Biplab Thapa Magar
OPL – Fall 2020

Manual for Pinochle for Java/Android

Bug Report

No known bugs in program

Feature Report:

Missing features:

All features have been implemented

Extra features:

No extra features were implemented

Description of Data Structures/Classes:

Classes:

1. **Card**: Represents a card in a game of Pinochle. Each card is comprised of a rank (of enum type Rank), suit (of enum type suit), and id.
2. **Deck**: Represents the stack of 48 cards to be used for the game of Pinochle. It also acts as the stock pile once cards have been taken out of the deck and distributed. Consists of a vector of 48 card objects
3. **Serialization**: This class handles the conversion of string data from file into object data concerning the cards that each player possesses in their various card piles. It also handles converting the same object data into serialized data to store in a file. In doing this, it interprets and translates how game state is stored in the save file and in the program during gameplay. It makes use of GroupOfCards and MeldsStorage objects.
4. **Player:** Represents a player in a game of Pinochle. It also generates the logic for what card to play and what melds to play. This logic is used differently by its child classes. It has two child class: Human and Computer. Its composition includes MeldServices object and GroupOfCards object.
5. **Computer:** Represents the computer player in a game of Pinochle. It inherits from the Player class. It uses logic from Player class to play cards and melds during a game of Pinochle.
6. **Human:** Represents a human player (user) in a game of Pinochle. It handles prompting user for input on what cards to throw and what cards to play for meld. It inherits from the Player class. It also provided suggestions, using logic from the player class, to the user on what card/melds to play

7. **GroupOfCards:** Represents a group of cards, in any order, in a game of Pinochle. It handles retrieval and searching of cards in terms of various parameters. It has one child class: MeldInstance.
8. **MeldInstance:** Represents a group of cards that combine to create a meld. It is a child class of GroupOfCards. A meld instance can contain cards that do no combine to create a meld as well. However, the MeldInstance object marks itself as an invalid meld if that happens. This class self-contains the logic to validate whether a group of cards combine to create a meld or not. MeldInstance makes use of Card class objects.
9. **MeldsStorage:** Represents a group of MeldInstance objects. MeldsStorage stores MeldInstance objects by separating them based on meld types. A MeldsStorage object can be used to store the melds a player has played during a round.
10. **MeldServices:** This class encapsulates various services pertaining to melds. It is used especially by the Player class and its subclasses to help figure out what melds can be played and should be played. It also keeps track of the melds a player has played during a round. It uses a MeldsStorage object to store players' melds.
11. **StringUtilities:** This class encapsulates the various functionalities for cleaning, interpreting, parsing, and converting strings in the Pinochle program. It is a utility class whose functions are all static functions.
12. **GameModel:** The GameModel class represents the state of a game of Pinochle. It handles the flow of the game as its methods are used to update the game state according to the rules and workings of the Pinochle game. It is primarily used by the view and controller (in accordance with the MVC) pattern to display the game to the end user and to update the model according to user input.
13. **WelcomeActivity:** An Android Activity class that represents the Welcome Screen of the game of Pinochle. It gives the end user the options to either start a new game or to load a new game.
14. **GameActivity:** An Android Activity class that represents the main gameplay of a game of Pinochle. It accesses the a GameModel object and displays the game to the user according to the object's state. It also updates the model according to the user input. In this way, it fulfills the View and Controller function in the MVC pattern
15. **CardViewAdapter:** A class that is used to generate side-scrollable views. It is an extension of the RecyclerView.Adapter class and acts as an "adapter" to the RecyclerView widget by taking a list of objects in the game and creating lists of items to display inside the side-scroll view (RecyclerView). This particular implementation of the adapter is used to display a list of cards in the side-scroll container.

Enums:

Rank : Ace, Ten, King, Queen, Jack, Nine

Suit: Clubs, Hearts, Diamonds, Spades

Meld: Flush, RoyalMarriage, Marriage, Dix, FourAces, FourKings, FourQueens, FourJacks, and Pinochle

How to Run

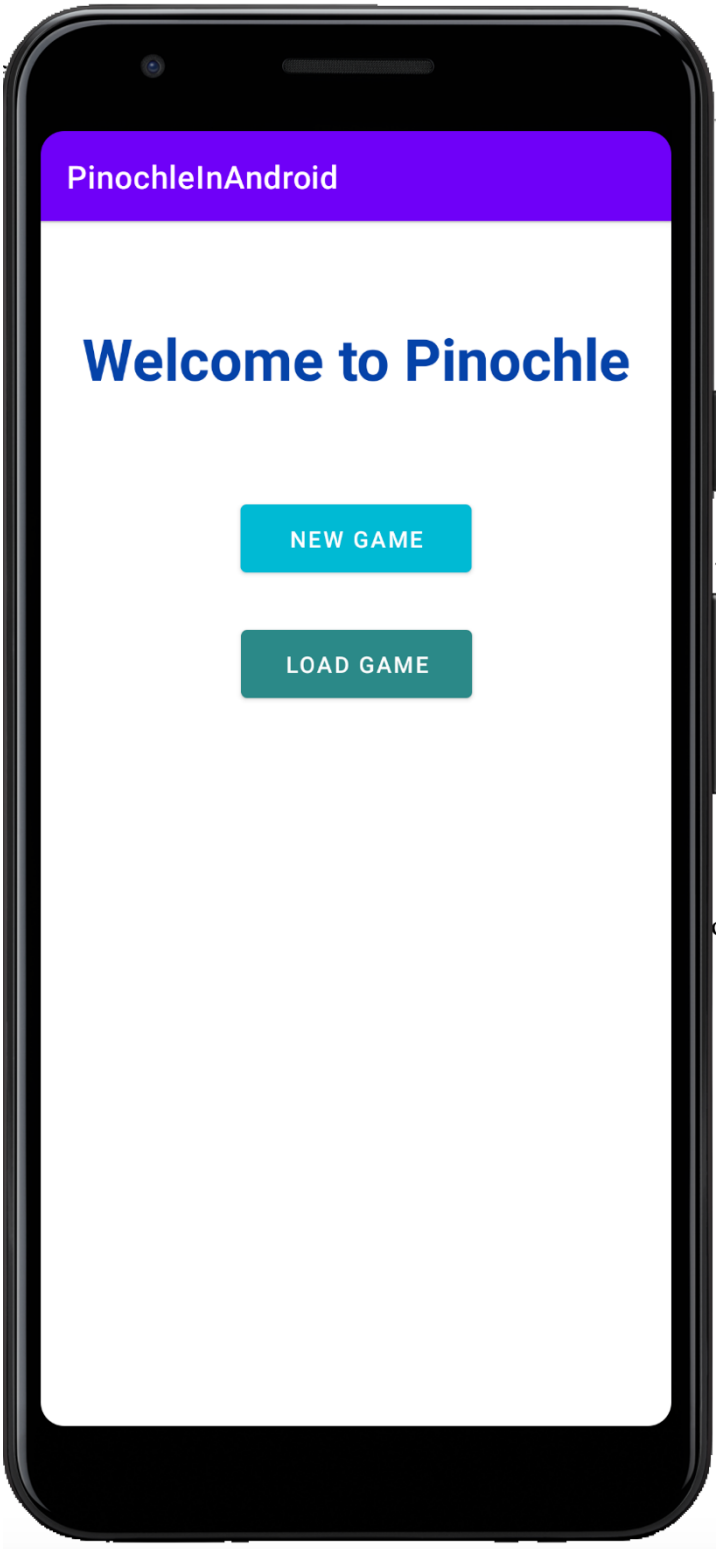
1) In the src folder, locate the pinochle.apk file
2) Copy the file to Android device or Android emulator
3) Open the file and accept permissions so that the file is installed
4) Open the installed PinochleInAndroid application

OR

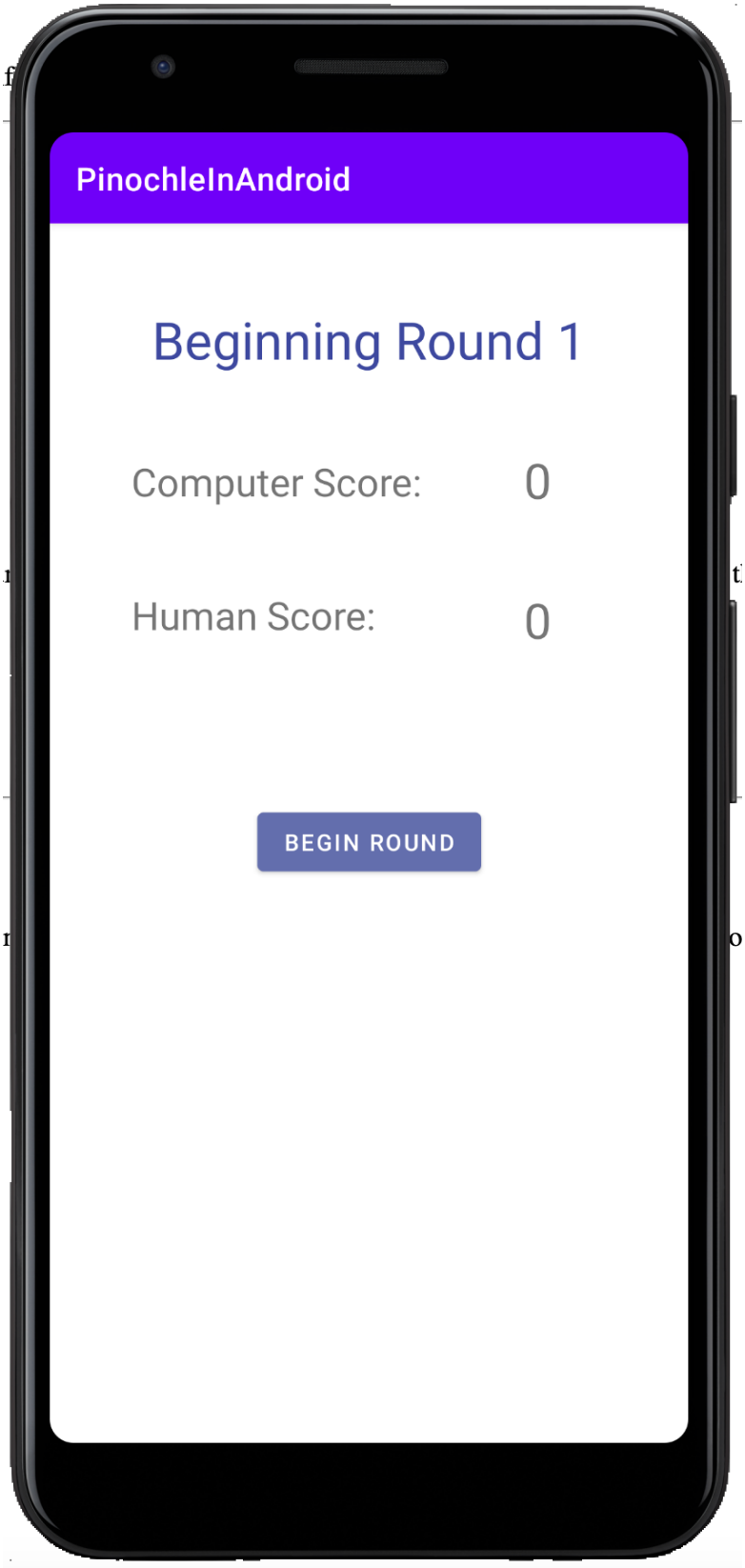
1) **Open the PinochleInAndroid project folder using Android Studio, and run and build the project folder and use an emulator to run the application.**

Screenshots

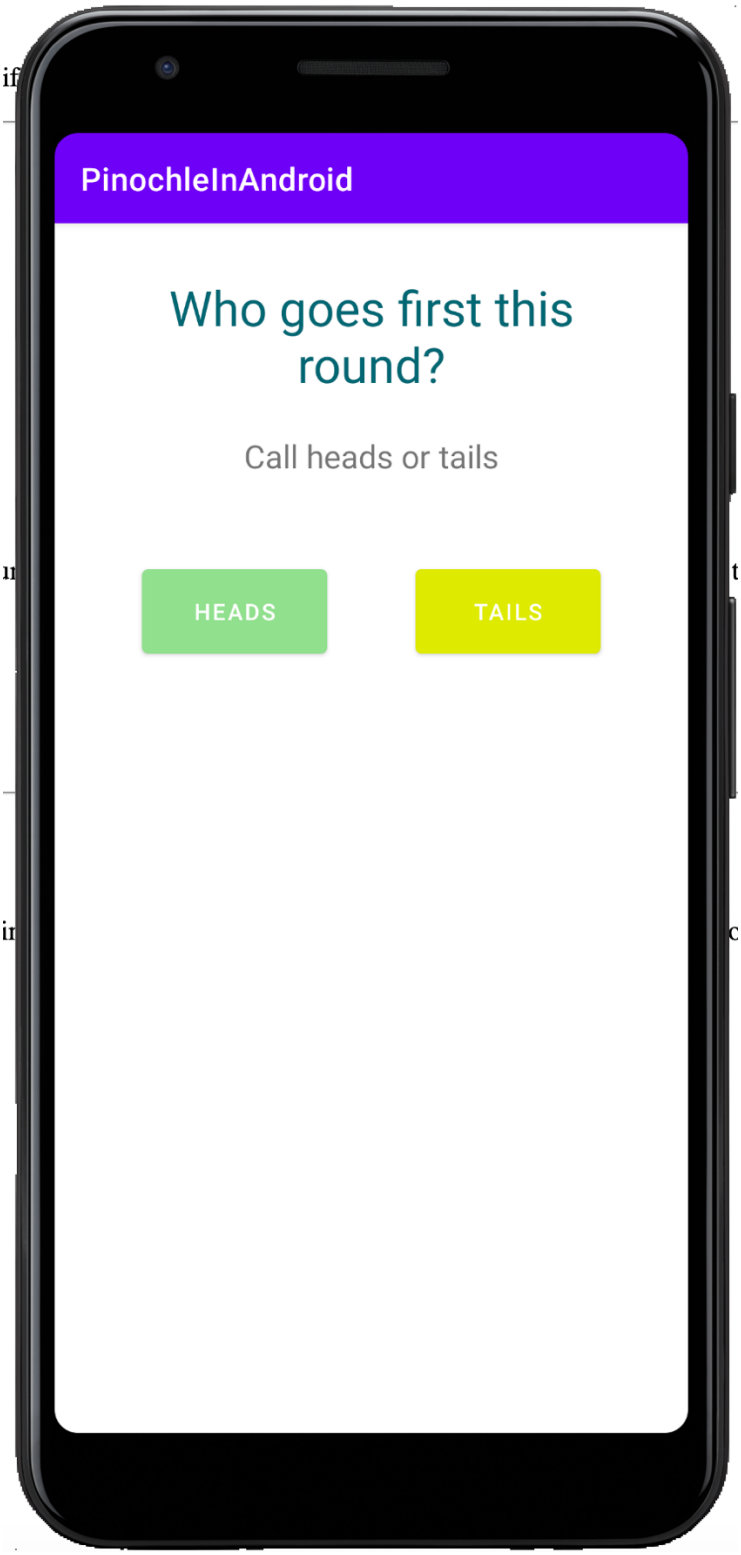
1. Welcome Screen



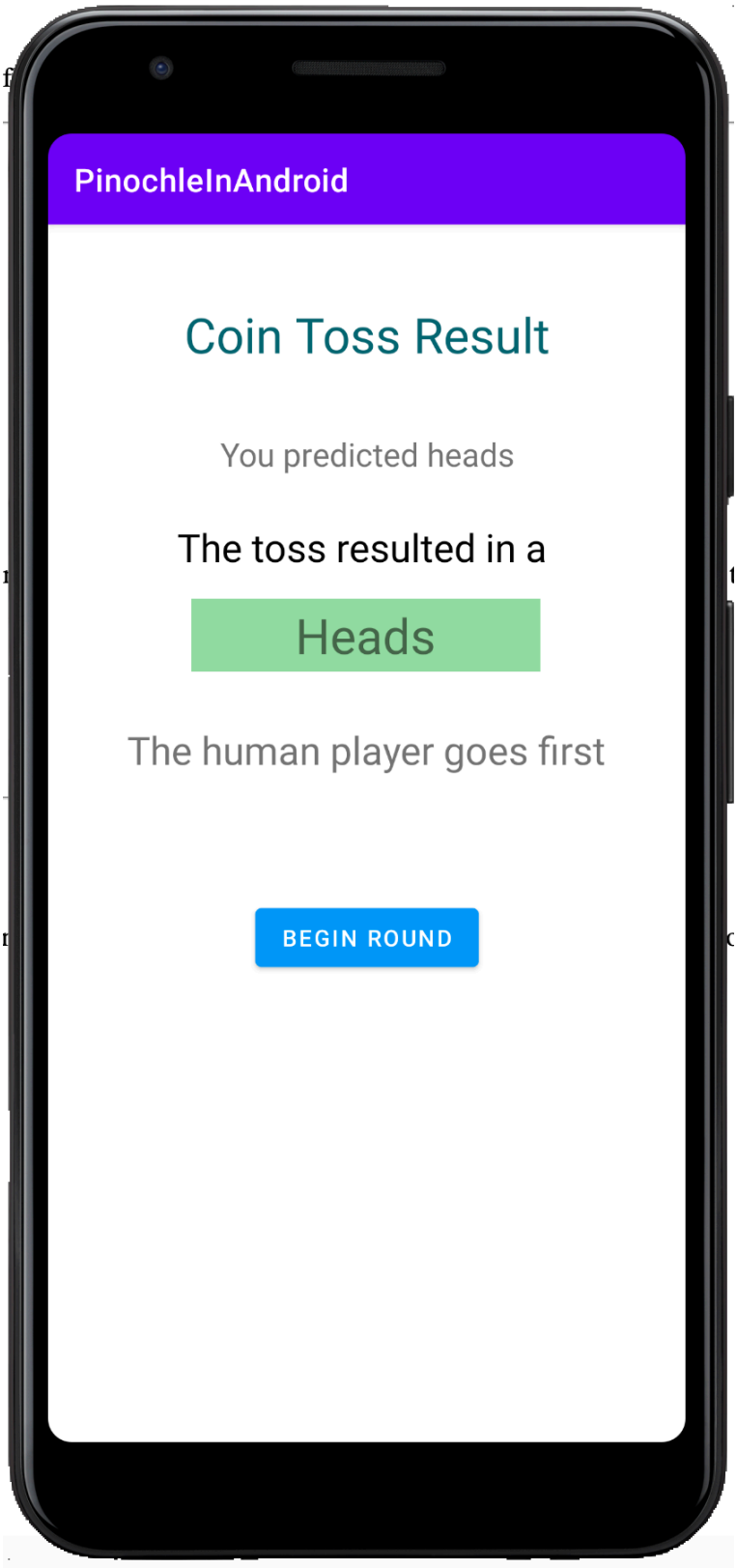
2. Starting a new game



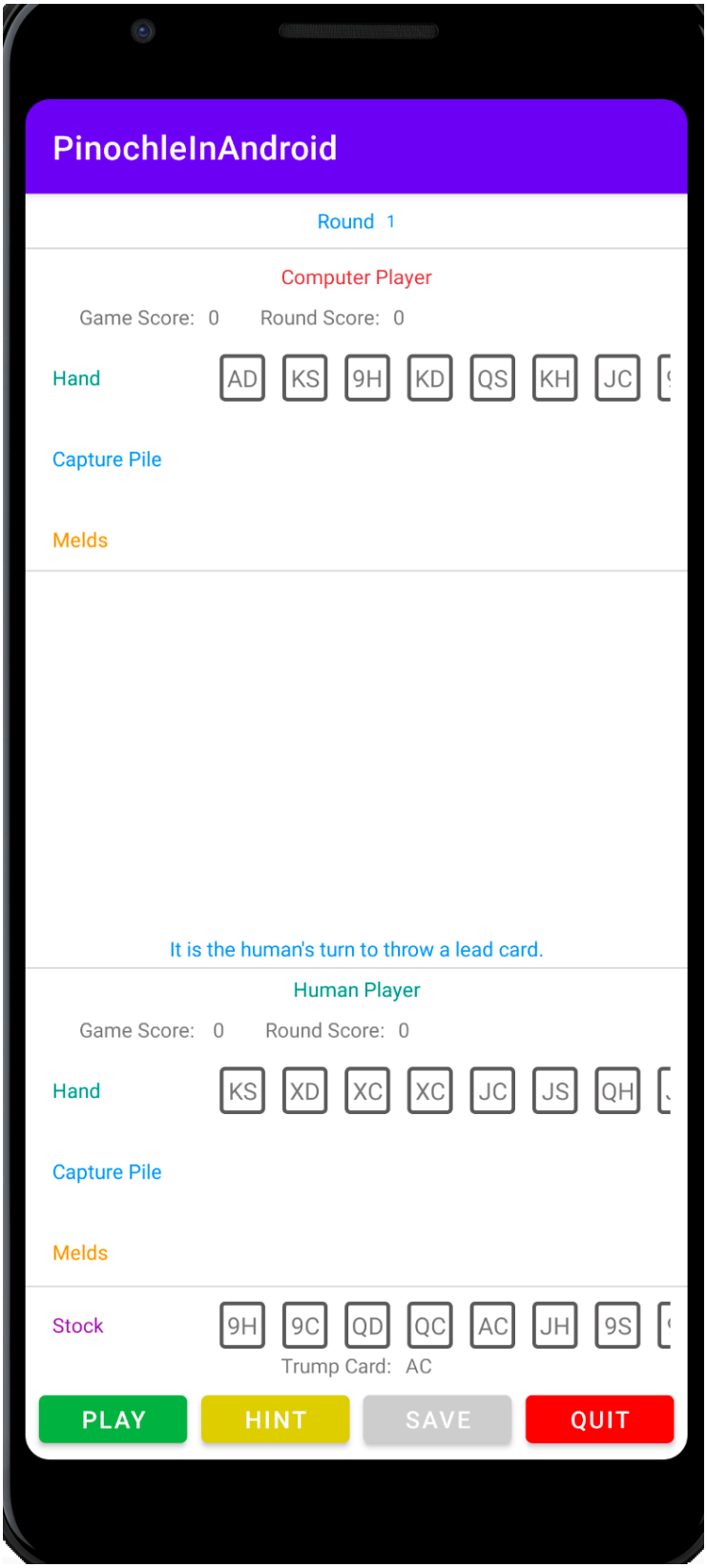
3. Coin toss


PinochleInAndroid
Who goes first this round?
Call heads or tails
HEADS
TAILS

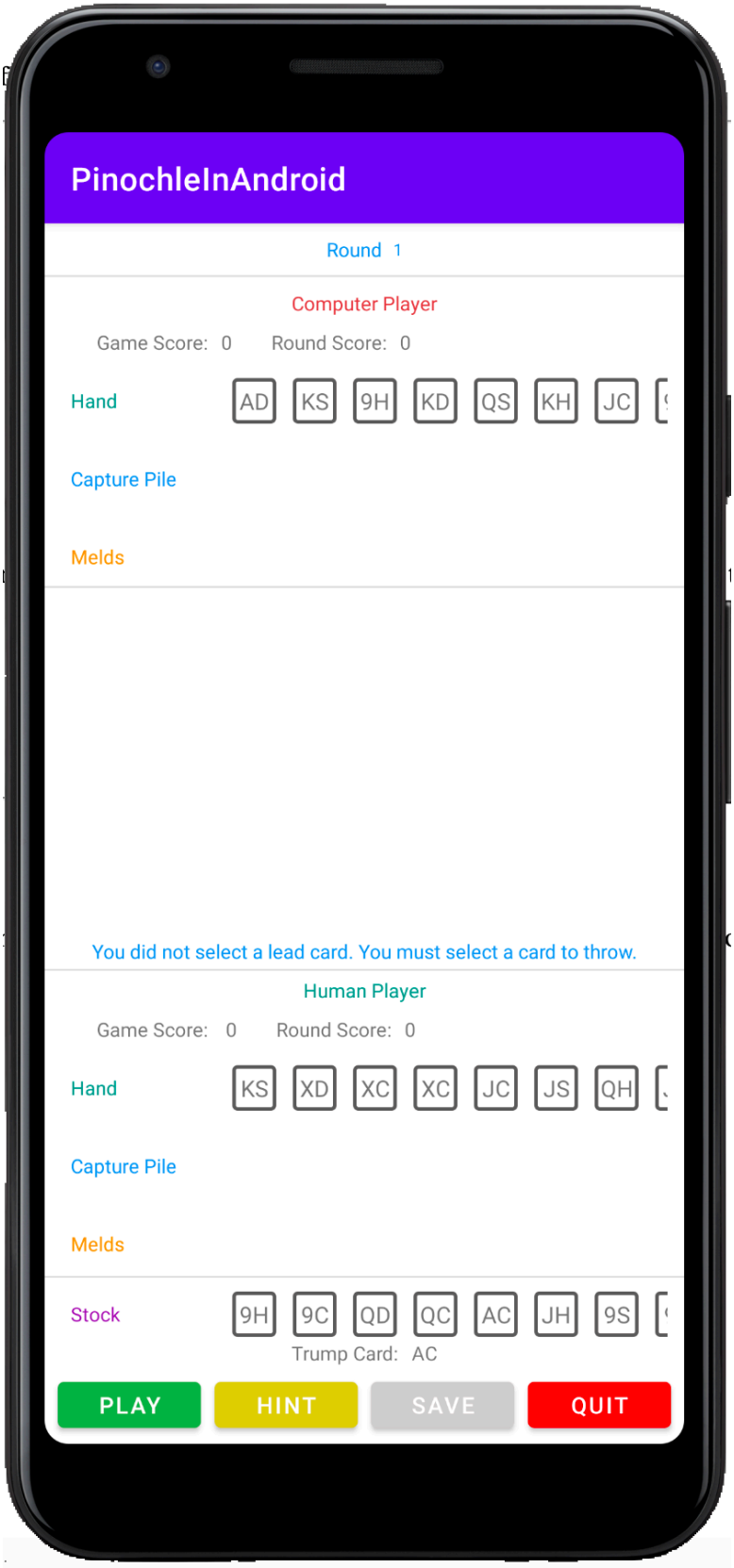
4. Showing coin toss result



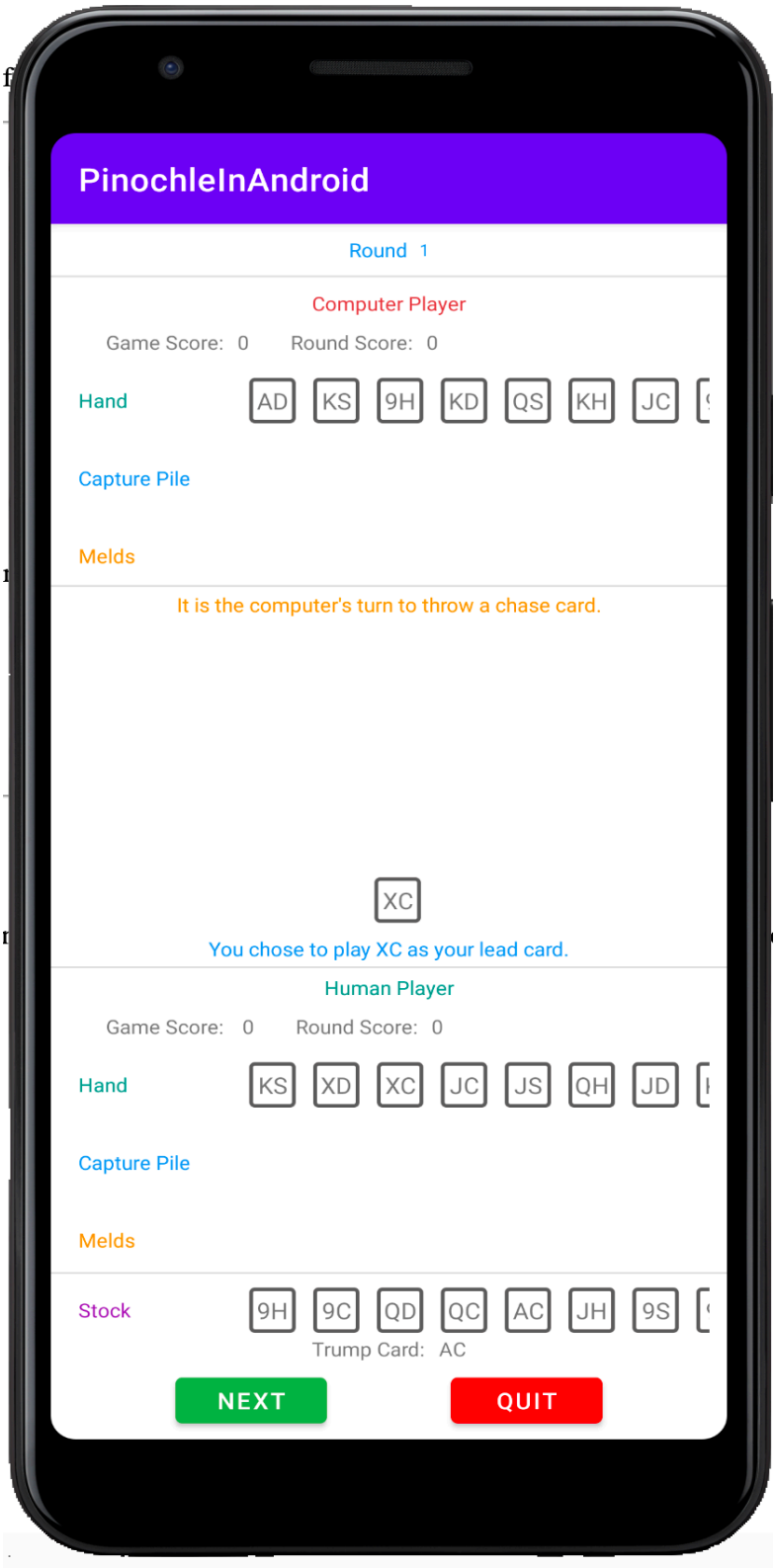
5. Beginning a new round



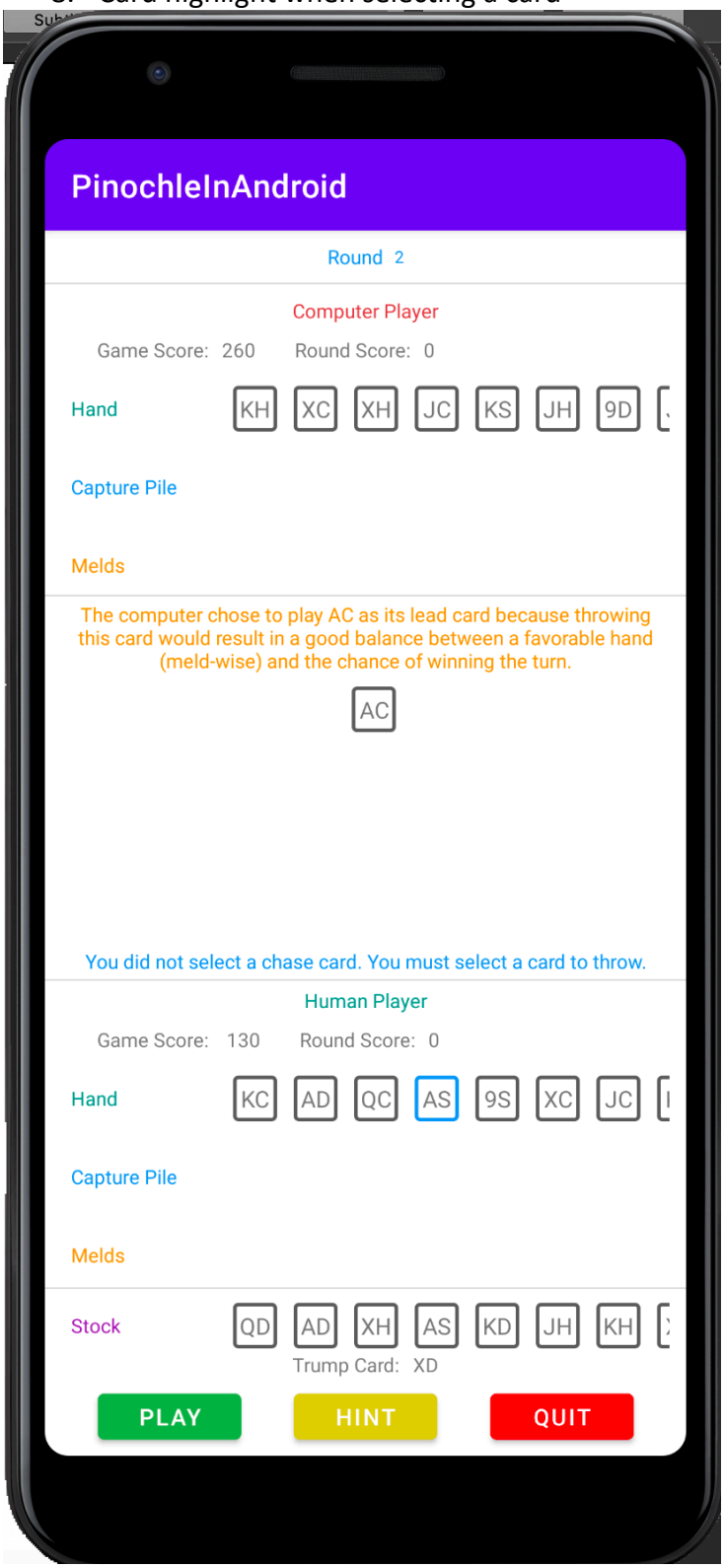
6. When user clicks on “Play” without selecting a card



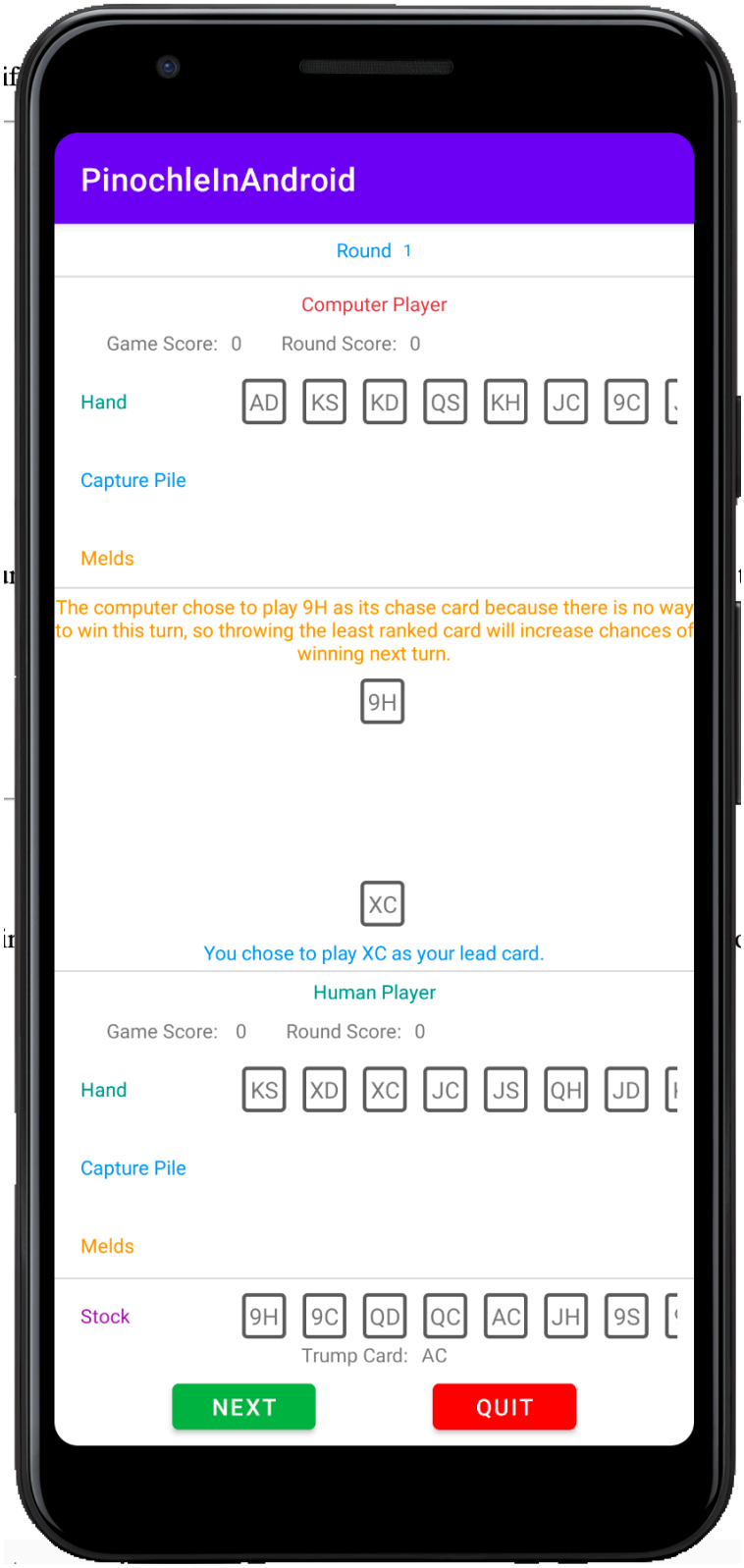
7. Human making a move



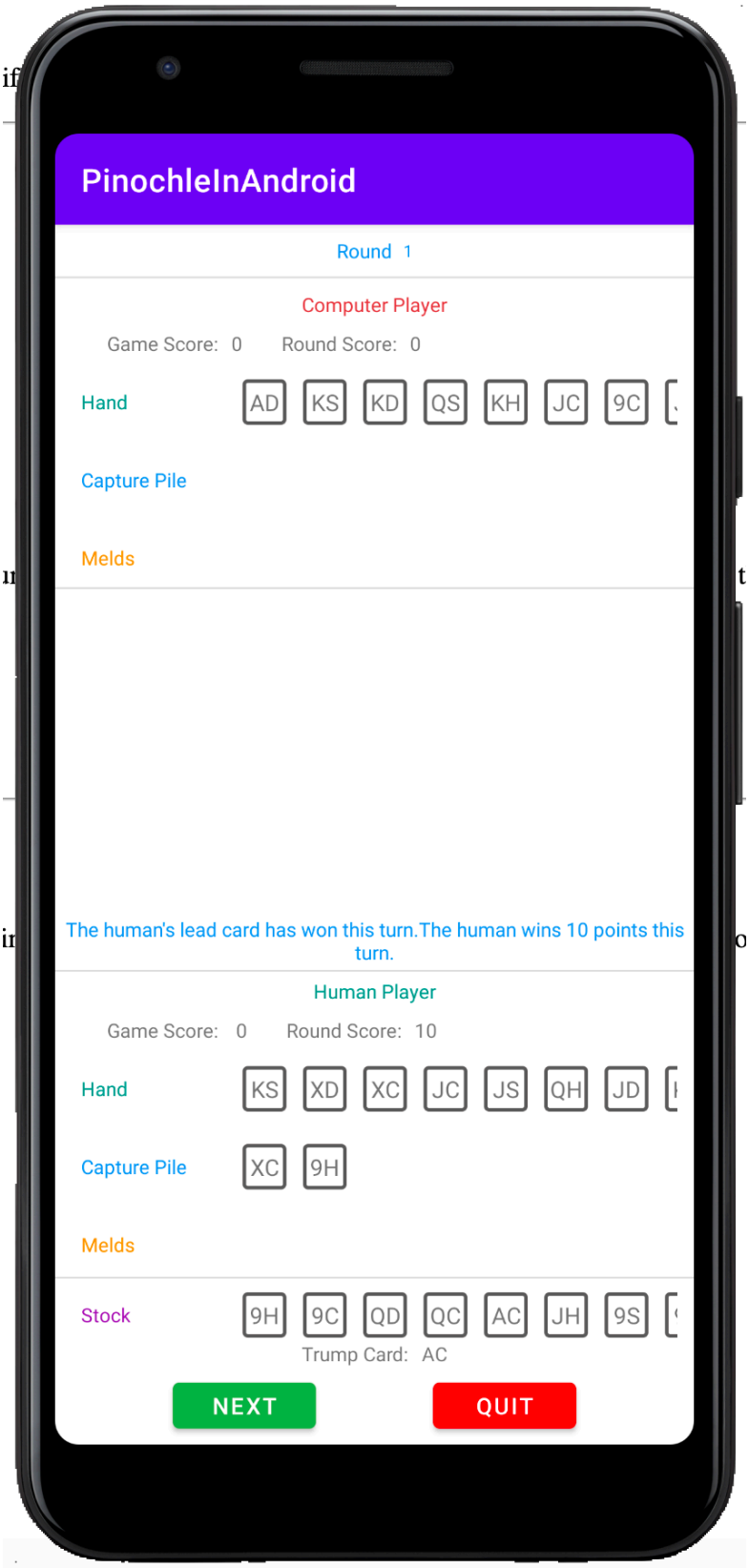
8. Card highlight when selecting a card



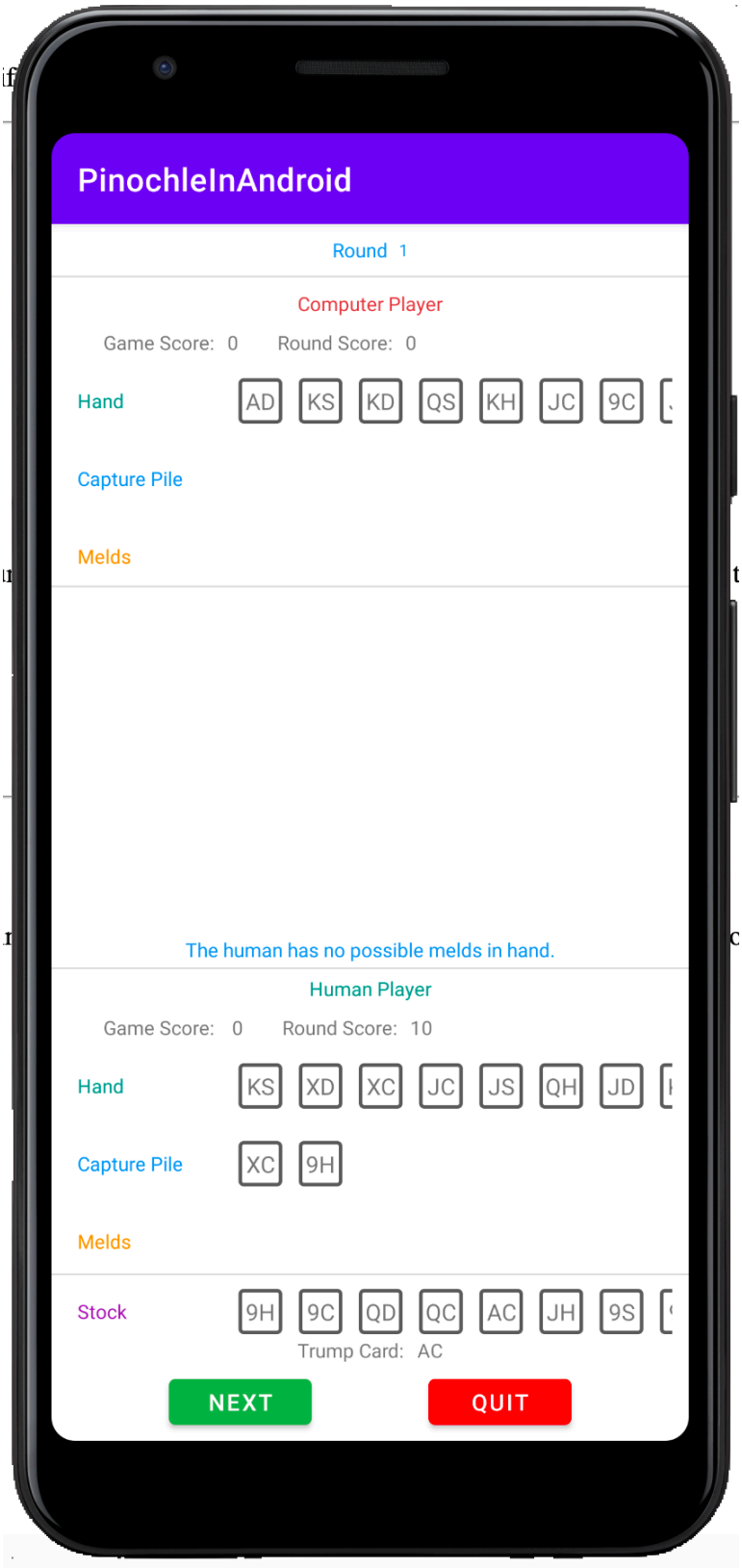
9. Computer making a move



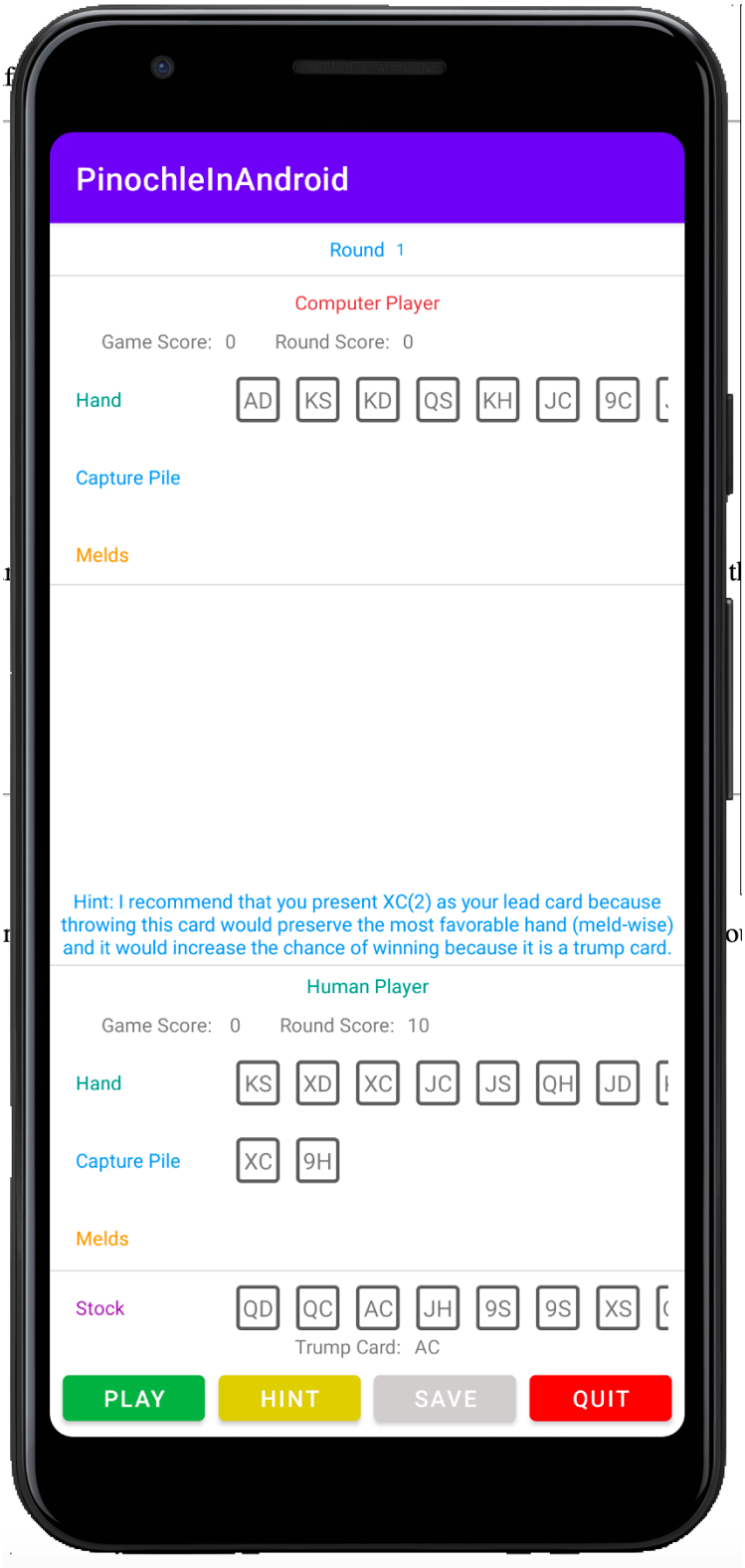
10. Human winning a turn



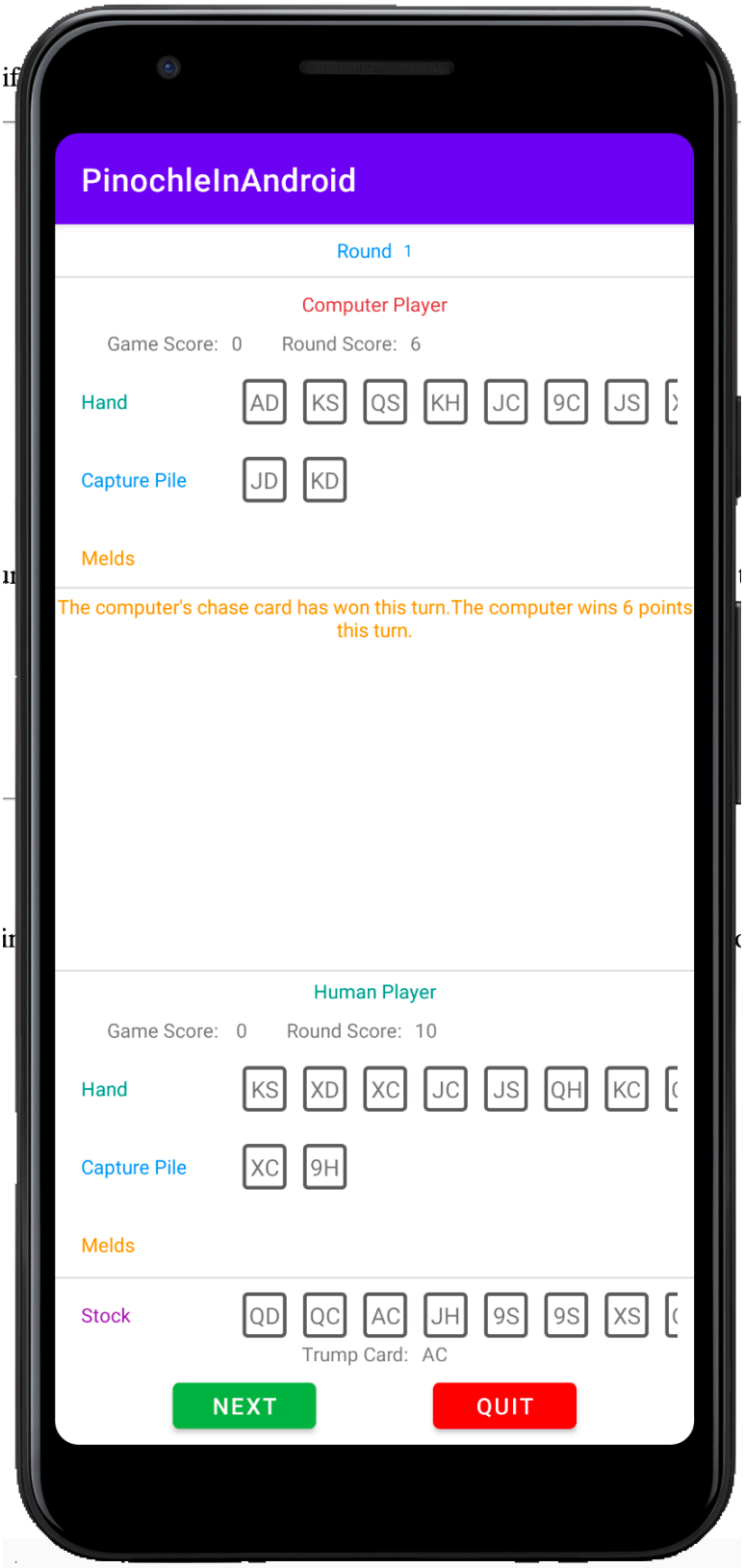
11. Human with no melds to play



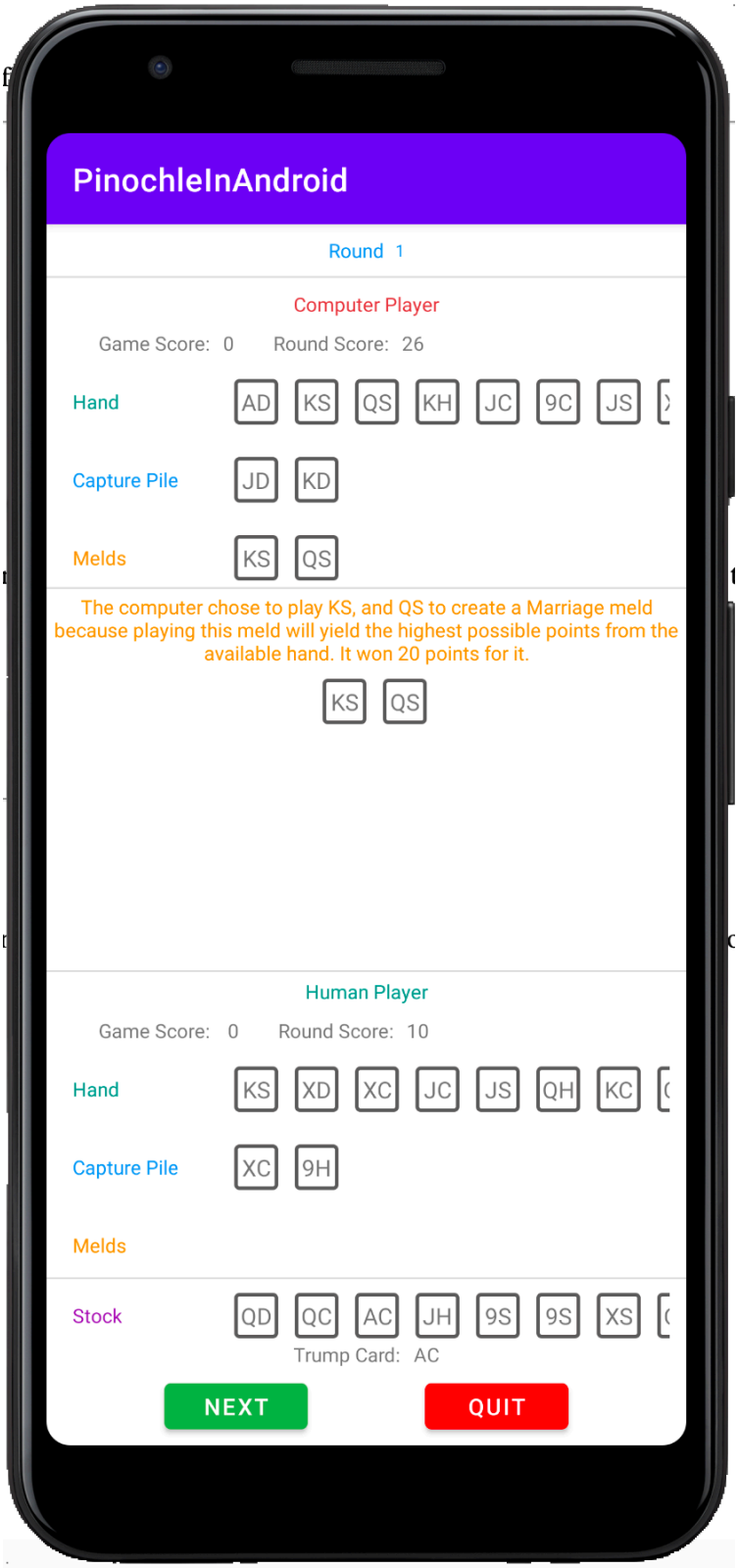
12. Human asking for hint



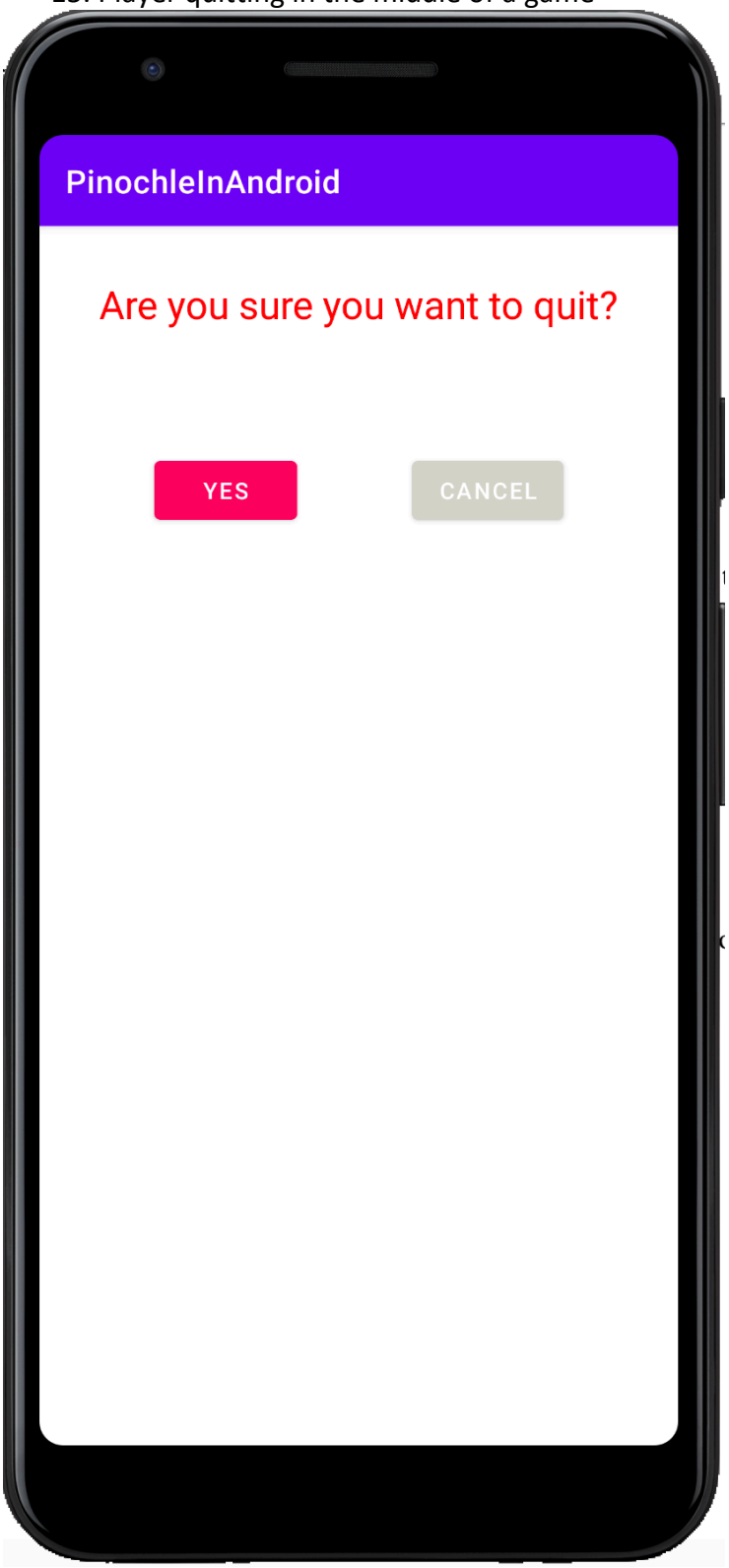
13. Computer winning turn



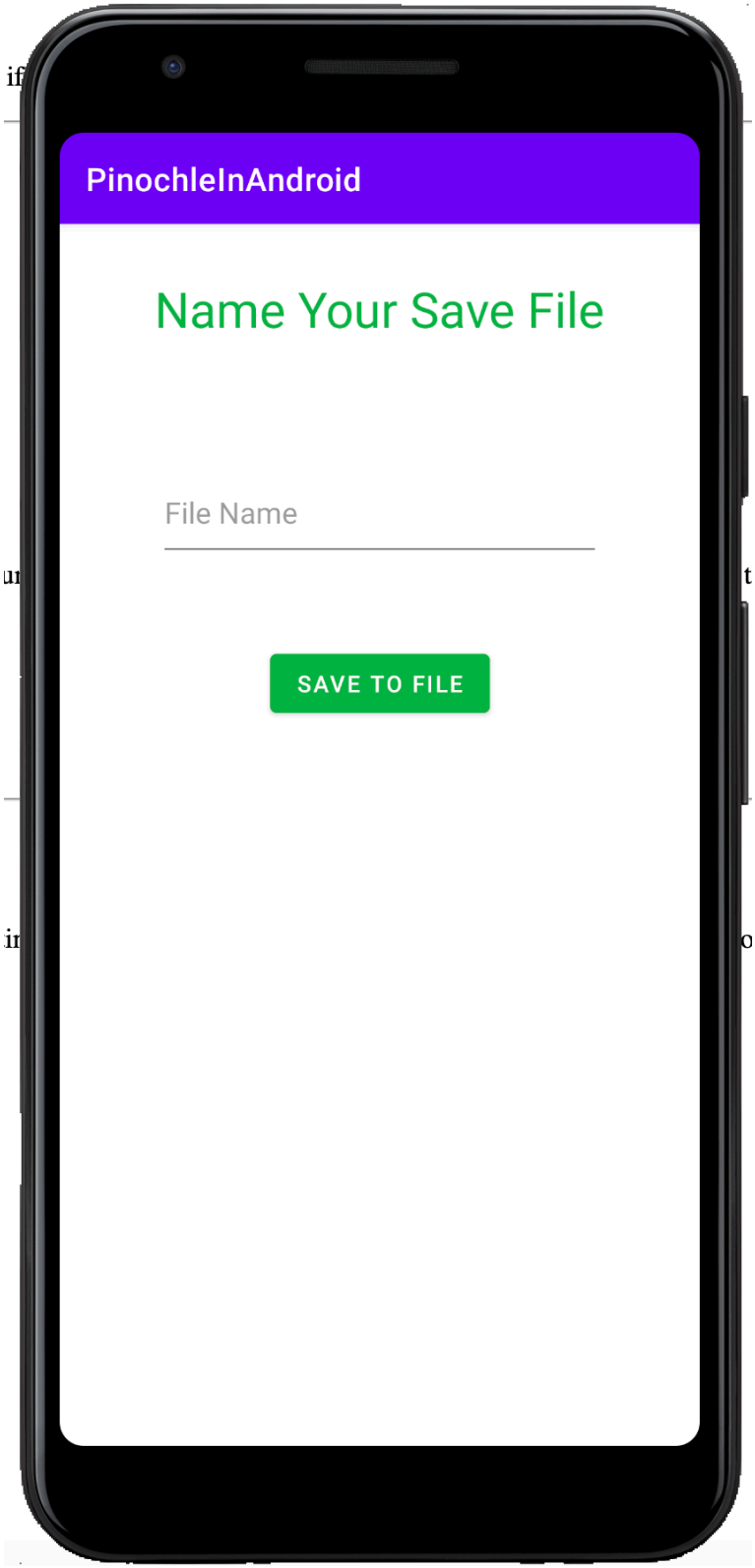
14. Computer playing a meld



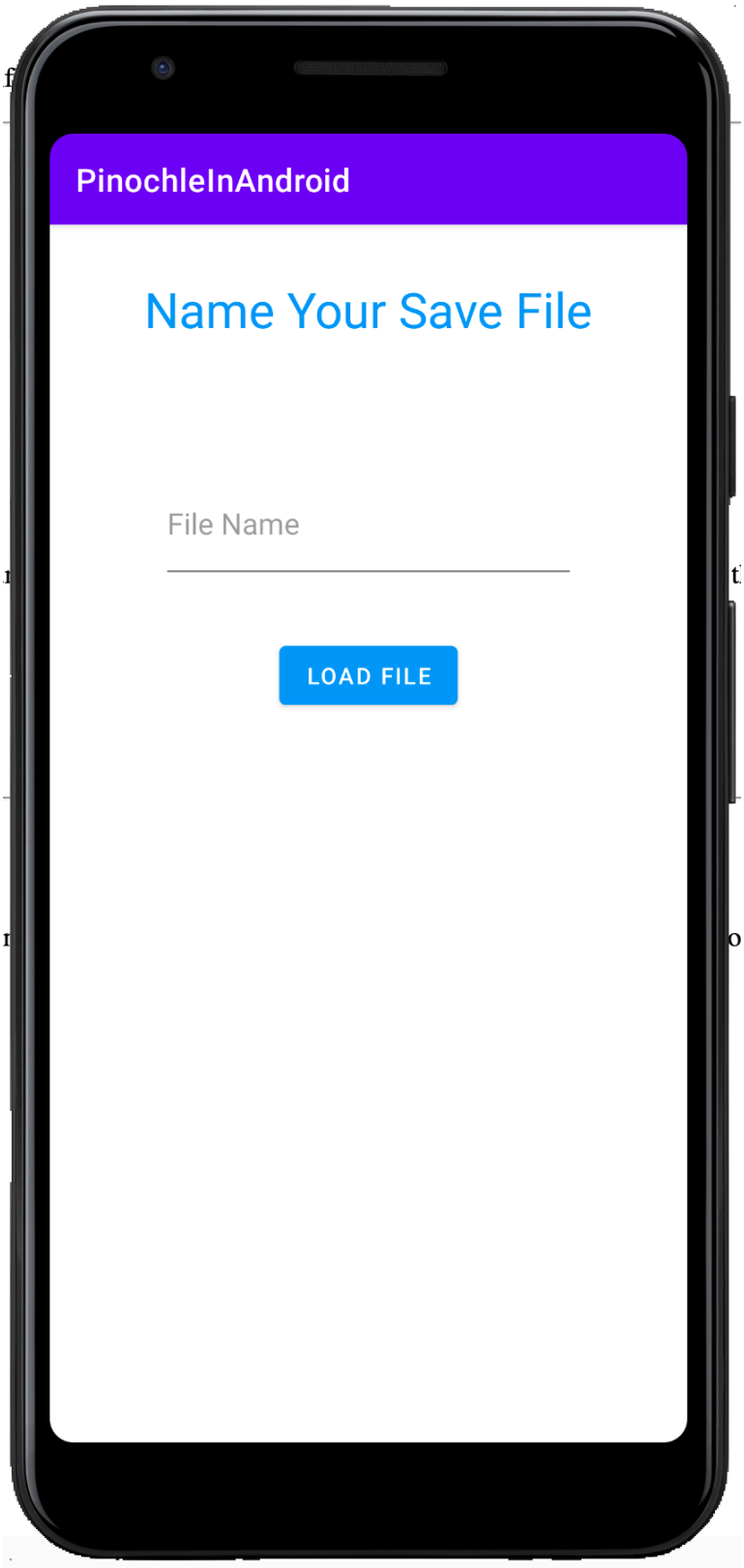
15. Player quitting in the middle of a game



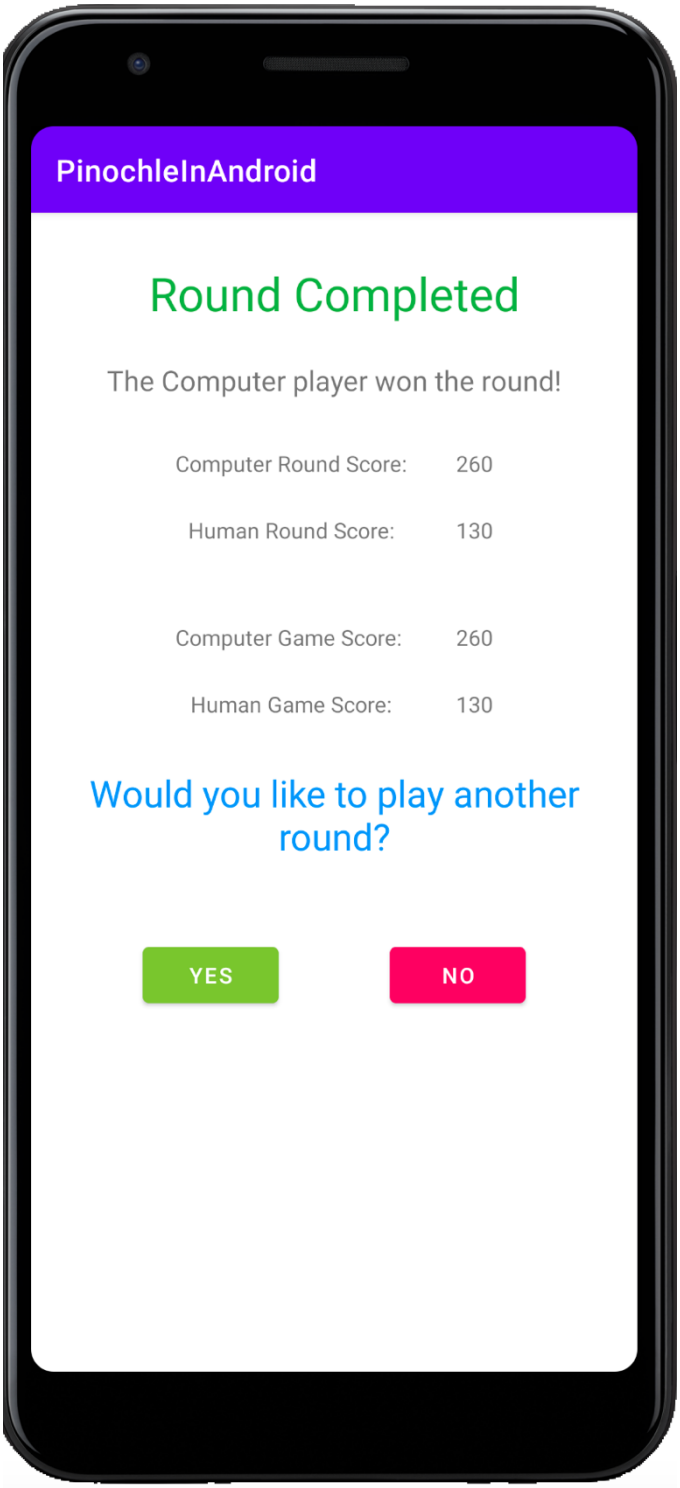
16. After clicking on the save game butotn



17. After clicking on Load Game button in the home screen



18. After a round has ended



19. After a game has ended