Content:

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. It is necessary to move the tiles with the numbers of the field, and arrange them in the ascending order from left to right.

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.

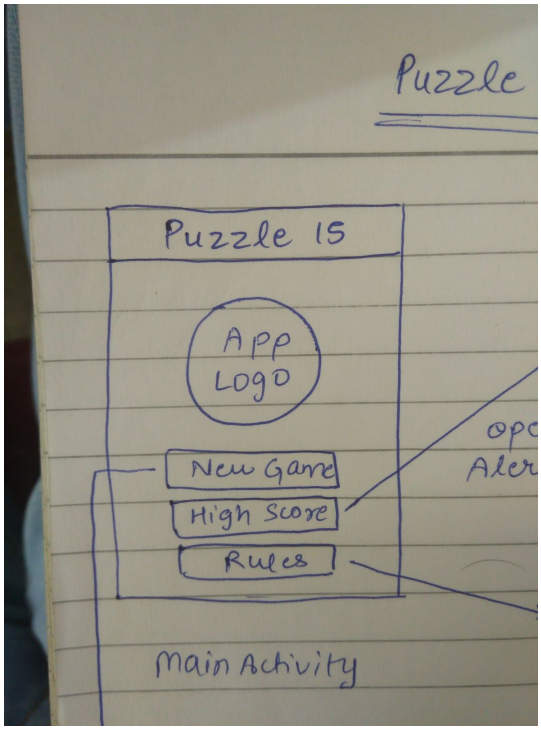
Features

Puzzle15 includes the following features:

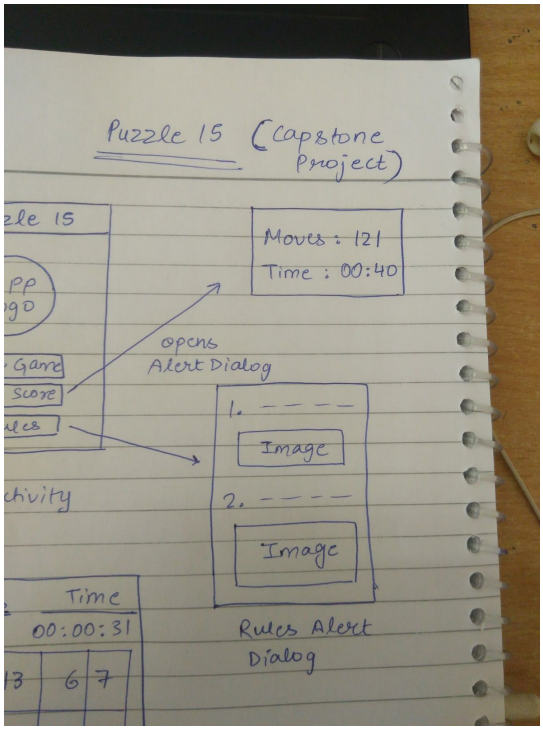
- Count moves in a particular game
- Starts a timer in the start of the game
- Check and store High Score using SharedPreferences

User Interface Mocks

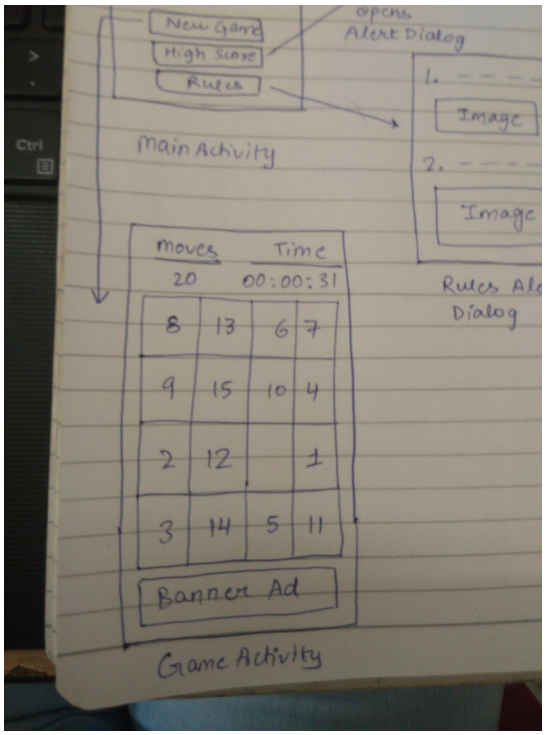
Screen 1



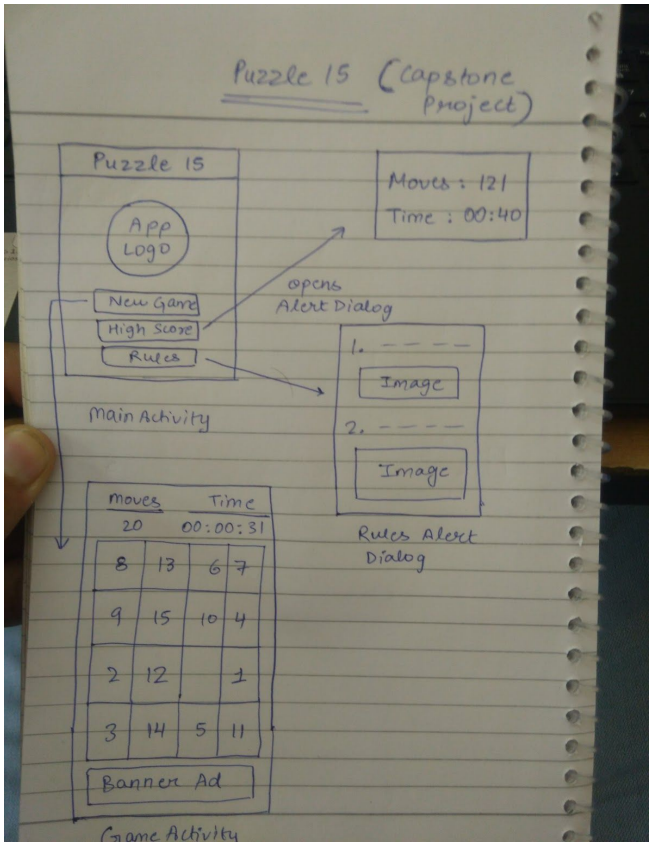
MainActivity of the application will have App Title on the top and app logo below it.
User will be shown three buttons:
New Game button will opens the GameActivity using intent and the timer will start.
High Score button opens up an AlertDialog which shows the best game stats which took the least number of moves and least time to solve the game.
Rules button will open an AlertDialog which will show user rules of the game with some short description and related images.


Puzzle 15 (Capstone Project)
Moves : 121
Time : 00:40
opens Alert Dialog
Image
Image
Time
00:00:31
Rules Alert Dialog

Screen 2


New Game
High Score
Rules
opens Alert Dialog
Image
Image
Main Activity
Moves
20
Time
00:00:31
8 13 6 7
9 15 10 4
2 12 1
3 14 5 11
Banner Ad
Game Activity

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.



This image explains the complete flow of the application.

Key Considerations

How will your app handle data persistence?

High Score will be saved in the phone memory using Shared Preferences and retrieved back from shared preferences.

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.

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.

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 GameActivity.

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 shared preferences.
- If it is high score, then update the shared preferences.

Task 5: Others

On clicking High Score button, display high score by fetching from shared preferences.
Ask user once if he/she wants to close the app using an AlertDialog.