Content:

- Description
- Intended User
- Features
- User Interface Mocks
  - Screen 1 - MainActivity
  - Alert Dialogs
  - Screen 1 (Landscape Mode)
  - Screen 2 - Game Activity
  - Complete Application
  - Widget Screen
- Key Considerations
    - How will your app handle data persistence?
    - Describe any corner cases in the UX.
    - Describe any libraries you’ll be using and share your reasoning for including them.
    - Describe how you will implement Google Play Services.
- Next Steps: Required Tasks
  - Task 1: Project Setup
  - Task 2: Implement UI for Each Activity and Fragment
  - Task 3: Functionality of the application
  - Task 4: Error Correctness
  - Task 5: Home Widget
  - Task 5: Other Screens

GitHub Username: rajatgoyal715

Puzzle15

Description

The classic game for those who want to pass the time and to train brains. Popular puzzle game invented by Noah Chapman in 1878. The aim of this game is to move the tiles with the numbers, and arrange them in the ascending order from start to end.

Intended User

This is an application which can be used by people of any age group. This does not have any prerequisite except an urge to train your brain. People who try to think logically will definitely enjoy this classic game.

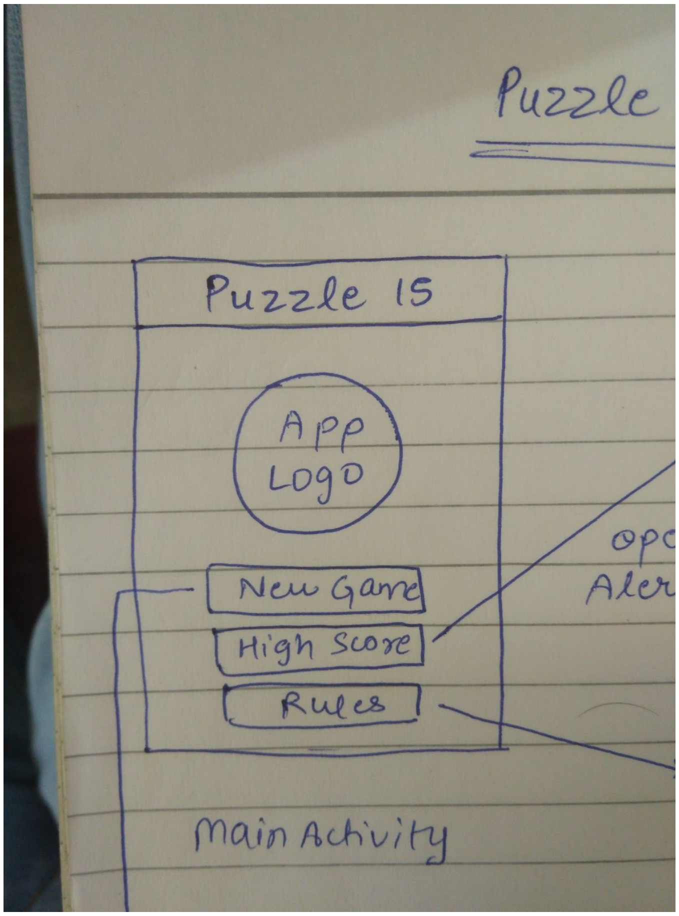
Features

Puzzle15 includes the following features:

- Moves and timer based classical game.
- High Score stored using Content Provider and linked to Firebase also.
- Widget which shows the stats of the game i.e. number of games played, number of games won and all time high scores fetching from Firebase and a play button which opens the application and starts the game on just one click at the widget screen.

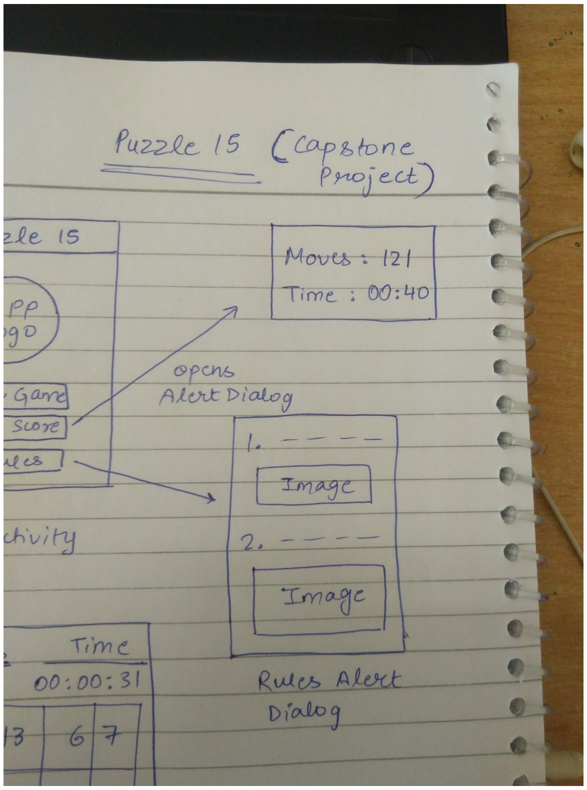
User Interface Mocks

Screen 1 - MainActivity

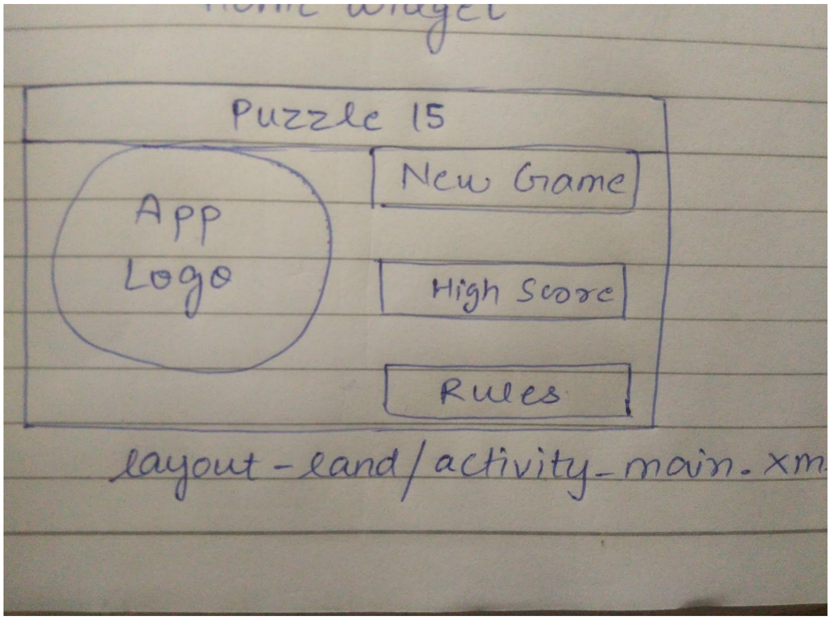


MainActivity of the application will have App Title on the top and app logo below it.
User will be shown three buttons:
New Game button will opens the Game Activity using intent and the timer will start.
High Score button opens up an AlertDialog which shows the all time high scores and other stats.
Rules button will open an AlertDialog which will show user rules of the game with some short description and related images.

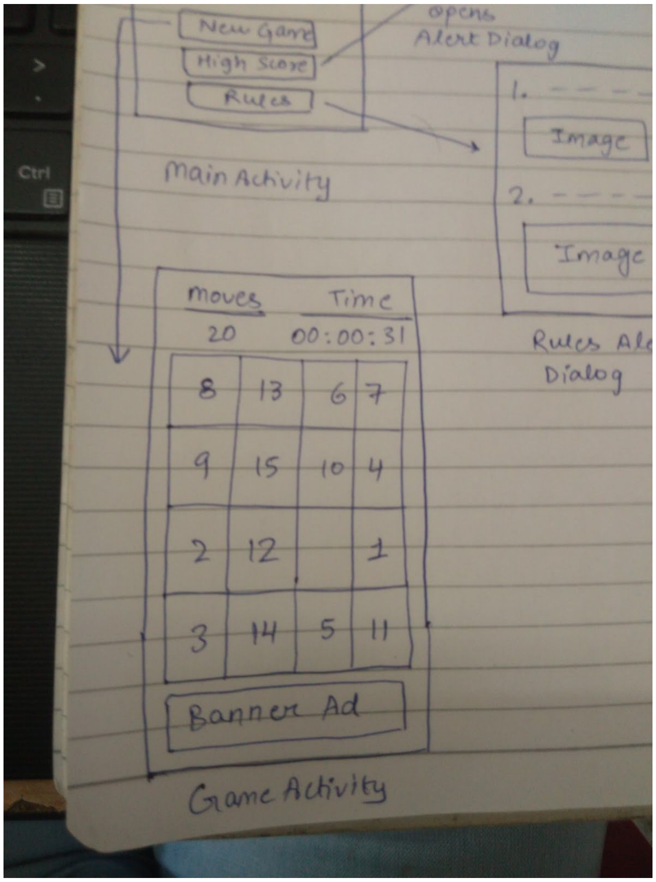
Alert Dialogs



Screen 1 (Landscape Mode)



Screen 2 - Game Activity

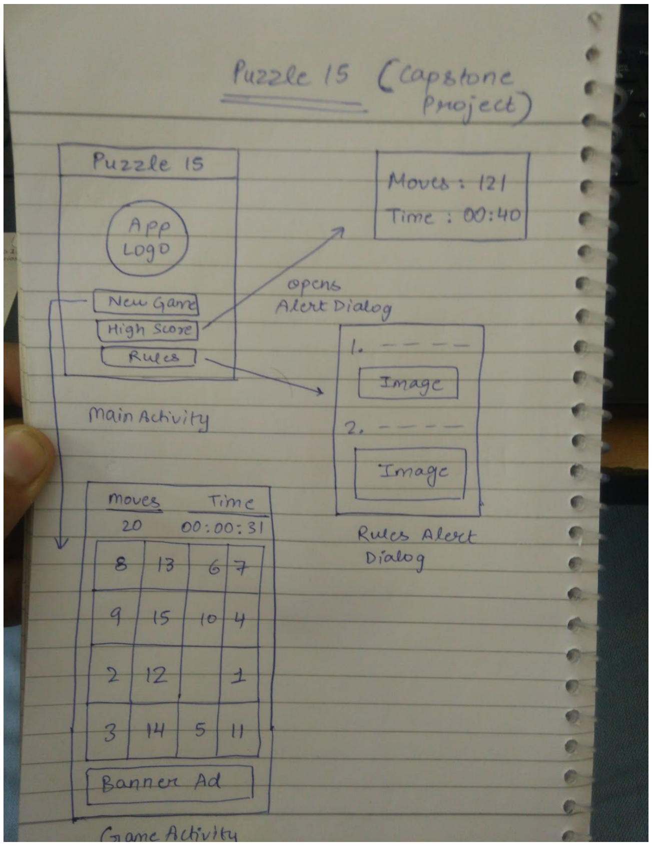


8	13	6	7
9	15	10	4
2	12		1
3	14	5	11

GameActivity screen will show 16 buttons numbered from 1 to 15 which leads to one button being empty. Whenever user clicks on a button, if there is an empty block in any of its four directions, then it will swap the values. This leads to an increase in number of moves which will be shown at the top of the screen with timer.

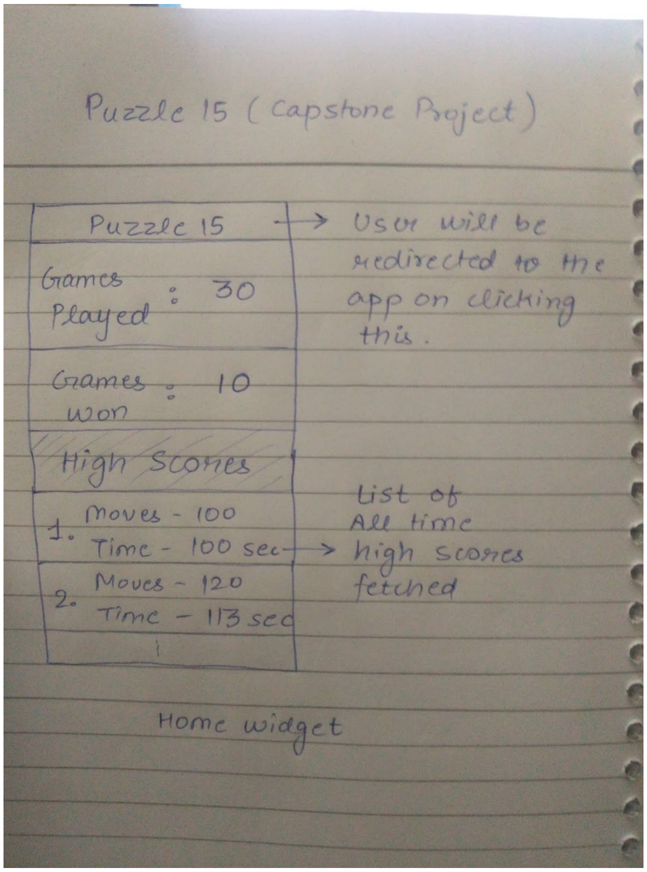
There will be a Banner Ad at the bottom of the screen, which will not tamper the user experience.

Complete Application



This image explains the complete flow of the application.

Widget Screen



Puzzle 15 (Capstone Project)

Puzzle 15 → User will be redirected to the app on clicking this.

Games Played : 30

Games won : 10

High Scores

1. Moves - 100
Time - 100 sec → List of All time high scores fetched

2. Moves - 120
Time - 113 sec

Home widget

The home widget of the app will contain the stats of the game like number of games played in total, number of games won and all time high scores.
If the user clicks on the “Play the Game” button, a new game will be started in the application.

Key Considerations

How will your app handle data persistence?

All the statistics of the game including all high scores will be saved in the Firebase using Realtime Database and fetched and displayed in the app and home widget using Firebase.
If the internet connection is not working, then the stats will be stored in the phone memory using Content Provider and whenever the user comes online, then it will update the Firebase Realtime Database. App uses IntentService to update the statistics of the game in widget, when required.

Describe any edge or corner cases in the UX.

When the activity is paused, by any means, then the table should be stored and used back when the activity resumes. This should work in case of clicking home button and on rotating the device also.

Describe any libraries you'll be using and share your reasoning for including them.

I'll be using ButterKnife for binding views and for click methods.
I will be using android-gif-drawable library for showing GIFs in the application and Firebase Realtime Database to store the statistics related to the game.

Describe how you will implement Google Play Services or other external services.

This application will make use of Google AdMob to show two types of ads. Banner Ad will be shown below the table of numbers and Interstitial Ad will be shown when user wins or loses a game.
The application will also make use of Firebase Crash Reporting to get aware of any crashes the application face.

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and break them down into tangible technical tasks that you can complete one at a time until you have a finished app.

Task 1: Project Setup

- Create a new Android Studio project named “Capstone”.
- Change strings.xml file with app_name as “Puzzle 15”.

Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity.
- Build AlertDialog for Rules button.
- Build UI for Game Activity.

Task 3: Functionality of the application

Build the logic for the game:

- Fill the matrix of numbers with random function taking values from 1 to 15.
- Check number of inversions with the help of this Link.
- If the matrix is not solvable, then make it solvable by switching the positions of last two cells.

Task 4: Error Correctness

Take care of corner cases:

- Store the matrix and score properly.
- If user has won the game, then check if it is high score, by fetching stats from stats using Content Provider.
- If it is high score, then update the stats using Content Provider.

Task 5: Home Widget

Implementation of home widget includes:

- Design the UI.
- Fetch data from Firebase.
- If data stored locally and on Firebase is different, then update Firebase Realtime Database.
- Update UI accordingly.

Task 5: Other Screens

Implementation of home widget includes:

- Design the UI.
- Fetch data from Firebase.
- If data stored locally and on Firebase is different, then update Firebase Realtime Database.
- Update UI accordingly.