

# COMP 4521 Project Report – MemMatch!

Group 16

Name: CHENG On Kiu

Student ID: 20516055

Email: okcheng@connect.ust.hk

## 1. Introduction

Because of brain aging, elderly people may find themselves forgetting things easily, some of them may be suffering from Alzheimer's disease. To prevent brain aging, doing a brain workout is needed, playing brain games such as Sudoku, Crosswords can help. This project aims to design a memory card matching game application to train users' memory, response and cognitive functionality. The app is suitable for all ages.

## 2. Design and Implementation

### 2.1 Requirement Analysis

- Provide a memory card matching game that has two game modes (*Basic*, *Timed*) and three levels of difficulty (*Easy*, *Medium*, *Hard*)
- Obtain player data from existing Google account
- Use colorful pictures as matching pairs
- Single player mode

### 2.2 Design

#### Installation requirement

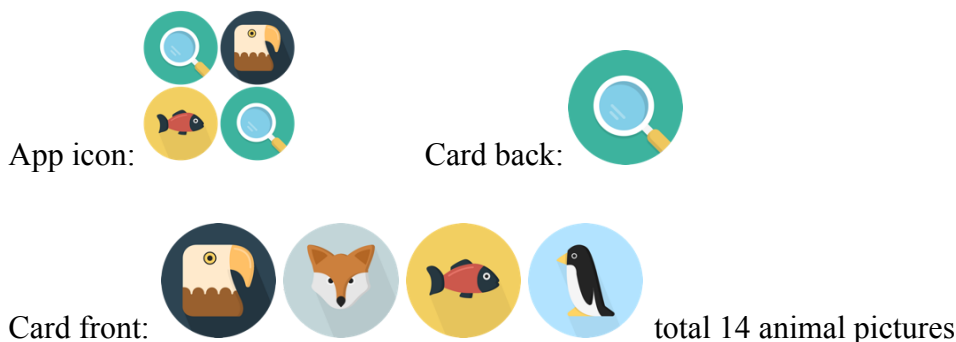
- minSDK: 21
- Compatible with phones with all pixels

#### Data storage

- **Firestore** [1] store player's Google account name, email address and ID
- Internal storage for the pictures

#### Specific features

- Use pictures from **Wikimedia Commons** [2] as the app icon, front and back sides of the memory cards



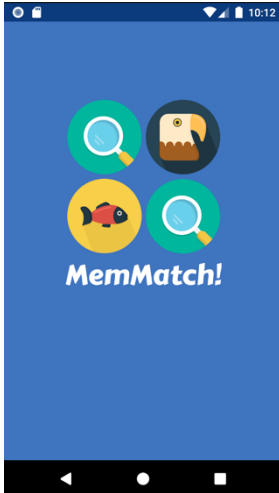
- 2 Gifs from **Lottie Files** [3] as animation when player win/lose the game



- Background of the layout is generated by SVG STRIPE GENERATOR [4]

## Activity Layouts (All without Action Bars)

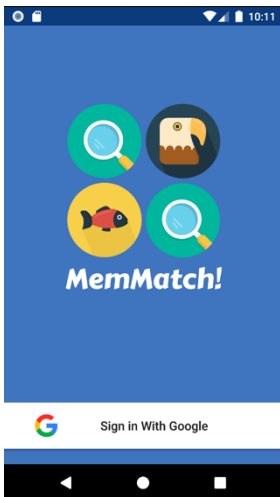
### Splash Activity



← Image view: app icon

← Text view: app name MemMatch!

### Login Activity

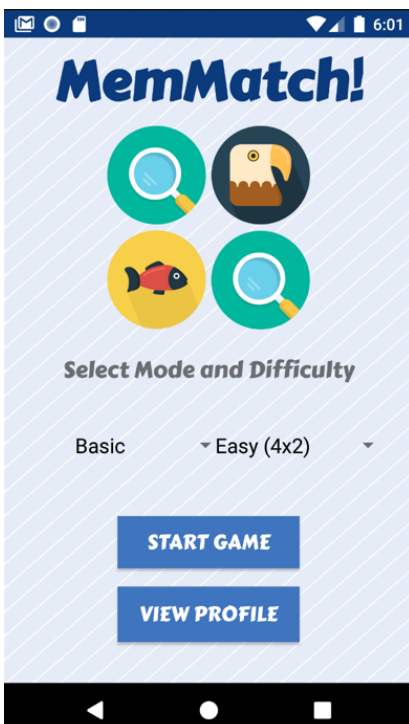


← Image view: app icon

← Text view: app name MemMatch!

← Button: sign in Google Account

### Menu Activity



← Text view: app name MemMatch!

← Image view: app icon

✓ Text view: Select Mode and Difficulty

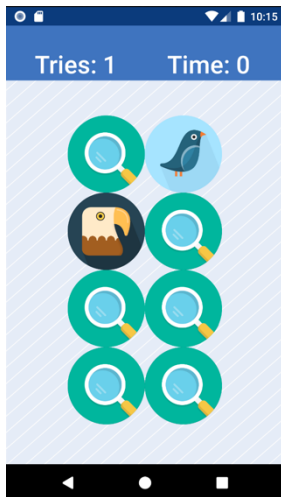
✓ Spinner button (left): Choose *Basic mode/ Timed mode*

← Spinner button (right): Choose Difficulty *Easy (4x2)/ Medium (4x4)/ Hard (6x4)*

← Button 1: Start a new game

← Button 2: View the profile

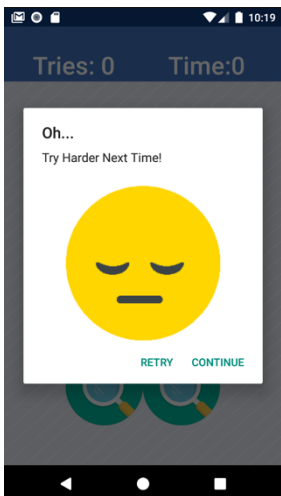
## Game Activity



← Text view (left): current number of tries

↖ Text view (right): time left in *timed mode*

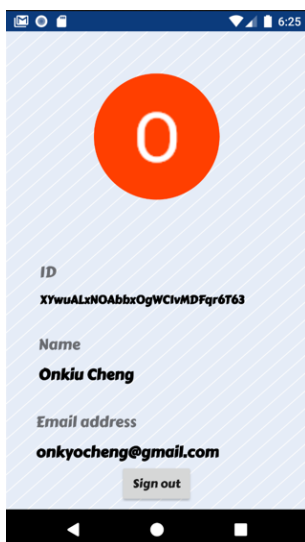
← Grid layout with image view of animal pictures inside



Alert Dialog:

- Win/lose title
- Win/lose message
- Layout with win/lose gif image view inside
- Retry game button
- Continue button

## Profile Activity



← Circular Image View: Google account profile picture

← Text view: Google account ID

← Text view: Google account name

← Text view: Google account email address

← Button: sign out Google Account

## 2.3 Implementation

### Splash Activity

- Use the handler delay function to give time to the Firebase authorization function to check whether a user has logged into his/her Google account or not.
- If yes, start Menu Activity, else start to Login Activity

### Login Activity

- Use button to trigger Google login
- Login Google account by using Firebase authorization
- If login is successful, save the account data in shared preference and start Menu Activity

### Menu Activity

- Allow users to choose game mode and difficulty by spinner buttons, selected items will give out a key variable respectively
- After pressing the start game button, the game mode and difficulty key variables will pass to Game Activity and starts Game Activity
- Pressing the view profile button will start the Profile Activity

### Game Activity

Card set up flow:

1. Load animal image name array
2. Set gridlayout size according to game difficulty key values (Easy: 4x2, Medium: 4x4, Hard: 6x4), add an image view to each child grid
3. Choose image id randomly and arrange their position in a grid layout: for example in Easy level, randomly put 4 animal image names in a new array, double the id elements and shuffle their order. Name the tag of the image view in each child grid by the selected animal image source name sequentially. Set all image views in grids to be the card back image.
4. If the game mode is *Timed*, create a countdown timer (Easy: 30s, Medium: 50s, Hard: 80s)

Game flow:

1. When the player clicks an image view, its image resource name will be equal to the one saved in its tag (flip card). If the game mode is *Timed*, the timer will start.
2. If a card is clicked repeatedly, the app will broadcast a repeated card message
3. When two cards are flipped every time, the number of tries will increase by one (shown in text view). If their tag names are equal, they are no longer clickable and become invisible. If not, their image resource will be a card back image (flip back).
4. After every click, the system will check if all image view in grids is not clickable. If yes, the game ends, and the win alertdialog will pop up. It shows the total number of tries, a gif and retry/continue buttons. The retry button will repeat the card setup process and restart the game, continue button starts Menu Activity.
5. In *Timed* mode, when the countdown timer finishes and there are clickable image views left, the lose alertdialog will pop out.

## Profile Activity

- Show Google account information (profile picture, ID, name, email address)
- Sign out button will sign out account and start Login Activity

## **Testing and Evaluation**

The app is being tested in the emulator and real device (Android 10, pixel 3), the evaluation will be focused on testing through the real device. Monitored by the profiler, there is no CPU usage used by the app, the memory usage keeps around 150 MB, and energy use is very light. There is no network activity except login and logout activities. The app gives a very light workload to the real device. There are very few error logs.

The app runs smoothly even if it involves a certain number of image views. Pressing the back button can direct to the correct activity with no redundant activity and running program left, other action bar buttons also work well. The app will not be collapsed in all situations tested such as rotating the device to landscape orientation (app orientation is set to portrait), leaving the app inactive for a long time and being interrupted by phone calls or message notifications.

The game flow is reasonable. Since the card images are colorful and large enough, players can nicely distinguish them. Most of the text views have suitable font sizes except the spinner buttons, users who have bad eyesight might find it hard to read.

## **3. Conclusions**

This application fits with its aim and requirements, it creates a memory match game interface with different numbers of cards and game modes. Pictures are colorful and attractive for players. However, the application is quite simple. Originally this app should contain a leaderboard activity that saves and gets users' information and game records from the Firebase, but it could not be finished on time. This makes the login activity useless. time l database The game experience can be improved by adding sound effects.

## **4. Reference**

[1] Firebase Authorization

<https://firebase.google.com/docs/auth/android/google-signin>

[2] Wikimedia Commons

[https://commons.wikimedia.org/wiki/User:CFCF/Flat'n'round#Mon\\_Mar\\_14\\_16:43:38\\_CET\\_2016](https://commons.wikimedia.org/wiki/User:CFCF/Flat'n'round#Mon_Mar_14_16:43:38_CET_2016)

[3] Lottie Files

<https://lottiefiles.com/>

[4] SVG STRIPE GENERATOR

<https://svg-stripe-generator.web.app/>