Deltapath Android Programming Test

Description

The goal of this programming test is to develop a card matching game. The flow of the game should be as follows

1. In the home screen, the user can choose one game from a list to start. See "Game Types".
2. After the user chose the game, it should display a game board, having a 4*4 grid containing all cards which are face down. Score starts with 0. Cards are randomly generated based on the game that the user chooses.
3. Tap to choose a card. This action will flip the card to face up, and score - 1.
4. Tap on a chosen card to unchoose the card. This action will flip the card to face down, and no score changes.
5. When a card is chosen, check whether the number of chosen cards meets the requirement of checking "matching". Based on the game types, the rules of the "matching" is different.
6. If it doesn't need to check "matching", do nothing.
7. Otherwise, check whether the chosen cards are "matching" or not. If they match, those cards will not be able to choose again, keep face up and greyed out. If they mismatch, unchoose all the chosen cards except for the newly chosen one. (Two cards, unchoose the first card. Three cards, unchoose the first two cards)
8. There should have a text label indicating the current score.
9. There should have a button to start over the game.
10. There should have a button to end the game. The game is also considered finished when all cards are matched.
11. **(Bonus)** When the game ended, ask the user to input his name. The user's name and the score would then be stored in the database. The user should also be notified of his score and current rankings. There should have a button in the game board page to show the high scores. The high scores table should contain three columns: Rank, Name, Score.

Game Types

1. Playing cards. (Required)

DEFINITION:

The deck consists of 52 cards, including 13 ranks of each of the four suits, clubs (♣), diamonds (♦), hearts (♥) and spades (♠).

MATCHING RULES:

- If the rank of two cards are the same, they are matched. Score + 16.
- If the suit of two cards are the same, they are matched. Score + 4.
- Otherwise they are mismatched. Score - 2.

RESOURCES:

- Card back: Provided.
- Card front: String consists of suit and rank on provided card front background. Suit uses unicode "♠", "♥", "♦", "♣". See the Wikipedia for more information. Rank uses A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K. For example:

2. Set. (Bonus)

DEFINITION:

The deck consists of 81 cards varying in four features: number (one, two, or three); symbol (diamond, squiggle, oval); shading (solid, striped, or open); and color (red, green, or purple). More on Wikipedia.

MATCHING RULES:

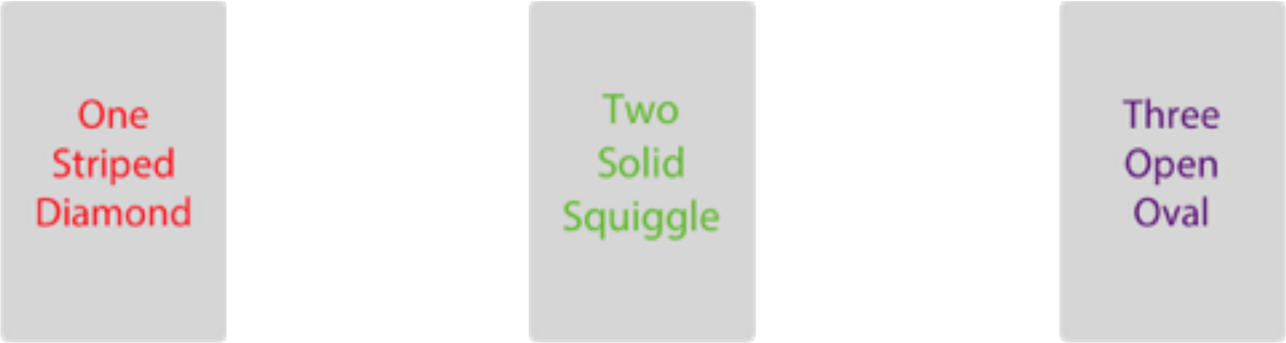
- If three cards form a Set, they are matched. Score + 16.

> *A set consists of three cards which satisfy all of these conditions:*
> *They all have the same number, or they have three different numbers.*
> *They all have the same symbol, or they have three different symbols.*
> *They all have the same shading, or they have three different shadings.*
> *They all have the same color, or they have three different colors.*

- Otherwise mismatched. Score - 2.

RESOURCES:

- Card back: Provided.
- Card front: Strings consists of the number, the symbol and the shading in literal using the color of the card on provided card front background. For example:



Requirement

1. Android app supporting Android version 4.0 and above.
2. The app should be easily extended. For example, adding a new game type with different matching rules should be easy.
3. It is up to you to decide the layout. Make sure it looks good across all size of devices and different orientation.
4. Write comments in wherever you think is appropriate.
5. Finish the test as soon as possible.

Evaluation

1. The time to finish this test
2. User Experience
3. Code readability and coding style
4. Class design, data structure, logic and algorithm
5. App performance and stability

Delivery

You should send us the archive file that containing the following.

- All the source code files and necessary files to build and launch the app.
- The .apk file.
- Readme or project build procedure
- User guide as a bonus

Note: Make sure your email service provider allows you to send the archive file containing the .apk file. Some email service may check the archive file and block the email from sending. If so, you can set a password to the archive file and provide the password in the email.