CE881 Assignment 1

# Assignment 1 Report

Sudoku Puzzle & Solver App

Student ID: FK18726
Reg Number: 1804162
2-3-2022

Page Limit: 5 Pages of Content
Pages: 5 Pages (5 pages of content + title page, table of contents, figures, and references)

Reg Number: 1804162

## Contents

Reg Number: 1804162

# Introduction

## Problem

This application attempts to address two problems: the problem of entertainment as users of this app will wish to have access to puzzles to keep themselves occupied; and the problem of users not having access to existing puzzle solutions.

## Purpose

This applications purpose is to allow the user to complete randomised digital sudoku puzzles, as well as allow them to check solutions to puzzles that are completed using other media, such as in newspapers.

## Main Features

The main features of my application consist of the following:

- Sudoku Puzzle
- Sudoku Solver

## Business Model

I believe that the best business model for my application will be by implementing advertisements. This is because due to the popularity of Sudoku, there exists multiple different applications and platforms that one could use; most of which are not paid downloads. Using advertisements would allow users to freely access the app; removing any risk on their side if the app is undesirable, which in turn makes them more likely to keep it installed.

I will, however, be using them sparingly; only having them active after a specified number of puzzles are completed by the user, or after a specific number of puzzles are solved by using the solver. This will hope to limit the annoyance the user will experience by having their session interrupted. The specific form of adds that will be used are ‘interstitial advertisements’, where the advertisement takes up the full screen. This allows the main application views to remain clean, making them more appealing to the user.

## Problem Statement & App Goal

| | |
|---|---|
| **Problem** | I want to have the ability to both complete digital puzzles, and to check the validity of my physical puzzle solutions. |
| **Solution** | This app provides both randomised digital sudoku puzzles, as well as a solver that allows the user to check if their solution to a physical puzzle is correct. |

# Background

## Idea Generation & Idea Validation

When thinking of an idea for my app, I first started to think about what I would like to see in an application, or what applications that I would likely use. This was my starting point, as to create an app that I would remain invested in during development, and to know what features may or may not feel useful, I would have to have some interest in the application besides just the development side.

I came up with the following application ideas:

- Journal App
- Home Inventory-Keeping App
- Puzzle Game App

Whilst a journal app would be the simplest application to create for this assignment, I did not want to base my application around this premise, both to ensure that I wouldn’t become disinterested with the project, and to ensure that the complicity of the app was suitable for this assignment’s marking scheme. Furthermore, journals are supposed to feel ‘personal’; meaning that these will almost exclusively be written on physical media. For these reasons, I decided to discard this idea.

A home-inventory keeping app would likely see more use and utility, however as with many utility apps like this, it is completely dependant on the user. Whilst it may start out fine, the user may begin to slowly be less and less consistent with the app as it will require a lot of input from them. This limits the effective market that I will have for this app, so for this reason I have discarded the idea.

The last Idea I had was to implement a puzzle game. As games such as Sudoku and Chess have seen a sharp spike in interest in recent years with popular players taking their games to video and streaming sites such as YouTube and Twitch, the market size for these kinds of apps has heavily increased. This will also allow me to implement features that I wish were present within many of the apps that I have tried; and provide me with a starting point as these games have a well-established identity; so, should not be changed. Of these, I am much more familiar with how Sudoku work, so had refined my idea to a Sudoku puzzle app.

The relative popularity of Sudoku over the past five years can be seen in **Figure 6**: Relative popularity of 'sudoku' search term over the past five years, where a popularity of ‘100’ is the peak popularity within the time scale. As can be seen, there has been a gradual and reasonable stable increase in interest over time, indicating that there is likely to be a large audience for an application like this.

For these reasons, I had decided on a Sudoku puzzle app.

With this, I had also decided to include a puzzle solver. This provides an opportunity for people to learn, as they can check puzzles that have been completed via external media that may not have solutions provided. This increases the potential audience for my application.

## Market Research

### Similar Apps

As Sudoku is a well-known game, there are hundreds of apps that allow the user to play. I have selected what I can see as the best selection for both sudoku games and solvers from the google play store.

- ‘Sudoku.com - Classic Sudoku’ by Easybrain [1]
- ‘Sudoku – Classic Sudoku Puzzle’ by Beetles Studios [2]
- ‘Sudoku – Classic Sudoku Puzzle’ by Guru Puzzle Game [3]
- ‘Sudoku’ by Brainium Studios [4]

These are the top four most downloaded sudoku puzzle apps available on the play store, with the number of downloads ranging from fifty million to ten million. Other sudoku puzzle apps on the play store don’t reach anywhere near this; being around fifty thousand to one hundred thousand downloads. That indicates that there is clear preference to what these apps can offer, so these should be the apps that are focused on.

The feedback on these apps is also similar, with most of the negative points being related to the developers over-complicating features of the app, or a too high frequency of advertisements. This is something that I wish to avoid.

There is also one app that I use and am more familiar with, that has a little over one million downloads on the play store: ‘Sudoku – The Clean One’ by Dustland Design [5]. This is an app that I personally use. It uses a simple and clear layout, with minimal interruptions to the user’s session. It uses an advertisement-based business mode (using full-screen interstitial advertisements); however, offers the user the chance to remove these ads with an in-app purchase. As advertisements are reasonably spaced out however, this is often not necessary.

However, none of these apps to my knowledge include a solver feature as part of the app itself; instead relying on other apps to specifically solve this purpose. This is something that could let my app stand out.

### Market Size

As Sudoku is a well-known game, and with the increase in popularity of games such as Sudoku on video and streaming websites, there is a large audience that can be reached with an app like this. This is supported by the high number of downloads for each of these apps, as people are unlikely to have installed multiple similar apps; indicating that there is a demand for applications like this. Furthermore, implementing a solver feature as part of the app itself would see more interest from both users wanting both a puzzle and a solver, as well as those just wanting a solver. The latter users would also have a possibility of deciding to use the puzzle feature as their main app at a later point, as they would already be used to the layout of the solver.

## Features

### Essential Features

- Sudoku Puzzle
  - This feature allows the user to solve a randomly generated puzzle from a selected difficulty (which determines how many numbers are shown by default).
  - If a puzzle has been previously created but is incomplete, the user will be able to either resume their previous puzzle or generate a new one.
- Sudoku Solver
  - This feature allows the user to find a solution for an existing puzzle, by entering the provided values from the puzzle they wish to solve.
- Difficulty Selection
  - As stated above, this is essential to allow the user to complete a puzzle at a difficulty they are comfortable with.
- About Page
  - This is a requirement, as although Sudoku is well-known, users may be new to completing Sudoku puzzles. For this reason, a summary of the game and its rules need to be accessible to the user.
- Grid Row & Column Highlighting
  - To make the puzzle clearer for users, highlighting the full column and row of the selected cell in the grid would allow them to see the puzzle much easier; as well as give feedback on the currently selected cell.
- Entering Numbers Using Buttons
  - It is important to have the user interact with the app itself instead of their keyboard, as it allows the layout to remain consistent. Furthermore, this also allows for the possibility of numbers being hidden if nine occurrences of the number are already present.
- Exit App Confirmation
  - To avoid the user accidentally leaving the app, an alert should be created to confirm that they wish to leave.

### Desirable Features

- Daily Streak
  - This feature gives a reason for the player to keep revisiting the application, as completing a puzzle every day will increase their streak.
  - If the currently active puzzle is entered into the solver, their streak will not be increased on completion of the puzzle, as this is effectively cheating.
- Personal Greeting
  - This will be shown on the main menu, and provides a slight personal touch to the app. It will use the user's name (which will have been input and saved previously), as well as retrieving the current time of day from the user's device to use the correct prefix.
  - This will require a feature for the user to change their name at any time from the main menu.

- Theme Selection
    - While this is not essential, it would be nice to include if it can be done in the provided time. Selecting the theme of the application would make the user more likely to keep using the app, as they would be able to choose a look that is more appealing.

## Design & Structure

The referenced wireframe figures have been made smaller in this report. They can be seen in their full size within a separate folder alongside this report in my submission.

### App Design & App Activity Lifecycle

When the app is first launched, the 'MainActivity' activity should be launched. This will start by launching a Splash Screen (**Figure 1: Splash Screen** in a separate thread using the 'AsyncTask()' class. After time has passed, this task will finish, and the content view will be changed to the layout for the Main Menu (**Figure 10: Main Menu**). This layout contains the menu in which all other activities can be accessed.

On the first time that this app is launched, a dialog fragment will be created; prompting the user to enter their name (**Figure 8: Enter Name**). If no name is entered here (if the entered string contains only whitespace), then the default name of 'User' will be given. The current hour of the day will then be retrieved to determine the correct prefix for the greeting ('Good Morning/Afternoon/Evening'). This process will be performed within a different thread using the 'AsyncTask()' class, to ensure that the activity is able to properly wait until the full greeting can be returned without crashing due to the main thread being paused.

The user's name can be changed in the same way at any time by pressing the 'Change Name' button in the Main Menu.

If the user presses the 'Back' button on their device, then they will be prompted with an alert dialog to confirm that they wish to leave the app (**Figure 9: Exit Dialog**). Pressing 'Cancel' will close the dialog and pressing 'Confirm' finishes the activity, killing it and closing the app.

Selecting the 'About' button from the Main Menu will create a new intent, which launches the 'AboutActivity' activity. This activity displays details about the rules of Sudoku, an explanation of the app modes, and information about the app's development (**Figure 7: About Section**). To return to the Main Menu, the user can press the 'Back' button on their device to kill the activity.

Selecting the 'New Game' button from the Main Menu will create a new intent, which launches the 'GameActivity activity. This activity contains the main puzzle game and displays the interactive grid for the randomised puzzle. When this activity is run from the context of the 'New Game' button, the user will be asked to select a difficulty from which the puzzle will be generated (**Figure 3: Difficulty Selection**). Like the prompt to enter the user's name, this will be done using a dialog fragment, and will be performed in a separate thread using the 'AsyncTask()' class. After selecting a difficulty, the user must press the 'Confirm' button, which then closes the dialog and generates a puzzle in the main game view (**Figure 4: Puzzle Grid**). If the 'New Game' button is selected when an uncompleted puzzle exists in the apps save data, then a warning to the user is given to confirm if they wish to overwrite this puzzle with a new one (**Figure 2: New Game Warning**).

Selecting the 'Continue' button from the Main Menu will create a new intent, which launches the 'GameActivity' activity. This launches the main game view **(Figure 4: Puzzle Grid**), however instead of prompting the user to select a difficulty, the saved puzzle state is loaded into the game view.

Selecting the 'Solver button from the Main Menu will create a new intent, which launches the 'SolverActivity' activity. This activity contains the solver algorithm for the app and displays

the interactive grid where the user can enter values, and then search for a solution (**Figure 5: Solver Grid**).

As with the About section, the activities for the game view and solver view can be killed by pressing the 'Back' button on the user's device.

### Saving & Loading of Data

The user's entered name will be saved when it is first entered into the dialog prompt. This is then loaded every time the 'MainActivity' activity is created, to properly greet the user.

Within the main puzzle game, the state of the puzzle will be saved every time the user changes a cell in the grid. This is to ensure that if something were to happen to cause to app to stop, then their progress will not have been lost. This puzzle state will be loaded into the game view when the character selects the 'Continue' button.

The user's current completion streak will be saved alongside the current day of the year, after a user has completed a puzzle on a consecutive day. It will be loaded every time the user intends to play a puzzle game, however if there has been more than one day since their last puzzle, then the

## Content

From the current state of the app, content that is still required to be added is full functionality of the puzzle grid interaction. The only interaction the user can have with the grid is to select a cell and have the row and column highlighted to make it easier for the user to see.

The puzzle generation mode and the solver mode also need to be implemented; both of which will be using nearly the same grid interactivity. This means that it can be kept modular, with both modes referencing the same layout, which should reduce development time.

I also need to refine the theme and overall look of the app; including the layout xml files for each of the activities.

For further development past the deadline, it would be nice to implement variations of puzzles that the user can play (such as Killer Sudoku), as well as offer different themes to pick from.

Update cycles in regard to my APKs will consist of major changes to my project; such as refinement of layout, addition of missing or new modes, or bug fixes that may not have been present on an emulated device, however were present on my own device.

## Plan

I first want to finish the grid view, to allow the user to input values into the grid. As this view can be used in both the puzzle and solver modes due to the same grid and inputs being required, this will allow me to then focus on each mode separately without much interruption. This should not take long to implement, so should be finished in the next few days.

I will then focus on implementing the full puzzle generation mode. As the layout is already completed, it should mostly be just the algorithm behind this method so should not take long to finish; and should be completed by the middle of next week (but preferably by the next lab).

At this point, I can then focus on improving the appearance of the app itself, as I will have all the necessary layouts ready; meaning that this would be a good point to refine parts of my project before adding any more. This will be something that will need to be looked at throughout development, however finishing the above tasks will hopefully give me a couple of days to focus on this.

The next part of the app will be algorithm for the solver mode algorithm. As with the puzzle mode, this should not take long (a few days) due to the interactivity being modular.

# Figures

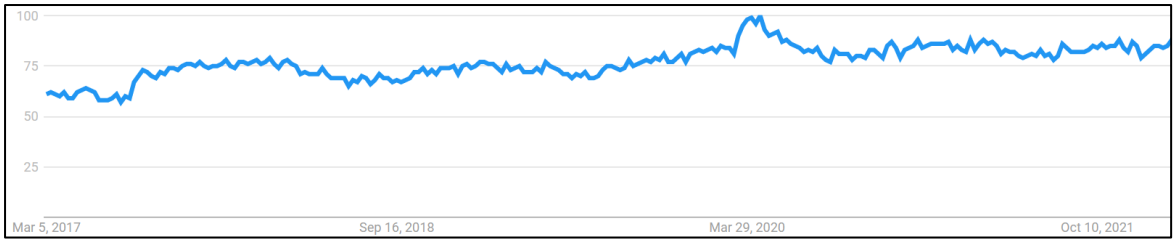


*Figure 6: Relative popularity of 'sudoku' search term over the past five years*

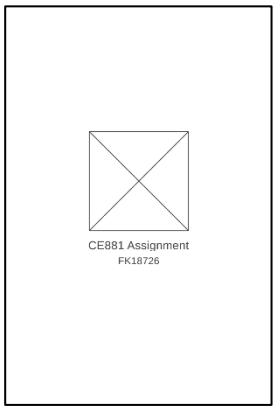


*Figure 1: Splash Screen*


*Figure 10: Main Menu*

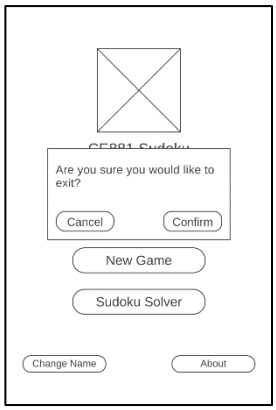


*Figure 9: Exit Dialog*

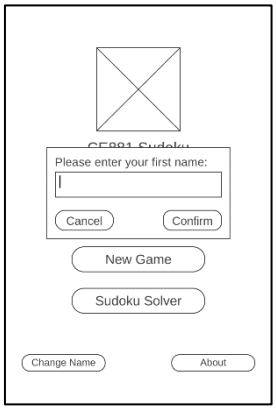


*Figure 8: Enter Name*

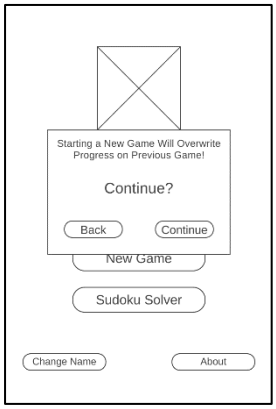


*Figure 2: New Game Warning*

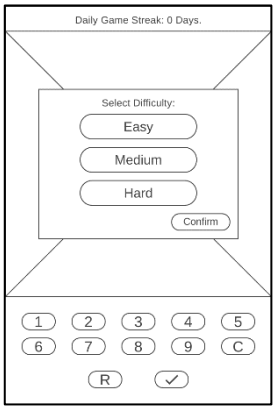


*Figure 3: Difficulty Selection*

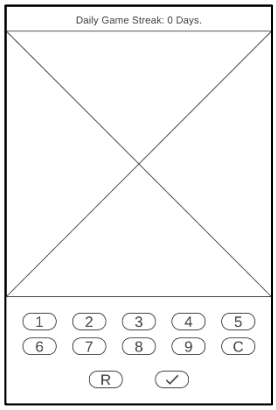


*Figure 4: Puzzle Grid*

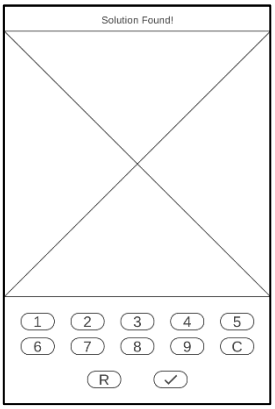


*Figure 5: Solver Grid*

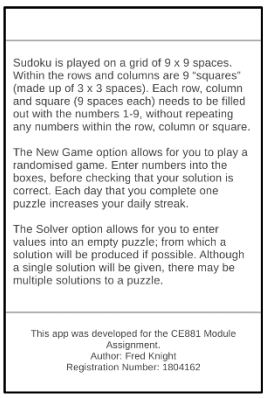


*Figure 7: About Section*

## References

[1] Easybrain, “Sudoku.com - Classic Sudoku,” [Online]. Available: https://play.google.com/store/apps/details?id=com.easybrain.sudoku.android.

[2] Beetles Studios, “Sudoku - Classic Sudoku Puzzle,” [Online]. Available: https://play.google.com/store/apps/details?id=easy.sudoku.puzzle.solver.free.

[3] Guru Puzzle Game, “Sudoku - Classic Sudoku Puzzle,” [Online]. Available: https://play.google.com/store/apps/details?id=sudoku.puzzle.free.game.brain.

[4] Brainium Studios, “Sudoku,” [Online]. Available: https://play.google.com/store/apps/details?id=com.brainium.sudoku.free.

[5] Dustland Design, “Sudoku - The Clean One,” [Online]. Available: https://play.google.com/store/apps/details?id=ee.dustland.android.dustlandsudoku.