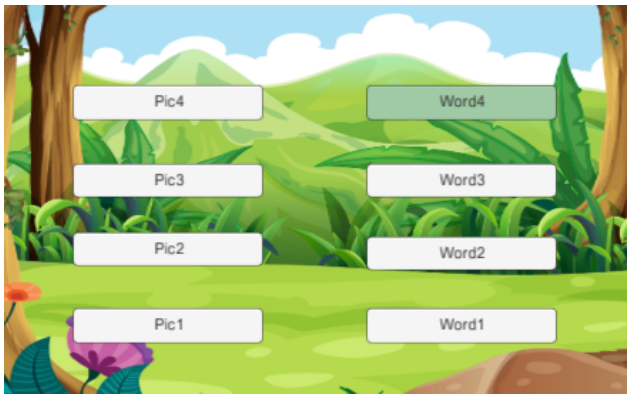

Figure 5.8: Learn scene

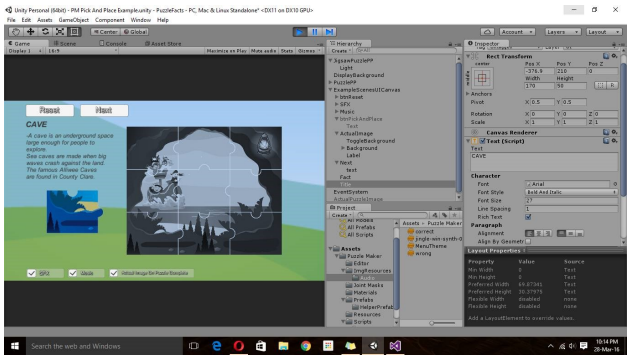

Figure 5.9: Puzzle one

Figure 5.12

- Added three different puzzles for each book.
- This is a sliding puzzle.
- At the bottom I decided to add a button that will bring you to one of the books.

Figure 5.13

- This is a Join the pieces puzzle.
- That also contains one fact about another book.
- Decided to add only one fact per book since the customer is using this as a promotion software.

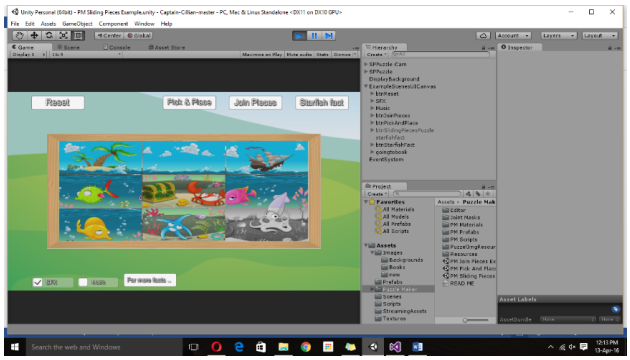

Figure 5.10: Puzzle two

- So in the button "For more fact..." customers will be brought to the website where they can buy the book to find out more facts.

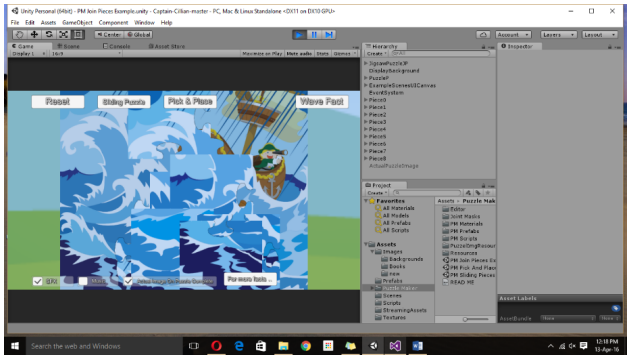

Figure 5.11: Puzzle three

5.5 Explore

Figure 5.14

- The explore section will only contain the selection of books that have been made.
- Each book will be linked to a website there the book can be bought.
- This is the best way we can promote the clients books and also the clients website.

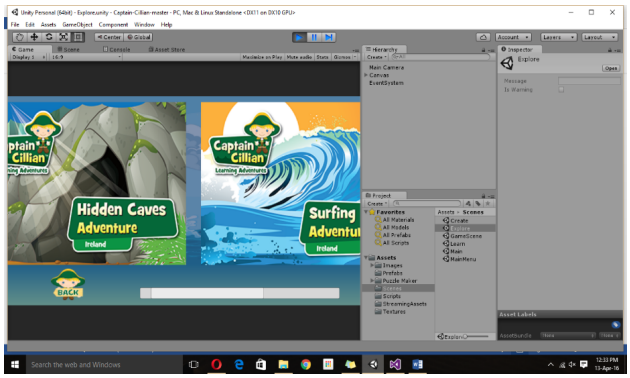

Figure 5.12: Explore

5.6 Database and uploading software/Additional Diagrams

The following screenshots show our database and how the database functions. We chose to use this particular database because it has a free trail for one year. Our client wasn't sure she wanted to use a database to upload more information on this application. She initially wanted to create a new app for each batch of books she created. But since we had to add a level of complexity to our project we decided to give her this option. After the year passes she has the choice of continuing to use this or to leave it aside.
Initially, the images themselves were saved as a Base64 string to the database, as DynamoDB uses Key/Value pairs to store data. However, downloading and converting each string proved to be too slow to allow for scalability, so the system was redesigned to use Amazon S3 (Simple Storage Service). The database was changed to store the image links and any additional information required. When the program is ran, it runs a scan on the relevant database table, and for each object in the table it finds, it saves and uses the link to search S3 for the equivalent file. For example, an image called "Test.png" will be saved like so in DynamoDB:
Theme: "Standard"
StickerName: "Test.png" When the respective section makes a call to the database, it will find "Test.png" and subsequently search for it under the folder "Standard" in S3.

In order to add to the database without logging in manually, a static page was made in NodeJS to allow uploading of additional images. The page makes a call to the database and returns a Set of each Theme in the database. These are then used to populate the "Select Theme" drop-down

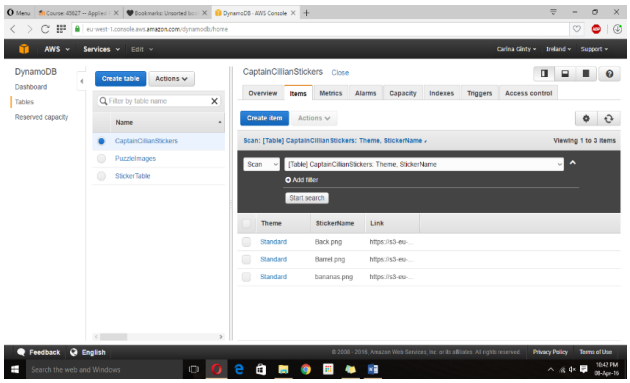

Figure 5.13: Database

list. The administrator can also create a new theme. The Upload button will then save the selected image using the given parameters to both S3 and DynamoDB.

Figure 5.14: Uploading Software

Chapter 6

System Evaluation

The PDCA Cycle- PDCA (plan–do–check–act or plan–do–check–adjust) is an iterative four-step management method used in business/teams for the control and continuous improvement of processes and products.

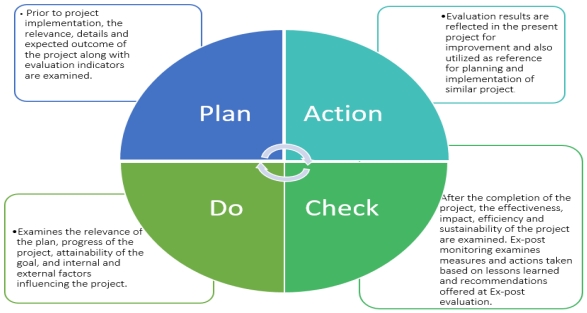

Figure 6.1: PDCA - Plan-do-check-action.More information about this diagram belob.

- PLAN: Establish the objectives and processes necessary to deliver results in accordance with the expected output (the target or goals). By establishing output expectations, the completeness and accuracy of the spec is also a part of the targeted improvement. When possible start on a small scale to test possible effects.
- DO: Implement the plan, execute the process and make the product. Collect data for charting and analysis in the following ”CHECK” and ”ACT” steps.
- CHECK: Study the actual results (measured and collected in ”DO” above) and compare against the expected results (targets or goals from

the ”PLAN”) to ascertain any differences. Look for deviation in implementation from the plan and also look for the appropriateness and completeness of the plan to enable the execution, i.e., ”Do”. Charting data can make this much easier to see trends over several PDCA cycles and in order to convert the collected data into information. Information is what you need for the next step ”ACT”.

- ACT: If the CHECK shows that the PLAN that was implemented in DO is an improvement to the prior standard (baseline), then that becomes the new standard (baseline) for how the organization should ACT going forward (new standards are enacted). If the CHECK shows that the PLAN that was implemented in DO is not an improvement, then the existing standard (baseline) will remain in place. In either case, if the CHECK showed something different than expected (whether better or worse), then there is some more learning to be done... and that will suggest potential future PDCA cycles. Note that some who teach PDCA assert that the ACT involves making adjustments or corrective actions... but generally it would be counter to PDCA thinking to propose and decide upon alternative changes without using a proper PLAN phase, or to make them the new standard (baseline) without going through DO and CHECK steps.

6.1 Testing and Robustness

—-CONTINUING FROM METHODOLOGY ON THE TESTING

- 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.

- 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.

- 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.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.

6.2 Performance Benchmarks

"Benchmarking is the systematic process of measuring one's performance against recognized leaders for the purpose of determining best practices that lead to superior performance when adapted and utilized."

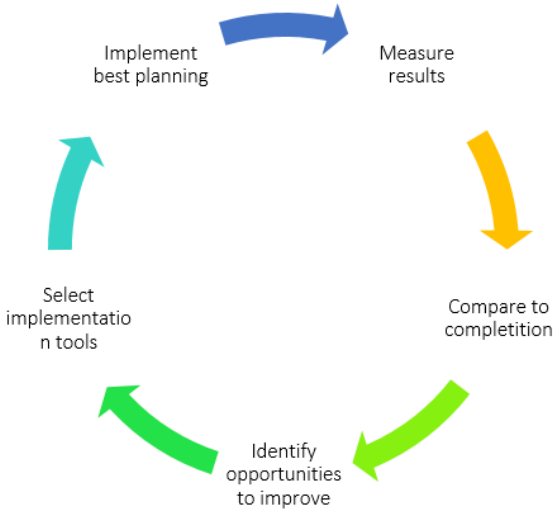

Figure 6.2: Performance Benchmarking Diagram

6.2.1 Determine what to Benchmark

What we are benchmarking is the performance of our application that was created in Unity. The method that we approached was placing our application on the android phone. This showed that our application was running. This process allowed us to see where the project was having loading problems. When approaching the section that connected to the database there was a long loading time, where the application was checking if new input was added to the database. The rest of the application had great loading time.

6.2.2 Defining the Measures and Solutions

To define the time that the application was taking to run when connecting to the database we ran the application on the phone. The loading time was about 20 seconds. It took the application 20 seconds to connect to the database add all the new input and play the game chosen. After researching how to fix this problem, Diarmuid decided to add a timer. The timer basically added a time limit on the loading time of the section. A 10 second limit was set, if the application didn't pull the data added in these 10 seconds the application starts the game without pulling the new assets added.

6.3 Comparing – Outcomes against Objectives

This section will hold the process of comparing the outcomes against the objectives we have set in the beginning. Every section of the application will undergo this procedure individually. This is in order to explain fully what each part was actually meant to do and what it came out in the end doing. The purpose of this section is to see all the solutions that were tried, all the work that went into doing each section and the end product. There were only three sections where games were added, therefore only these three sections will be explained in this section.

Create and Play

The Create Play section was from the outset meant to be a story creator application. The client had envisioned an interface similar to Toontastic, an iPhone application with a similar concept, where users can add objects and characters to design a scene and write a story based on their design. However, the design of the story creator was constrained due to a lack of professional animations and sound assets that give Toontastic a high level of polish.
The section began as a simple scene designer. The user chose a backdrop to set the scene and could then add a number of sprites to the scene to design the story. An input field was added to give the scene a story. All sprites are loaded in using Unity's built in Resources folder, so all assets in the scene are loaded dynamically when the section is selected.
In order to add a layer of complexity to the application, I decided to add a noSQL backend to allow additional content to be added to the application at any time. Using Amazon Web Services (AWS) I set up a DynamoDB (noSQL) table and an S3 (Simple Storage Service) bucket, or folder. The S3 bucket stores each new sticker to be imported into the scene as a PNG image file and the noSQL stores required info for each image, such as sticker name and its respective theme. Taken from the AWS Unity SDK, this function was used to pull objects from the database:

Taken from the AWS Unity SDK, this function was used to pull objects from the database: private void FindAllStickers(Dictionary¡string,AttributeValue¿ lastKeyEvaluated) var request = new ScanRequest TableName = "Captain-CillianStickers", ExclusiveStartKey = lastKeyEvaluated ; // Client scanned asynchronously // For each item in result, a new dictionary is made with the Attribute Value client.ScanAsync(request,(result)=¿ foreach (Dictionary¡string,

Figure 6.3: First attempt at this section. Basic information with images.

AttributeValue¿ item in result.Response.Items) // New sticker found, add info to new sticker object addItem(item);

The addItem() function takes in a dictionary of each item and creates a new sticker using the attribute values in each dictionary. The stickers created are then added to a separate dictionary, using the sticker name (i.e. image file name) as the key and the sticker object as the value. This information is used to pull the required sticker from S3. Each sticker is passed as a parameter into the getObjects() method.

string file = s.themeName + "/" + s.stickerName; S3Client.GetObjectAsync(S3BucketName, file, (responseObj) =¿ var response = responseObj.Response; if (response.ResponseStream != null) Texture2D t = new Texture2D(4,4); byte[] imageData = new byte[response.ResponseStream.Length]; // Use downloaded image bytedata to load the image t.LoadImage(imageData);

Figure 6.4: First attempt at this section. Basic information with images.

When the save scene button is pressed, the UI elements of the scene (menus, buttons) are made transparent and a RenderTexture object is made

using what is currently being rendered by the camera. I used this instead of the built in Application.TakeScreenshot() function for a number of reasons:

- The TakeScreenshot() function runs asynchronously, making it difficult to quickly hide the UI elements before the screenshot is taken.
- Unity's built-in screenshot capabilities tend to be saved in a lower resolution and is slower to load. Additionally, there is no option as to where the screenshot is to be saved. This means saving each screenshot in a specific folder is next to impossible.

The RenderTexture method was implemented as shown:
RenderTexture rt = new RenderTexture(Screen.width, Screen.height, 24); mainCamera.targetTexture = rt; mainCamera.Render(); Texture2D imageOverview = new Texture2D(mainCamera.targetTexture.width, mainCamera.targetTexture.height, TextureFormat.RGB24, false); imageOverview.ReadPixels(r, 0, 0); imageOverview.Apply();

CREATE AND PLAY – Story Viewer The second part of the Create Play section is the story viewer, where users can view their previously created stories. The stories saved using the RenderTexture method described above are written to Unity's PersistantDataPath, which can be used to save resources to any devices with a unity build. Each folder saved in this path will signal to the app that there is a story saved in it. The application brings each subdirectory into an array using:
string rootPath = Application.persistentDataPath; string[] directories = Directory.GetDirectories(rootPath); // Add file path for each scene in each story to an array string[] fileEntries = Directory.GetFiles(storyFolder);
Using this, the user can browse through the different saved stories in the scene. The stories are displayed using the first scene in the story (fileEntries[0]) as a thumbnail preview.

CREATE AND PLAY – NodeJS Sticker Uploader In order to allow the client to consistently avail of the Backend services without needing to manually add each item to both DynamoDB and S3, I wrote an image uploader using NodeJS, JQuery and Ajax.

When first loaded, Node makes a callback to the server and adds all existing themes in the database to a set. This populates the drop down list for easier use by the client. The client can also add an additional theme, if ¡New Theme¿ is selected the New Theme name input box becomes visible using AJAX. When Upload is clicked the following function is called:

Figure 6.5: First attempt at this section. Basic information with images.

Figure 6.6: First attempt at this section. Basic information with images.

Using the npm package 'Multer' the formdata (the image selected) is passed into the Node post function '/upload/:theme/', upload.single('sticker') The upload.single('sticker') parameter allows passing of a file as a parameter using an input from the HTML form with the id 'sticker'.

Learn Irish

Step 1 – Game Logic The Learn Irish section of this project is a basic match-up children's game, where you have to match the pictures with their corresponding Irish words. The aim of the game is to learn some new Irish words.

In this picture there are two buttons. The first button is the boat image on the left-hand side and the second button is the obvious button image on the right. To match-up the picture of the boat with the word 'Boat' you click either button followed by its match. Once a button is clicked it is highlighted green to let you know it's been selected/clicked.

This highlighted button can be seen in the image above (the button is

```
$('#frmUpload').submit(function(){
  if (newTheme) {
    selectedTheme = $("#newTheme").val();
    alert("Uploaded to " + selectedTheme);
  }
  var form = $("#frmUpload");
  var formData = new FormData($('#frmUpload')[0]);
  var otherData = form.serialize();

  var request = $.ajax({
   url    : form.attr('action')+'/'+selectedTheme,
   type   : form.attr('method'),
   // Form data
   data : formData,
   //Options to tell jQuery not to process data or worry about content-type.
   cache: false,
```

Figure 6.7: First attempt at this section. Basic information with images.

Figure 6.8: First attempt at this section. Basic information with images.

actually highlighted green in the game. The screenshot seems to have picked up the highlight as a greyish colour for some strange reason). When an Image button is clicked (like the boat image button above) the Irish word for that image is played. The idea is to help the child match both the image and spoken Irish word with the written Irish word. Thus learning Irish. The game is complete when you successfully match-up all the pictures with their corresponding Irish words.
The scripts I used for the Learn Irish game/ section:

- AWSManager2.cs
- AddImageScript.cs
- LearnGameScript.cs

The Unity scene is called:

- Learn.unity

Script: LearnGameScript.cs
The first thing I needed to do was set up the game logic. I added four picture buttons and four word buttons to the scene. These buttons would act as the images and words that you have to match-up. Initially these buttons were

Figure 6.9: First attempt at this section. Basic information with images.

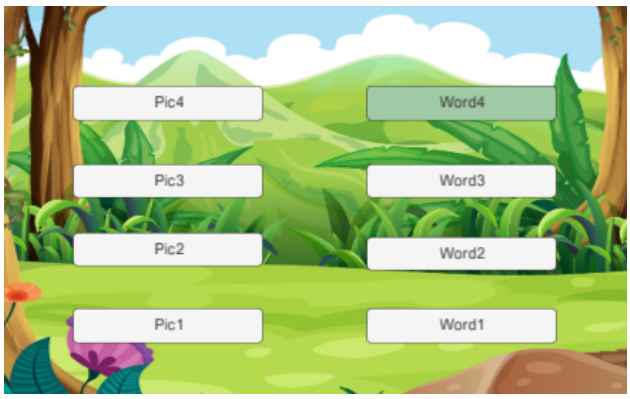

Figure 6.10: First attempt at this section. Basic information with images.

static and not randomised on the screen, as you can see from the image below.

Next step was to find out how I would go about making the images match with the words. I decided to use integer values to determine this: private int pic1 = 0; private int pic2 = 100; private int pic3 = 400; private int pic4 = 600; private int pic5 = 800; private int pic6 = 1000; private int pic7 = 1200;

private int word1 = 200; private int word2 = 300; private int word3 = 500; private int word4 = 700; private int word5 = 900; private int word6 = 1100; private int word7 = 1300;

These integers had a different value to begin with which would be changed/ set when their respective buttons were clicked: public void setPic1(int picNum) pic1 = picNum; public void setWord1(int wrdNum) word1 = wrdNum; In this instance both these functions would set the integer values of both buttons to the same value (i.e int value 10) . This way pic1 would equal word1 (match-up). if (pic1 == word1) butPic1.SetActive (false); butWord1.SetActive (false); endGame++; This worked fine but I had to find a way to reset the button value if the wrong match-up was created (pic1 with word3). So I had to create a function that would only be entered if two buttons were pressed and if the two buttons pressed were not equal, reset all the values. if(clickCount clickCount = 0; // clickCount reset at start of

function.

if (pic1 == word1) butPic1.SetActive (false); butWord1.SetActive (false); endGame++;

if (pic2 == word2)

butPic2.SetActive (false); butWord2.SetActive (false); endGame++; else // resets values if the 2 buttons are not equal. pic1 = 0; pic2 = 100; word1 = 200; word2 = 300; The int value clickCount controls the entry to this function. The clickCount value is incremented everytime a button is clicked: public void setClickCount() clickCount++; To determine if two buttons were clicked I used: if(clickCount So if the remainder of clickCount divided by 2 was equal to zero, this indicated the value of clickCount was 2 (button pressed twice). Next I needed to shuffle up the positions of the buttons on screen. I used two functions to do this:

- private void initializePics()
- private void initializeWords()

These functions work in the same way to shuffle the picture buttons and word buttons respectively. Here is an example of the initialize word function: private void initializeWords() // Looping through each word button foreach (Transform child in words.transform) yPositions.Add(child.position.y); xPositions.Add(child.position.x); List¡float¿ temp = yPositions; List¡float¿ temp1 = xPositions; int i = 0;

foreach (Transform child in words.transform)

if(i ¡ numsShuffle2.Count()) // assign each word button a random xPosition yPosition. child.position = new Vector2(temp1.ElementAt(numsShuffle2.ElementAt(i)), temp.ElementAt(numsShuffle2.ElementAt(i)));

i++;

In the function above we are looping through each word button and adding it's x-position and y-position to two lists:

- public List¡float¿ yPositions = new List¡float¿();
- public List¡float¿ xPositions = new List¡float¿();

Next we loop through these buttons again and assign an xPosition and yPosition from our lists. The position is randomised using a shuffled array of integers zero to six: (numsShuffle2.ElementAt(i).

Adding Images from Database Scripts:

- AWSManager2.cs

- AddImageScript.cs

The database I used to store the game images was DynamoDB hosted by Amazon Web Services. DynamoDB is a noSQL database which uses a key-value type of storage. I uploaded my images to the S3 cloud storage provided by AWS. This S3 bucket was accessed by my DynamoDB database table. To access the database from our app we needed to download the AWS SDK and add it to the project. This provided us with sample scripts which we could alter to access our database tables. My AWSManager2 script is the altered script I used to connect to the database and bring in the images. Once the images were downloaded they were saved to a Dictionary¡Key, Value¿:

- private Dictionary¡string, Sticker1¿ stickers = new Dictionary¡string, Sticker1¿();

The next step was to add these images to the game scene. I used AddImage-Script.cs for this. The images were added to a List¡Sticker1¿ downloadedImages (Sticker1 being the object).

- Loops through downloadedImages and adds them to the scene in the form of game objects.
- Adds these game objects to a game objects array: goArray = new GameObject[downloadedStickers.Count];
- An int array called numsShuffle is created to handle the randomising of the game objects.
- numsShuffle = new int[downloadedStickers.Count];
- numsShuffle is then shuffled(randomised) using the reshuffle function: reshuffle(numsShuffle);
- The button for picture 1 is assigned an image randomly from the game objects array at position numsShuffle: butPic1.GetComponent¡Image¿().sprite = goArray[numsShuffle[0]].GetComponent¡Image¿().sprite;
- The word 1 button is assigned the same name as picture 1.
- butWord1.GetComponent¡Image¿().name = goArray[numsShuffle[0]].GetComponent¡Image
- Then the text for word 1 is set to whatever the name of word 1 is.
- butWord1.transform.GetChild(0).GetComponent¡Text¿().text = butWord1.gameObject.na

Adding sound The sound in the game comes in the form of spoken Irish words. Every time an image button is pressed it triggers its corresponding Irish word. This is vital to the game as it gives the child a hint as to what the Irish word could be if they are unfamiliar with it. Boosting their chances of learning new Irish words. An object of type LearnGameScript was created in the AddImageScript. This object was used to call the function setTheIrishWord. This function was feed an input of type string with the name of whatever button it was adding sound to: learnGS.setTheIrishWord1(butWord1.gameObject.name.ToString()); Once the setTheIrishWord function was given the string it then passed on that string to an other string in the LearnGameScript:

public void setTheIrishWord1(string word) irishWord1 = word;

This string irishWord1 was then used to populate a function: getIrishWord (irishWord1). This function takes a string input and sets the sound clip according to the string. public void getIrishWord(string irishWord) AudioSource audio = GetComponent¡AudioSource¿ (); wordInIrish = Resources.Load (irishWord, typeof(AudioClip)) as AudioClip; audio.PlayOneShot (wordInIrish); This function is then called every time an image button is pressed: public void setPic1(int picNum1) pic1 = picNum1; getIrishWord (irishWord1); This game is set up in a way to accommodate for future expansion. If the client (Corina – Creator of Captain Cillian) wants to add or change the game images she will be able to without having to rebuild the app.

Ocean Facts

What this section was describes as in the specification: https://github.com/AliLigi/C Cillian- here is where i did all my commits. These commits were merged with the other team members and added to the main github link.
The Ocean Facts section draws from the theme of sailing with Captain Cillian. It contains multiple scenes (Currently just an aquarium scene) which can be navigated through like a slideshow. The scenes contains pictures of relevant animals and facts about each.

Objectives from the Introduction overall to be implemented:

- More quality and less quantity.
- Cross platform.
- Overall publishing clients books.

Objectives specifically for this section:

- Having one working puzzle called Pick and place with different facts.
- Making this section fun as well as adding facts.

Process of getting the final outcome:

1 Created a template page in unity with facts and images of sea creatures- The page looked somewhat like the picture below. To make sure I was going for what the client wanted, I set up a meeting and displayed this template to her. She was not very happy with just having information with a few images. This is because she was imagining the application to be used in a classroom. Therefore she asked for a word search, or a hand man that the children could play in classes.

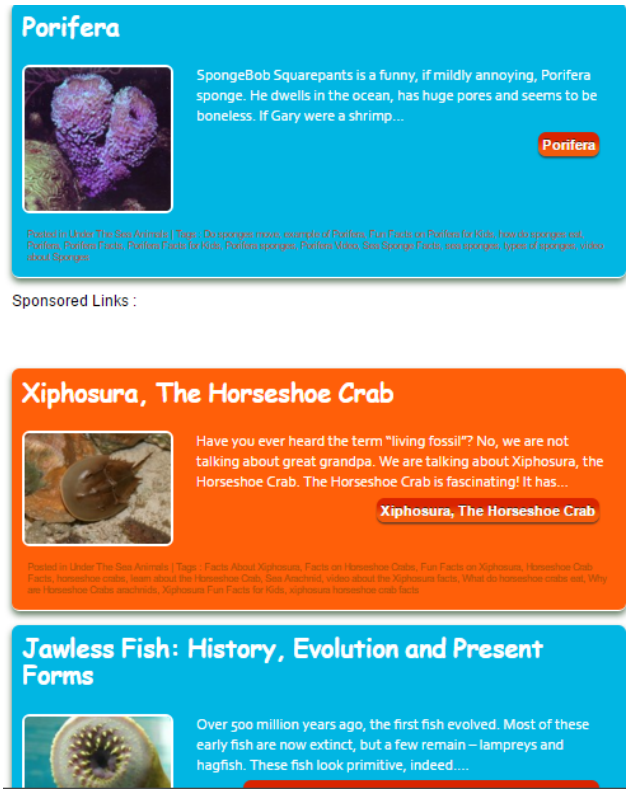

Figure 6.11: First attempt at this section. Basic information with images.

2 Made a word search: - after the meeting, I decided to make a word search and try to implement this to fit the application. But by making a word search there was no way I could add the facts she wanted to display. Therefore I had to go and research on more children's applications and games, where I could also implement some facts.

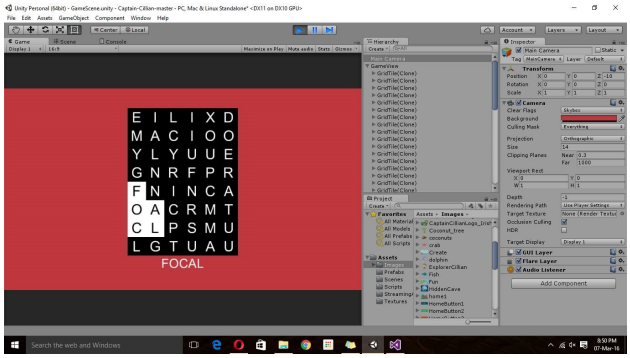

Figure 6.12: Second attempt at this section. Created a Word Game.

3 Puzzle Idea – after doing a few more days of research I found a great unity asset store package. This package contained a puzzle game. This brought me to the idea of creating one puzzle game and adding facts to it that the user can show and hide to their preference. This was a great idea and considering the time I had left to finish this project it was the idea I decided to implement in the end. https://www.assetstore.unity3d.com/en/!/content/25101

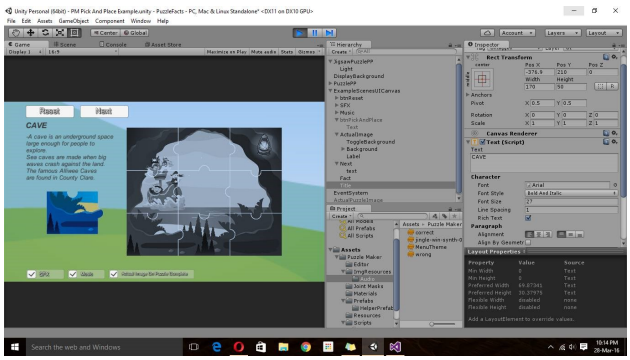

Figure 6.13: Final game choice. A puzzle game that also displays facts.

Outcomes:

- Choosing to do this puzzle was the best Idea in my case.

- It shows educational facts.
- It's fun and appropriate for children to play in classrooms as well as at home.
- Level of complexity is appropriate for the group of children we are targeting.
- Its cross platform.
- It has a really good quality because I used the unity assets package, so the game was basically done when I got it, it had sound and the movements were perfectly functioning. Therefore meeting the client's objectives.
- - Took a lot of trial and error but in the end it was worth all the research and time spent on it.

6.4 Limitations

- Time frame limitation: we were limited to the time we had to spend on the project. The first few weeks for us were more of getting the team members on bored with the project since it was ongoing from last year. A lot of research was done to see what the client didn't like about the last project and what we had to change. In our case we had to start the project again since we decided to use unity and to make the project cross platform. Some of the team members were not used to using Unity so that took extra work and extra time spent to understand how the software worked. Also the deadline 18th April.
- Software Limitations: since we had to do a cross platform application we were limited to what software we could choose. There is only a few amount of software that actually allow cross platform. We had a choice between Unity, Ionic and PhoneGap.

6.5 Opportunities

- Hardware/software evolution: we had the opportunity of working with new software that is in demand. These new software allowed us to create a more quality application, also a wonderful war to test this code by adding it to the phone and not changing the code. The new

online databases are very useful for projects created at the minute. Fee for a year it gives the clients you develop for an opportunity to try something new they perhaps never considered before.

- Using code that is already there and making it better: For example unity asset store , this gives opportunity to use code that is already made and that are very well made that are high in quality giving your application a polished look. Also GitHub had a lot of code samples you could take and change to suit your project.

Chapter 7

Conclusion

Since we are getting to the end I would like to point out the differences in the project this year and last year. This year we have achieved more of what the client wanted. We made an application that has more quality, unity in itself made this step possible. Also we have achieved to make the application cross platform which was a big objective for this application. We also added two extra features that were not even though about the year before. We have added a back-end to the project, a fully working database which will allow more information to go into the application. Also created a very simple uploading page at the end. This uploading page is used to make the clients work easier to upload assets onto the database.

We also had a better perspective of what the client wanted the application to look like since we had actual hard copies of the books she published and also a website. Reading these books and meditating on them made us come up with better ideas, better facts over all a more robust project.

This project challenged us to use new software. To adapt new things we have learned over the years. The amount of research that went into the project this year was definitely much more than last year. In the same time we pulled knowledge from other modules to achieve a better application. A few modules we have used knowledge from are: Research Methods, Semantic Web Development, and Mobile Application in c sharp, Database Management system and Distributed Systems.

Last year's project was mostly using software we have used in the three years, was not cross platform, it didn't look very professional and in the end it wasn't finished. Code wasn't used to its fullest potential. We wasn't any back-end to the project at that stage. There was an emphasis to get more

games than quality and that was not the objective to be met. Overall the team from this year made a better effort of meeting the client's objective and actually completing what was said would be done. We have used the technologies to the best of our abilities taking into consideration the opportunities we had. Taking into consideration all the ideas and opportunities given to us, we created a fully working project that is perfect for children to use.

7.1 Personal thoughts/How this project helped us personally

7.1.1 Alina Danci

- Personally this was the most stressful project up to date. A lot of work, research, time, meditating was put into getting this project finished to a quality standard. I feel like I have improved my team work skills during this project which will definitely help me in a real life job. I have also learned how to use existing code and adapting it to make a quality project. For me this was a huge research project. Finding out what games work for children, how I can adapt information to it since it is an edutainment application. I also focused on writing most of the documentation since my coworkers were better at coding then me. This way I discovered that I can write pages and pages and not get tired. This showed me I have a passion for writing. Enhancing my writing and research skills will also help me with writing on my developer blog, expanding ideas and maybe work with other developers to create some great work in the future. Every little thing I did this year prepared me for the real work environment outside of this college. I have worked on this project two years therefore I have gotten to see a lot of trial and error and a lot of frustration when code doesn't work the way it's meant to. That didn't take away from the satisfaction of finishing the project this year. Feeling like I did everything to make this client happy and meet her objectives. There were a lot of times I wanted to give up, I was pessimistic at the start thinking we can't make this project work. Perseverance was the key. As the time progressed and we kept getting things checked of the list and made me realize that everything is possible with time and research.

7.1.2 Diarmuid Byrne

- Throughout my four years in this course I have completed a lot of projects, from a lot of different modules. However, this project has been the highest scope by far, and I feel like I have not only met with the requirements set by the client, I have accomplished all the goals I hoped to achieve in undertaking the project. I learned a lot throughout development, not only soft skills such as working effectively in a team, timekeeping and project planning, but I have also greatly improved my technical experience. For the first time, I worked extensively in developing a reliable backend server and storage system that works cohesively with the frontend application. The iterative development that the project undertook meant a lot of research and re-hashing ideas to make them more suitable. For example, instead of using a SQL database to save images in blob format as originally intended, I opted for a noSQL database that complements an online storage service. It's worth mentioning that I have improved my eye for good design when it comes to high quality user-end applications. A very important aspect of this project was designing software that is easy to use for a specific target audience (in this case pre-teen children). Bright, colourful themes were used throughout the application and any required input and options were kept to a minimum to avoid confusion with the user-base. I have learned more working with this project (and other projects alongside it) in the last 8 months than I have since I began this course. I have become much more confident with the work that I produce and believe that will serve me well when working in the industry.

7.1.3 John Lavin

- I feel I gained a lot from the project this year. Not only did I enhance my coding and problem solving skills but I also improved other key skills such as researching and teamwork. A lot of work was put into research for this project. Spending several hours on StackOverFlow.com looking up code to putting my literature review from before Christmas to good use when designing my game. Getting the database to work for our game was by far the most difficult part. Several hours were spent on connecting the application to the database in order to access the images stored online (S3 cloud storage). This proved a very frustrating and timely task, but in the end I felt I had accomplished something worthwhile.

Working as part of a team has and always will be a necessary skill in the workplace. I feel myself, Diarmuid and Alina worked well together. We had weekly meetings to discuss progress and set out goals. We all improved our communication skills and always gave each other a helping hand when needed. We all came together to work out some of the more difficult parts of the project.
Working with the client gave me a valuable taste as to what a real world project could be like. Meeting up with the client and brainstorming ideas. Taking the clients specifications and turning them into a real product gave me a great sense of achievement. Overall I was happy with what I have accomplished and feel I've grow as a developer having completed this project.

Chapter 8

Client Review

The Captain Cillian software development team, have made good progress since 2015. They have enjoyed developing the Captain Cillian - Learning Adventure app and have learned many technical skills that will stand to them in their future careers. The brief was to focus on four themes and develop an activity/game under each theme/section. They split up the four themes and then each member developed an activity to link with the theme.

They have developed the app linked to a server that will enable it to be enhanced and updated in the future. There is great potential to further enhance the functionality of the App - for example the Explore Ireland section would benefit from facts and a fun activity linked to some of the story location details and the challenges set at the back of each book. There is potential to link this to a future development project.

The Irish words game needs to connect with a selection of the images and words in the books. This is something the team are working on for the final version. The jigsaw ocean facts section is presented well, however further scenes have been suggested that link to an illustration location from the book. The story board section contains some nice features. The flow and order of the story process is being revised to ensure a child can name their story from the start and add many scenes to their story. This section would benefit from a voice recording tool option also. It does contain a text box to type in words, however an additional feature, such as a voice recording would support a diverse learning group and enhance the experience in the classroom. Other features such as - a location/place name text box for or a map of Ireland built into each scene, might work well.

I'm looking forward to seeing the app completed and demonstrated from

start to finish.Well done to the team, I hope they have learned a lot from the experience.

Chapter 9

Bibliography

[**1**] A.
Author = ”Maria Virvou, George Katsionis and Konstantinos Manos”,
Title = “Combining Software Games with Education: Evaluation of its Educational Effectiveness”,
Year = ”2005”,

[**2**] B.
Author = ”Judy Robertson, Judith Good”,
Title = “Children’s narrative development through computer game authoring.”,
Year = ”2005”

[**3**] C.
Author = ”Ag. Asri Ag. Ibrahim ”,
Title = “Designing Software for Child Users: A Case Study of a Web Page Construction Kit for Children.”,
Year = ”2004”

[**4**] D.
Author = ”Sari Walldén, Anne Soronen”,
Title = “Edutainment: From Television and Computers to Digital Television.”,
Year = ”2004”

[5] E.
Author = "Asmaa Alsumait ",
Title = "Usability Heuristics Evaluation for Child E-Learning Applications.",
Year = "2010"

[6] F.
Author = "Stuart MacFarlane, Gavin Sim, Matthew Holton",
Title = "Assessing Usability and Fun in Educational Software.",
Year = "2005"

[7] G.
Author = "Stuart MacFarlane, Gavin Sim, Matthew Holton",
Title = "Assessing Usability and Fun in Educational Software.",
Year = "2005"

[8] H.
Author = "Clough, M.P, et al",
Title = "Nature of Technology: Implications for Learning and Teaching. ",
Year = "2013"

[9] I.
https://en.wikipedia.org/wiki/Unity$_{(}game_{e}ngine$)

[10] J.
http://docs.unity3d.com/Manual/AssetStore.html

[11] K.
http://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Introduction.html

[12] L.
https://en.wikipedia.org/wiki/Node.js

[13] M.
https://en.wikipedia.org/wiki/C$_{S}harp_{(}programming_{l}anguage$)

[14] N.
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Introduction