Table of Contents

1. Introduction

2. General Description

3. Functional Requirements

4. System Architecture

5. High-Level Design

6. Preliminary Schedule

1. Introduction

1.1 Overview

The product being developed for this project is simply a multiplayer game based on the popular television show countdown. This application will be available on the android app store to the public. We aim to make this a two player game as we feel it is in our scope.

The purpose of the application is to provide a feeling of nostalgia to the user. To achieve this, we want to implement many design cues from the shows. A good example of this is the classic countdown timer along with its music, creating an atmosphere of suspense and being part of the show.

There will be all the classic rounds like the letter, number and conundrum round. We aim implement the same points system as the game does as a means of determining the winner.

1.2 Business Context

There are two possible business contexts in relation to this project. These contexts are:

- **Selling the product**
  We could sell this product on the Android App Store at a set price. Multiple users downloading this application for a small fee could prove quite profitable providing the app is reasonably popular.

- **Internal Advertising**
  Another possible route for the app to take is internal advertising. A lot of free applications on the app store these days tend to go this route as a means of revenue. Throughout the application, adverts appear as banners, some even pause the app to display a full video advert with no option to exit it until the video plays out.

These contexts are both viable independently of each other, but we feel that these two used in conjunction with each other is possible but not a smart business move in market. These days no one wants to pay for an app and then be met with further adverts. They usually pay for an app to avoid such a thing.

1.3 Glossary

SQLite: **S**tructured **Q**uery **L**anguage is used to communicate with a database. SQLite is an in-process process library that implements a self-contained, server less, zero-configurational, transactional SQL database engine.

Peer to Peer Networking: This is a form of decentralized communications model in which each party has the same capabilities and either party can initiate a communication session.

Letter Round: This is a round within the countdown game. A player must pick nine letters to be used by all players, choosing from either vowels or consonants for each individual letter. Then the thirty second timer begins and the players have to make the longest word they can from these characters in the allocated time.

Number Round: This is a round within the countdown game. A player must pick six numbers. They can choose between a set of large or small numbers. The most large numbers you can choose is four. A randomly generated three digit number is chosen. The players must now use the chosen numbers to reach the target number using addition, subtraction, multiplication, etc. This is to be done within the allocated time.

Conundrum Round: This is a round within the countdown game. Nine random letters are shown, with players trying to make a word from all 9 letters within the allocated time.

Session: The period of time a user interfaces with an application. The user session begins when the user accesses the application and ends when the user quits the application. This also refers to the amount of time a user uses a website for. The session starts once the user logs in and finishes when the user logs out.

Google Play Game Services: This is what game developers use to create social leader boards and multiplayer games on the Android App Store.

2. General Description

2.1 Product / System Functions

Below is a list of the main functions of our project. This is a preliminary list and is open for additions should we think of anything worth adding.

- Sign in
- Join Game
- Randomly Selects Player
- Choose Opposing player
- Chosen player selects Letters
- Chosen Player Selects Numbers
- Letters round
- Numbers Round
- Conundrum Round
- Pause/Resume Game
- Display Scores
- End Game

*Sign up is not included as it is done externally by the google play games application on all android phones.

2.2 User Characteristics and Objectives

The application is available exclusively on the Android App Store. This will narrow our demographic to users with android phones only. It is also provided that the application supports the android version that the user has.

The target audience will be males and females between the ages of 12-60. The reason for the older demographic is because this is based on a game show that has an older audience due to it running for many years.

The application once downloaded will be a simple basic user-friendly interface to accommodate people with little or no knowledge of computing. There isn't a need for a complicated user interface as the app in its most basic form should be easy to grasp for this target audience.

2.3 Operational Scenarios

User Connects To Other Players:
From the main screen, the user will tap to join a game and then will wait while the application connects them to another player. Once connected, both players are given the option of starting the game. If unable to find another player after an allocated amount of time, the user will be prompted to try again or go back to the main screen.

User Pauses Game:
In the middle of a game session the user taps the option to pause. The pause screen shows up for the user but since this is a real time multiplayer game, the pause menu will not show up on the other player's screen, the game session will resume.

User Quits Game:
The user uses the option in the pause screen to exit the game. This brings the user back to the main menu of the application. The other player receives a notification on the screen and is then given a prompt to search for another game or return to the main menu.

User Selects Letters/Numbers:
For each round a different user will pick the nine vowels and consonants for the letter round or the six large or small numbers. Once the last letter or number is chosen, they are locked in and the round then commences.

Letter/Conundrum round:
These rounds are similar in that the user will have to drag and drop letters to the corresponding empty spaces to form the longest word possible. They have the option once satisfied with their answer to submit their answer early. Both users will then be scored.

Numbers Round:
This round involves the user dragging and dropping numbers into corresponding empty spaces but also to drag and drop specific mathematical operators in between these numbers to get to the target number. They have

the option once satisfied with their answer to submit their answer early. Both users will then be scored.

2.4 Constraints

Time:
We feel that time is the biggest factor in the scope of this project. Implementing a real time, fully working, aesthetic multiplayer is no small task.

Connection:
The application very much so depends on the user's internet connection, whether it be mobile data or Wi-Fi connection. If the user has a slow speed or is dropping connectivity, the game could be laggy. It could even get to the point of the session disconnecting.

Device Power:
While this may not be a huge problem these days, it is still not completely eradicated. With a game that can be visual, some devices may not be able to handle it with slow processors, little RAM and poor GPU. We aim to implement visual cues that need to be run smoothly. This requires a solid performing phone to do so.

3. Functional Requirements

3.1 Sign In To Google Play Games

- **Description**

This function is the very first thing that happens upon launching the application. It does not require the user to sign in themselves. This happens automatically by the application itself using the external application google play games which are installed on all android phones with google play services. Upon completion of this process the user will be brought to the main menu.

- **Criticality**

This function is a common aspect of multiplayer android games every time the application is launched. It is imperative to keep the competitive aspect to the game along with play games providing the ability to add other players, etc. This helps with the social aspect of the game.

- **Technical Issues**

Google provide the API to include in your application so you can use it as a template. We aim to implement this API for our finished application.

- **Dependencies**

None.

3.2 Join Game

- **Description**

Upon the initial google play games sign in. The user will have the option to join a game. Once choosing to do so, They will be brought to a connecting screen where they will wait for another player. Once connected to another player, both players will be brought in to a game session and wait for a short period of time for the game to begin.

- **Criticality**

This is one of the most critical aspects of the game we believe. Without this function, there is simply no way of implementing the multiplayer aspect of the game.

- **Technical Issues**

This will need to implement peer to peer networking in order to connect both players to each other. This is needed throughout the whole game session to maintain connection also.

- **Dependencies**

This depends on the sign in of the user.

3.3 Randomly Selects Player

Description
This function will be implemented at the start of every game. One player is chosen to pick the vowels and consonants for the letter round. They will then go on to choose them and lock in the letters for the round.

Criticality
It is not exactly extremely important to choose the player randomly but it does give a sense of fairness and non-bias when both players know that this is the way of starting the game session.

Technical Issues
This will use java and design elements with JavaScript to implement the visual choosing of the player.

Dependencies
This is dependent on the join game, connecting the users and maintaining a connection throughout.

3.4 Choose Opposing Player

- **Description**

After the first round of the game is finished and for all the following rounds afterwards, this function rotates the player who chooses the letters or numbers each round so a player never gets to choose for more than two consecutive rounds. They will go on to choose any letter or number for the round.

- **Criticality**

This is more important than the randomly chosen player. It is important to stay true to the game show and make sure that the players choosing are rotated after each round. Also to add to the fairness of the game

- **Technical Issues**

The same as randomly chosen player, it will use java and design elements with JavaScript to show the selection of the player.

- **Dependencies**

This is dependent on join game and randomly chosen player in order to choose the opposing player.

3.5 Chosen Player Selects Letters

- **Description**

Following on from the selection of the player, they will be prompted to choose nine letters. This will range from either vowels or consonants. Once this is finished the letters will lock in. They will display on both players screens as they are being chosen to allow for fair play. Once locked in, the letters round will then commence.

- **Criticality**

This is important to have as there will be no user chosen vowels or consonants to use for the letters round itself.

- **Technical Issues**

This will be using JavaScript to implement visual cues for the user to select vowels or consonants.

- **Dependencies**

This relies on the randomly chosen player or opposing player function along with join a game.

3.6 Chosen Player Selects Numbers

- **Description**

Following on from the selection of the player, they will be prompted to choose six numbers. This will range from small or large numbers. They can only choose four large numbers though. Once finished these numbers will lock in. They will display on both players screens as they are being chosen to allow for fair play. Once locked in, a target number will be made and the numbers round will then commence.

- **Criticality**

This is important to have as there will be no user chosen numbers to use for the numbers round itself.

- **Technical Issues**

This will be using JavaScript to implement visual cues for the user to select small or large numbers.

- **Dependencies**

This relies on the randomly chosen player or opposing player function along with join a game.

3.7 Letters Round

- **Description**

The letters round will commence. Both users will have thirty seconds to choose from the chosen letters to make the longest word they can. Once they have chosen their word they have the option to submit, if both users submit the round ends immediately to do scoring. Otherwise it waits on the thirty second timer. The words are then checked for validity and then appropriate scoring will be given out to the players.

- **Criticality**

This is one of the most important aspects to the game itself in order to play.

- **Technical Issues**

This will use java and JavaScript to calculate scores and validity along with visual cues to the round.

- **Dependencies**

This is dependent on the chosen player selects letters function along with join a game.

3.8 Numbers Round

- **Description**

The Numbers round will commence. Both users will have thirty seconds to choose from the numbers given to reach the target numbers using different mathematical operations. Once they have chosen their mathematical operations they have the option to submit, if both users submit the round ends immediately to do scoring. Otherwise it waits on the thirty second timer. The numbers are then checked to see how close it is to the target number and then appropriate scoring will be given out to the players.

- **Criticality**

This is one of the most important aspects of the game itself in order to play.

- **Technical Issues**

This will use java and JavaScript to calculate scores and validity along with visual cues to the round.

- **Dependencies**

This is dependent on the chosen player selects numbers function along with join a game.

3.9 Conundrum Round

- **Description**

The Conundrum round will commence. Players are shown a combination of two or three words with a total of nine letters. They have thirty seconds to form a single word using all the letters, and must submit before the thirty seconds are up. Whoever submits first gets the first option to check if their word is valid. If not the next user gets their word checked provided they submitted in time. Appropriate scores will be given out.

- **Criticality**

This is one of the most important aspects of the game itself in order to play.

- **Technical Issues**

This will use java and JavaScript to calculate scores and validity along with visual cues to the round.

- **Dependencies**

This only depends on join a game as it does not need users to choose letters.

3.10 Pause/Resume Game

- **Description**

Want to change some settings in the middle of the game session? Click to pause the game and a menu will come up to change settings. The game however will continue as this is a real time multiplayer game and the other user does not wait for you. When finished, the user can resume playing the game.

- **Criticality**

While not crucial, it is important to give the user the option to change some settings to the game and how they want the application to work.

Technical Issues

This will be implemented in java and possibly with the help of googles API.

Dependencies

This only requires the join a game function to maintain connection with the other player.

3.11 Display Scores

- **Description**

While Scores will be displayed throughout when the game is finished, a final score will show up on both the users' screens. It will then show the winner of the game.

- **Criticality**

This is quite important as every game needs to have a winner.

- **Technical Issues**

This will use java and JavaScript to decide on a winner and also use heavy visual cues to show the winner, which is important to add to the user's satisfaction of winning or dissatisfaction of losing.

- **Dependencies**

This relies on the join a game to maintain connectivity. Also it relies on the letter, number and conundrum round to calculate scores.

3.12 End Game

- **Description**

This is either done by the user from the pause menu or done when the game is finished. The game session has ended and both users will be brought back to the main menu where they started with the option to play again, etc.

- **Criticality**

This is important to give the option to the user to exit the game and go to the main menu and also to finish a game. The game session has to end in some way.

- **Technical Issues**

This will be done most likely with Googles API and built in functions to end a game session.

- **Dependencies**

This is dependent on join a game because you can't leave a game you did not join in the first place.

4. System Architecture

4.1 System Architecture Diagram

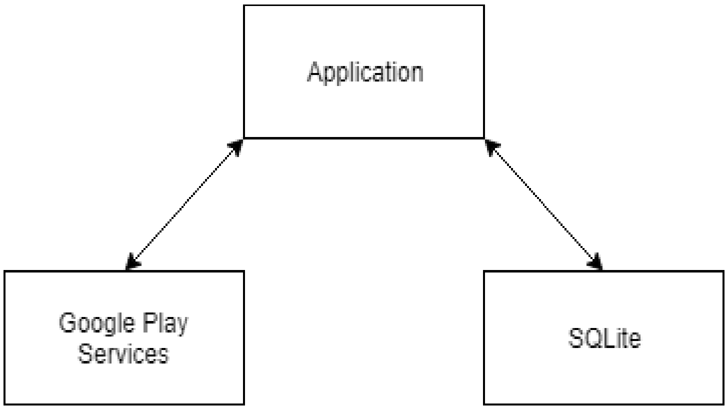


Fig 4.1 Shows the architecture of the product. The above diagram show there are three major aspects to the architecture of the system. The application (front end, what the consumers see), the Google Play Services (which allow users authentication and peer-to-peer multiplayer) and the SQLite (used for the dictionary to authenticate words).

4.2 Application

The application is the front end of this multiplayer game. Consumers will see and interact with the application to gain access to other elements of the architecture. The application will contain the content that displays the game in an aesthetically pleasing way. The application will also contain the computation behind the numbers round and the functionality of the game.

4.3 Google Play Game Services

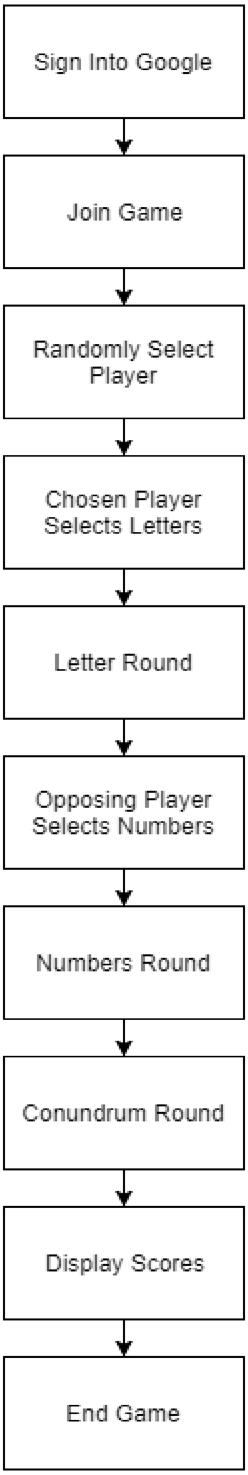
Google Play is being utilized as an authentication service for the multiplayer game that will allow users to connect to other players using the Google Play peer-to-peer service and real-time play.

4.4 SQLite

We plan to use SQLite as the dictionary for the game. It would be used to check whether a user input is a valid word in the letter round and return a true or false value.

5. High-Level Design

5.1 High Level Design Diagram



5.1

5.2 High-Level Design Description

Fig 5.1 is explained below.

- **Step 1: Sign into Google**
  User signs into Google Play Game Services using their Google account.

- **Step 2: Join Game**
  Player connects to a game instance using peer-to-peer connection.

- **Step 3: Randomly Select Player**
  One of the players is randomly selected to choose letters for the letter round.

- **Step 4: Chosen Player Selects Letters**
  The randomly selected player chooses the letters for the letter round.

- **Step 5: Letter Round**
  Players have to make the longest word they can using the letters selected within the 30 second timer.

- **Step 6: Opposing Player Selects Numbers**
  The opposing player chooses the numbers for the number round.

- **Step 7: Numbers Round**
  Players must try reaching the target number using the numbers selected and basic operators such as addition, subtraction, multiplication, etc.

- **Step 8: Conundrum Round**
  Players must unscramble the word using the letters provided.

- **Step 9: Display Scores**
  Display the scores of each player at the end of a full game.

- **Step 10: End Game**
  End the game between users.

6. Preliminary Schedule

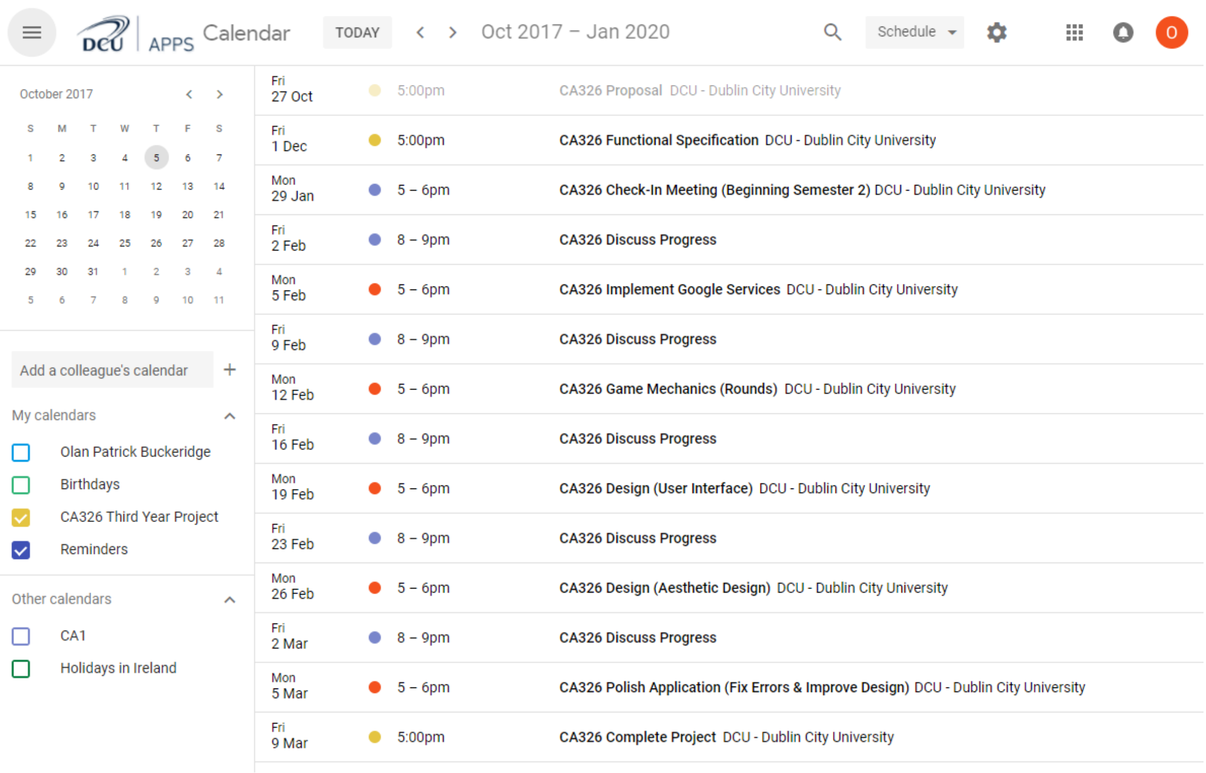
6.1 Overview of Preliminary Schedule

The schedule below was designed using Google Calendar and Google Sheets. **Fig 6.2** shows a full list of tasks and **Fig 6.3** shows a Gantt chart which is how and when these tasks are to be completed.

We chose to use Google Calendar as it integrates with DCU Apps and fits into the ecosystem we are using. It's easy to share and edit each other's content and we can schedule video calls using Google Hangouts as seen in the task list in Discuss Progress sections. Colour codes were used to display different tasks for example orange is deadlines for tasks and lavender is meetings. We feel like it was a better solution rather than Microsoft Project to track and complete our project.

The Gantt chart is a visual display of the information in the task list. We created this using Google Sheets. It adds a nice display to see where we are at in the production of the app and keeps us on schedule.

6.2 Task List



October 2017

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4
5	6	7	8	9	10	11

My calendars

- □ Olan Patrick Buckeridge
- □ Birthdays
- ☑ CA326 Third Year Project
- ☑ Reminders

Other calendars

- □ CA1
- □ Holidays in Ireland

Fri 27 Oct	●	5:00pm	CA326 Proposal DCU - Dublin City University
Fri 1 Dec	●	5:00pm	CA326 Functional Specification DCU - Dublin City University
Mon 29 Jan	●	5 – 6pm	CA326 Check-In Meeting (Beginning Semester 2) DCU - Dublin City University
Fri 2 Feb	●	8 – 9pm	CA326 Discuss Progress
Mon 5 Feb	●	5 – 6pm	CA326 Implement Google Services DCU - Dublin City University
Fri 9 Feb	●	8 – 9pm	CA326 Discuss Progress
Mon 12 Feb	●	5 – 6pm	CA326 Game Mechanics (Rounds) DCU - Dublin City University
Fri 16 Feb	●	8 – 9pm	CA326 Discuss Progress
Mon 19 Feb	●	5 – 6pm	CA326 Design (User Interface) DCU - Dublin City University
Fri 23 Feb	●	8 – 9pm	CA326 Discuss Progress
Mon 26 Feb	●	5 – 6pm	CA326 Design (Aesthetic Design) DCU - Dublin City University
Fri 2 Mar	●	8 – 9pm	CA326 Discuss Progress
Mon 5 Mar	●	5 – 6pm	CA326 Polish Application (Fix Errors & Improve Design) DCU - Dublin City University
Fri 9 Mar	●	5:00pm	CA326 Complete Project DCU - Dublin City University

Fig 6.2

6.3 Gantt Chart

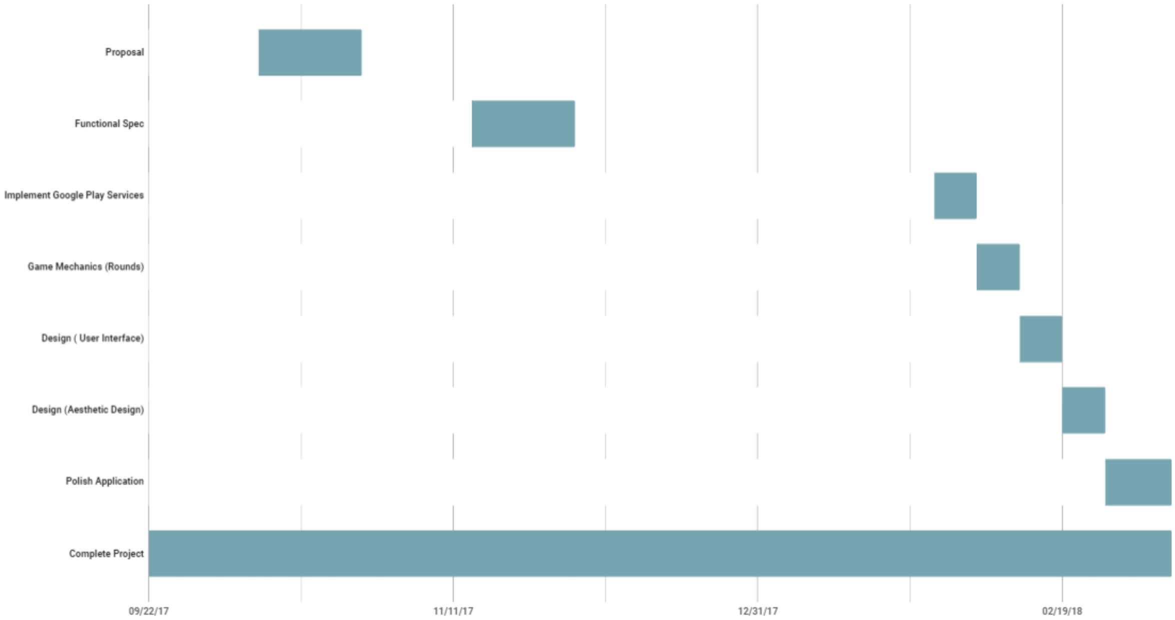


Fig 6.3

7. Appendices

Research Tools

- https://en.wikipedia.org/wiki/Countdown_(game_show)
- https://www.sqlite.org/about.html
- https://www.google.ie/search?q=google+calendar&oq=google+calendar&aqs=chrome..69i57j0l5.5727j0j4&sourceid=chrome&ie=UTF-8
- https://www.w3schools.com/sql/
- http://www.umlet.com/
- https://support.google.com/googleplay/?hl=en#topic=3364260
- https://developers.google.com/games/services/android/realtimeMultiplayer