Java Android Project Manual

Organization of Programming Languages

Professor Kumar

04/25/2020

Salil Maharjan

---

1 Bug Report

The program does not have any known bugs.

2 Feature Report

2.1 Missing Features

All features reported on the project description and grading rubric from the website has been successfully implemented.

2.2 Extra Features

The user player has the option to use a custom name in the game. If no name is entered, the player name defaults to “User”.

Application has custom icon.

The capture pile is sorted as mentioned in the grading rubric and it is extended so that the hand pile is also sorted for ease of readability.

Stacks and recommended cards are highlighted elegantly.

The computer strategy and user help mode, considers the stacks on the player’s capture pile to give recommendation for the next move.

3 Data structure/Classes Description

There are seven model classes used in the program: Game, Round, Player, User, Computer, Card, and Deck. There are two activity classes: MainGameActivity, and MainRoundActivity.

Game is the main class to start the game. It creates Round class objects as requested by the user and stores information like the round number, total player scores, and the Player class objects. It includes a method to load a saved game from a configuration file. MainGameActivity is the corresponding activity class. This activity handles user input to either start a new game or load a saved game.

The Round class holds all necessary methods to start a round of the game. It stores information about the cards on the deck and the layout during each round. The Round class uses the Deck class for creating a stock pile and to distribute cards. MainRoundActivity handles all the features required to play a round of the game.

Player is an abstract class and has virtual methods. User and Computer inherit Player and provide separate implementations of virtual methods declared in Player class. Methods that are common for both User and Computer player are defined in the Player class like sorting and arranging cards on hand and in the capture pile. Separate implementations for other methods which differ for the two players are defined in their respective classes.

The Deck class uses the Card class to create a standard deck of 52 cards. The Card class holds information about the card face and suit.

Vectors are used to store member variables like layout, player hands and capture piles. Stacks in the game are implemented using Vector of Vectors.

4 Log

March 23, 2020:

- Installed and read about android studio. (30 min)
- Developed and ran the “My First App“ tutorial from the website. (1.5 hour)

March 29, 2020:

- Started go stop on android studio. Created main activity xml and the view. (30 min)
- Added buttons to start and load game and their respective activity classes. (15 min)
- Finished writing the Card class. (30 min)
- Finished writing the Deck class. (30 min)
- Tested class. (15 min)

Total: 4 hours

March 31, 2020:

- Started implementing Game and Round class. (1 hour)

- Testing MVC design pattern using the My First App. (1 hour)
- Wrote function to display alert boxes in the starting activity class. (1 hour)

April 3, 2020:
- Started implementing Player, Computer and User class. (1 hour)
- Finished implementing classes used in the model. [code refactoring to be done to meet android requirements] (2 hours)
- Designed view for the GUI layout. (1 hour)
- Connected buttons to the view to the model classes to get the game running. (1 hour)
- App crashing due to low memory. Added code to Android manifest to increase heap size. (1 hour)
- Fixing the view design so that cards can be viewed more clearly. (1 hour)

April 4, 2020:
- Wrote an interface to make the model and view classes to communicate. (1 hour)
- Connected view components to functions in the model using the interface as a callback tool. (1 hour)
- Made card images button so that they can be clicked. (1 hour)
- Ran and debugging errors in the game. (1 hour)
- Found updates to be unsynchronized in the view and model classes. (1 hour)
- Found that the problem was because of the callback interface. (1 hour)
- Changing design pattern by reading more literature on MVC pattern. (1 hour)

April 6, 2020:
- Added functionality for user to enter player's name at the start of the game. (1 hour)
- Refactoring code so that the MVC design pattern is followed. (1 hours)
- Debugging and testing, found an error of the layout not updating appropriately. (1 hour)

April 9, 2020:
- Debugging code and found the error where an assignment operator was used for making a temporary copy of an object. In java, this creates a reference to the original object so there were random modifications being made to the main object. Solved issue by using Clone method. (1 hour)
- Debugging and other bug fixes. (1 hour)

April 22, 2020:
- Wrote methods to load and save game. (1 hour)
- Tested out test cases I had from the previous versions of the game. (1 hour)
- Debugging the issues found from the test cases. (1 hour)
- Refactored code and commented out code used for debugging. (1 hour)

April 24, 2020:

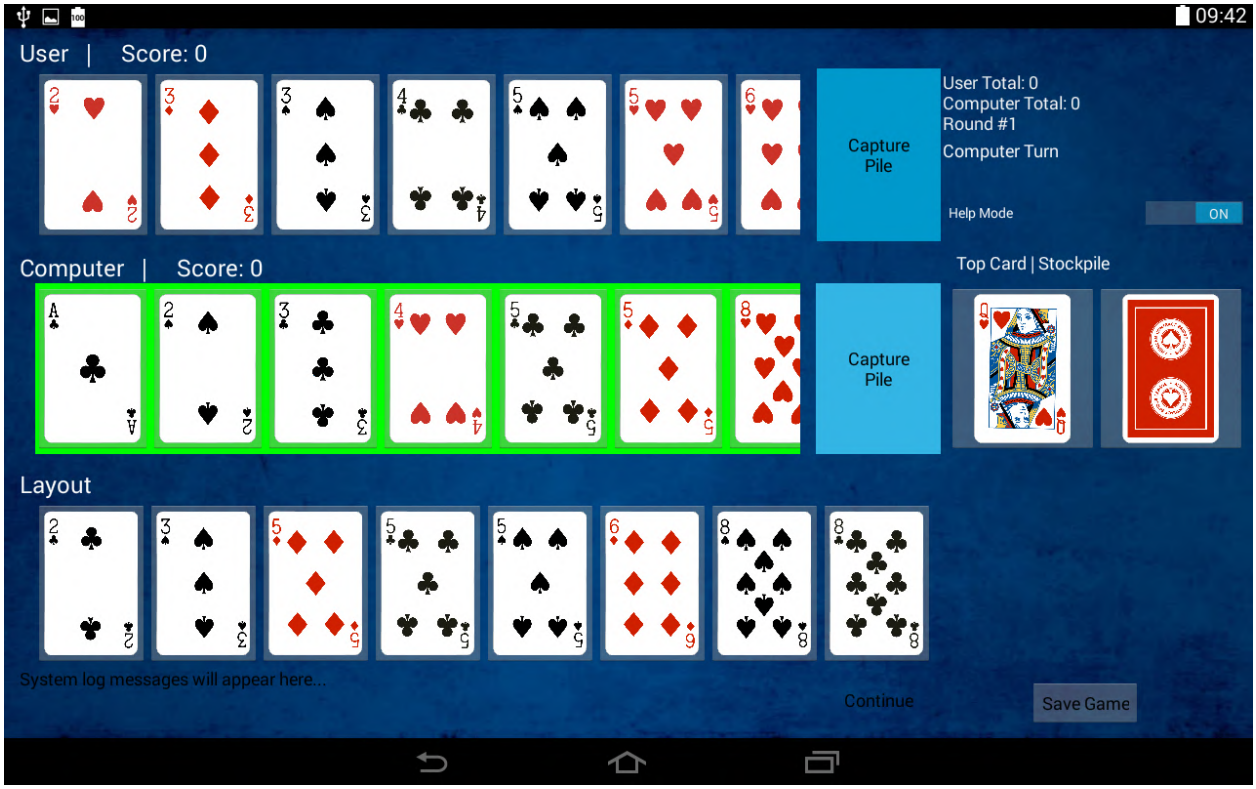
- Adding documentation and comments. (1 hour)
- Wrote the manual. (1 hour)
- Filled out grading rubric and prepared for submission. (1 hour)

Total: 32 hours

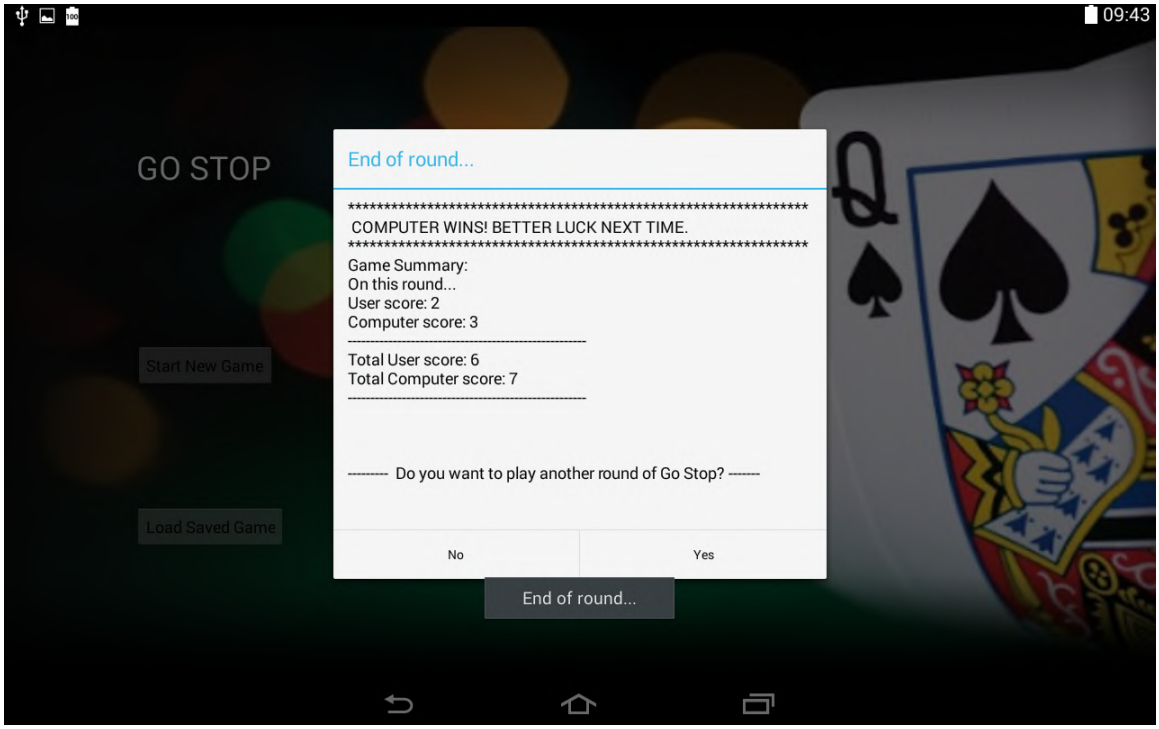
5 Running Instructions

The application was built using Android Studio. It can be imported and compiled as an Android project.

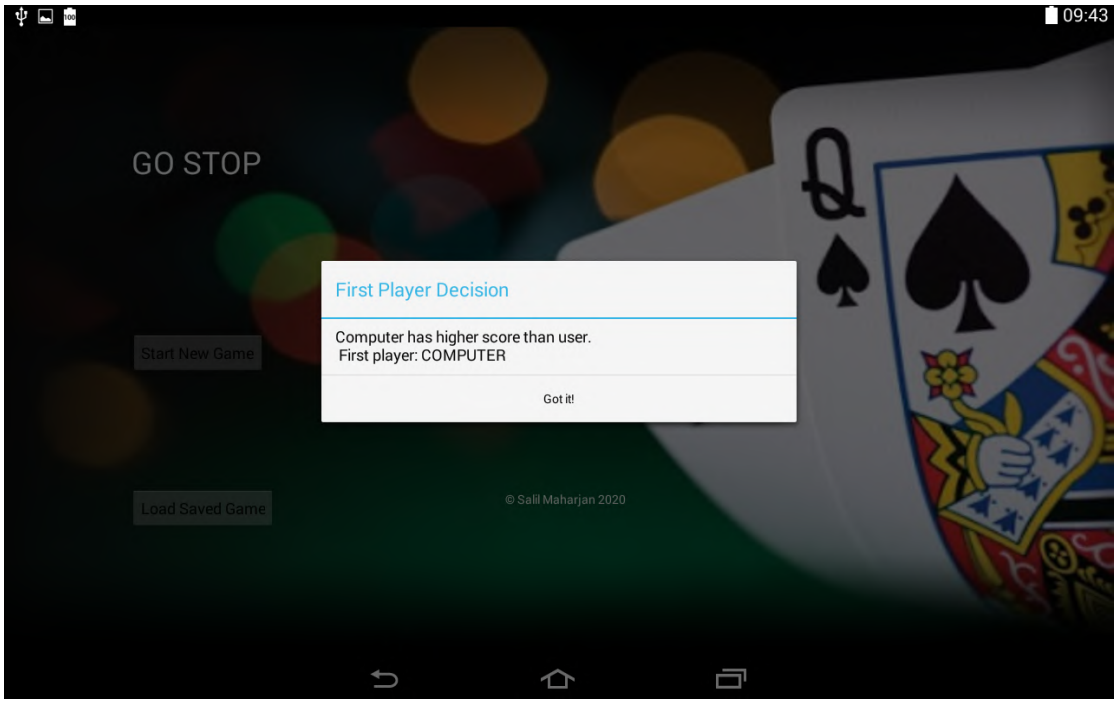
6 Program Screenshots:



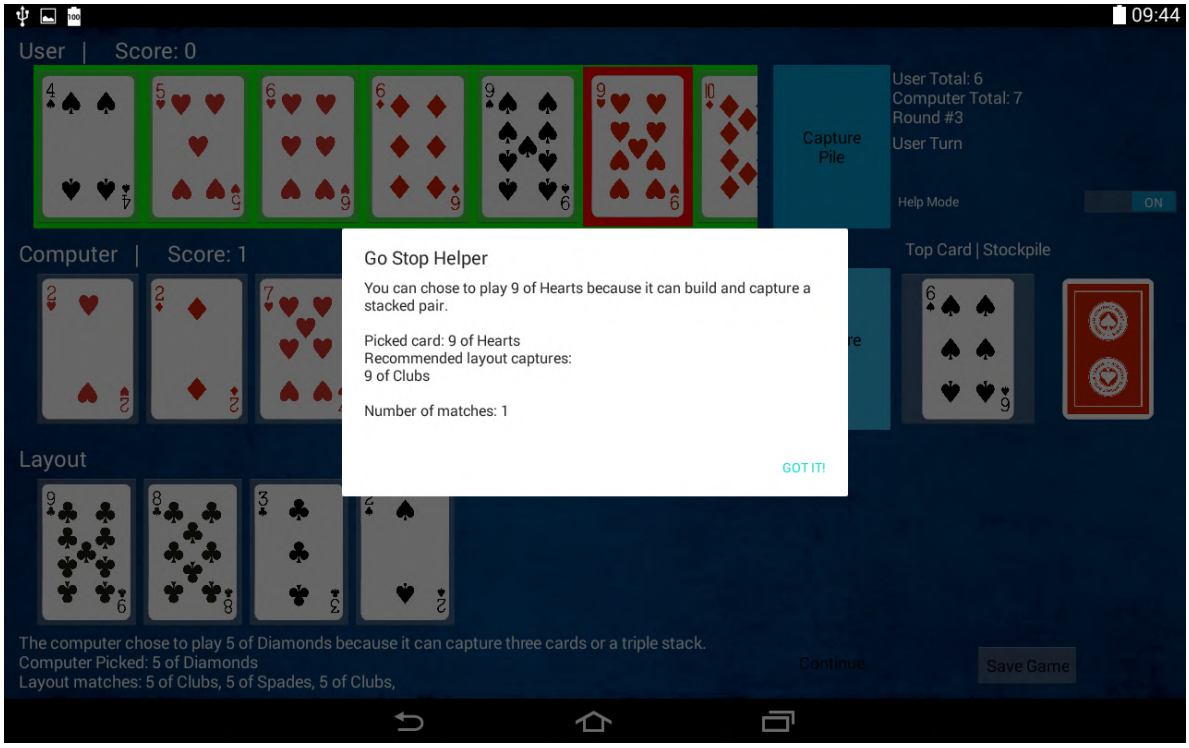
Screenshot 1: Loaded game and active player highlight.



Screenshot 2: End of round results and prompt to play again.

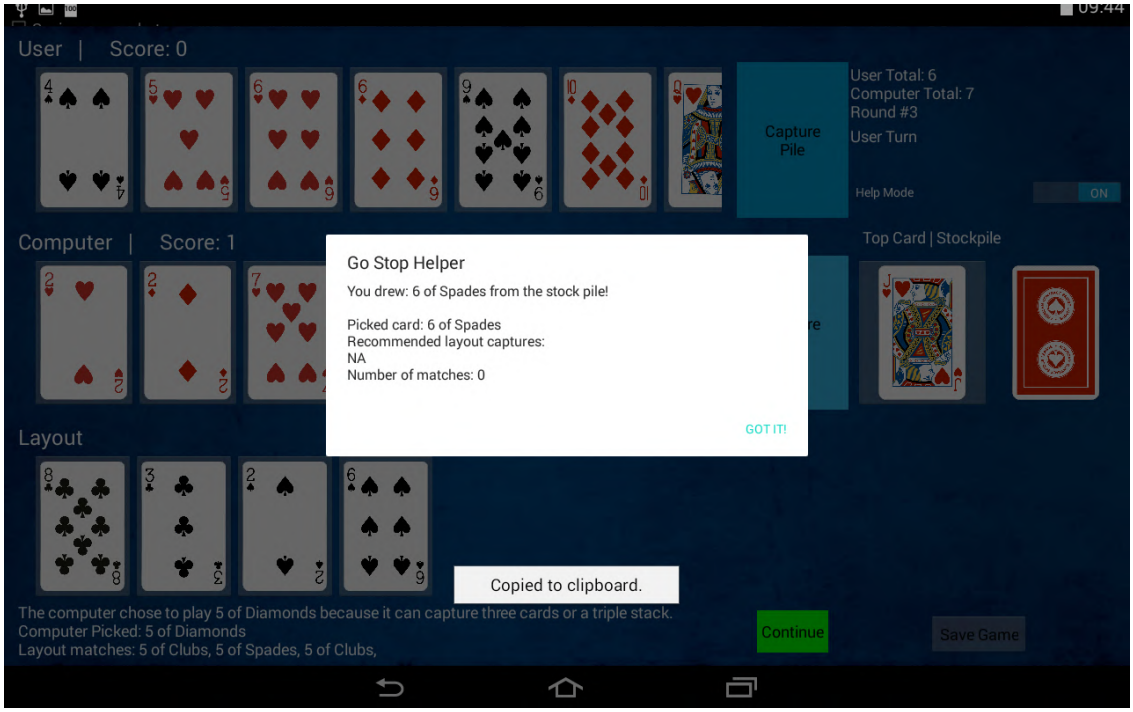


Screenshot 3: First player decision message.



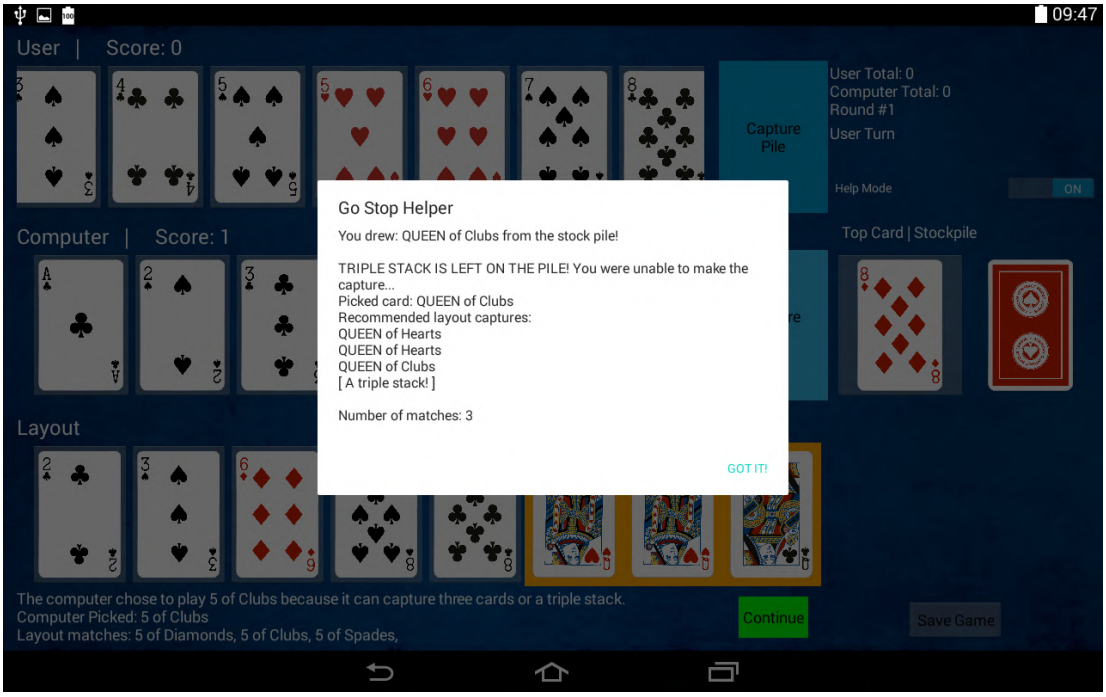
Screenshot 4: User Turn with help mode logic and highlighted recommended card.

Screenshot 5: Stack highlight and prompt to highlight to prompt to play stock pile.

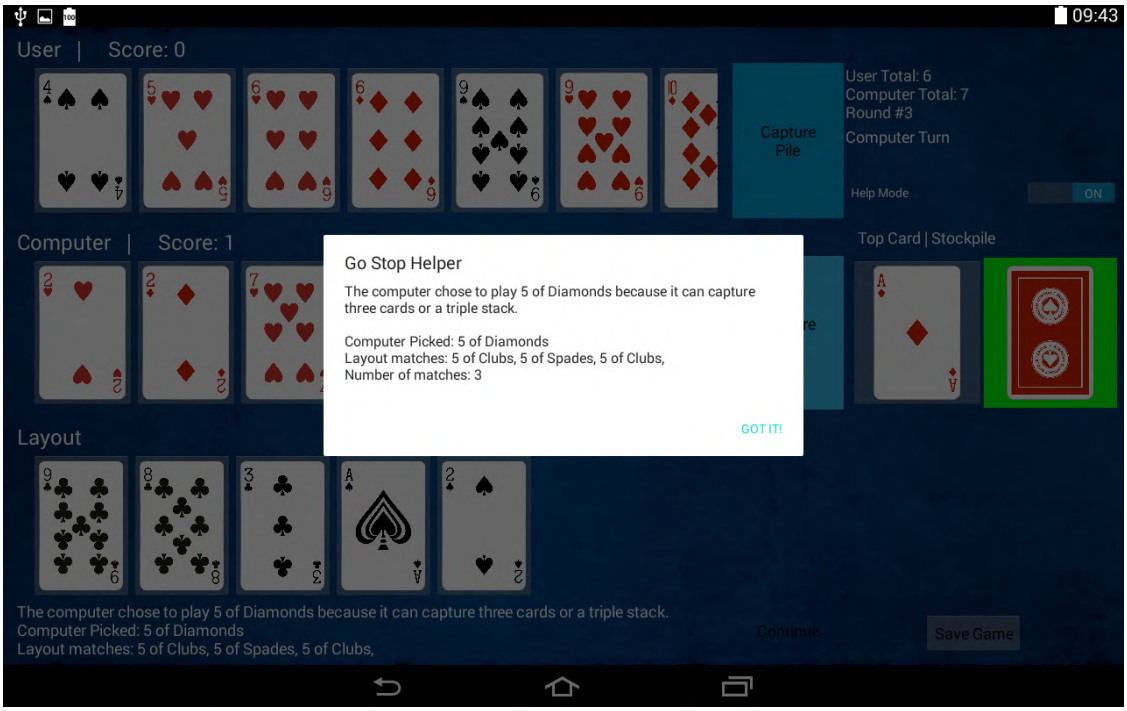


Screenshot 6: Stock pile move helper.

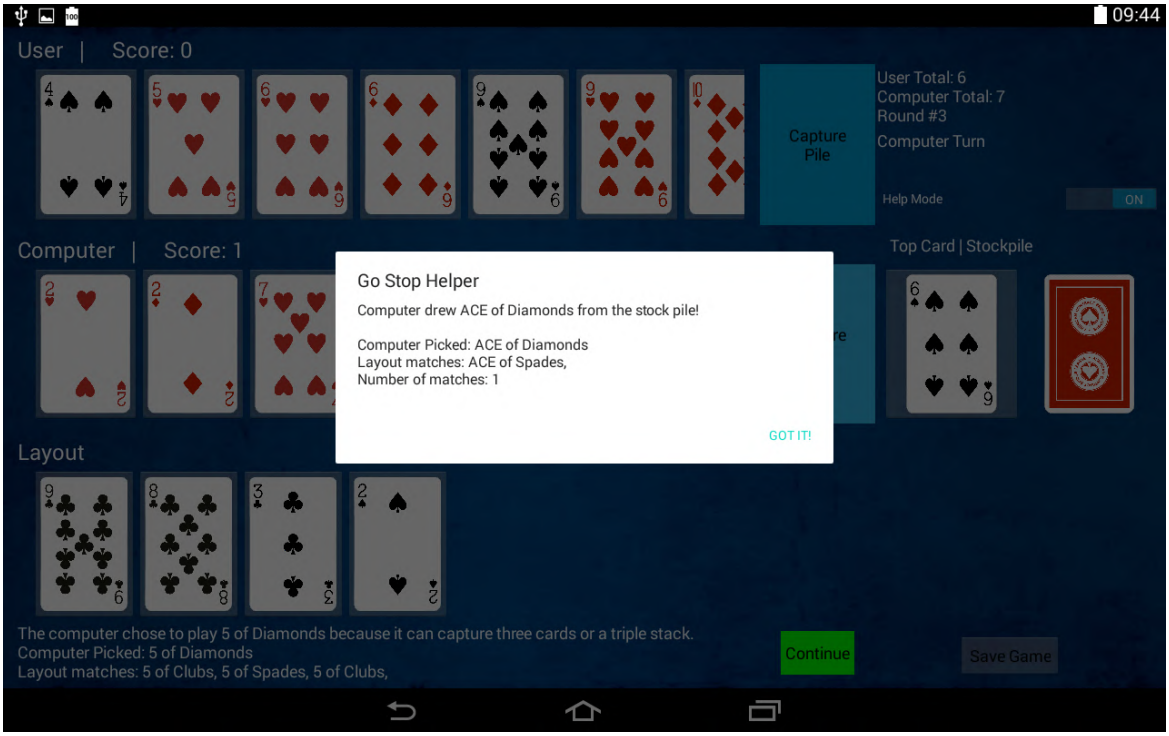
Screenshot 7: Stacks in capture pile that give the player a point are highlighted in orange. Active player in green highlight.



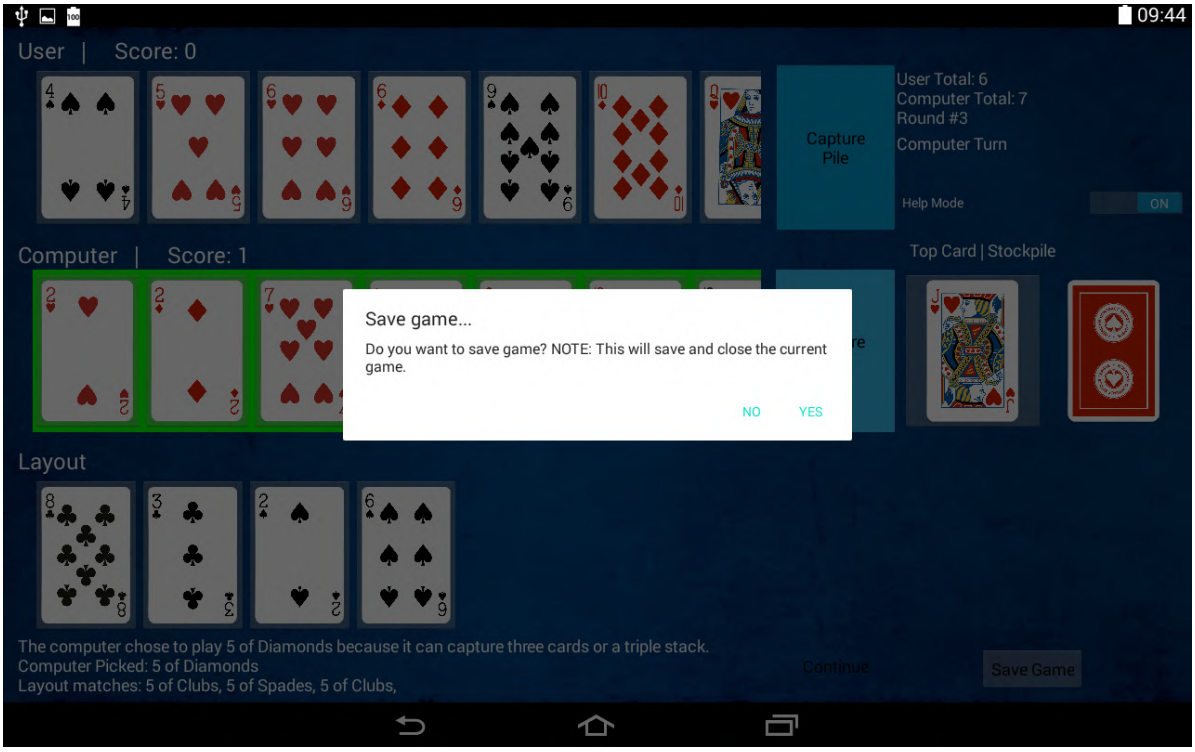
Screenshot 8: Triple stack notification. (When it is left or captured from the layout)



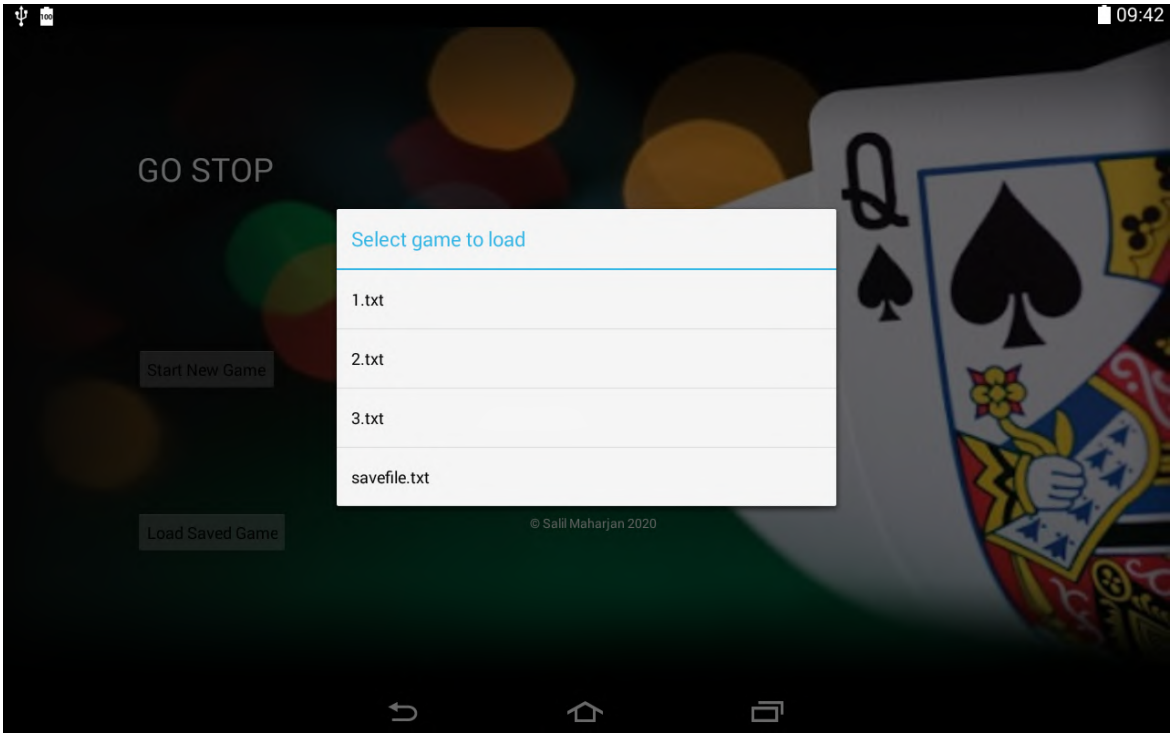
Screenshot 9: Game setup and computer's hand play turn.



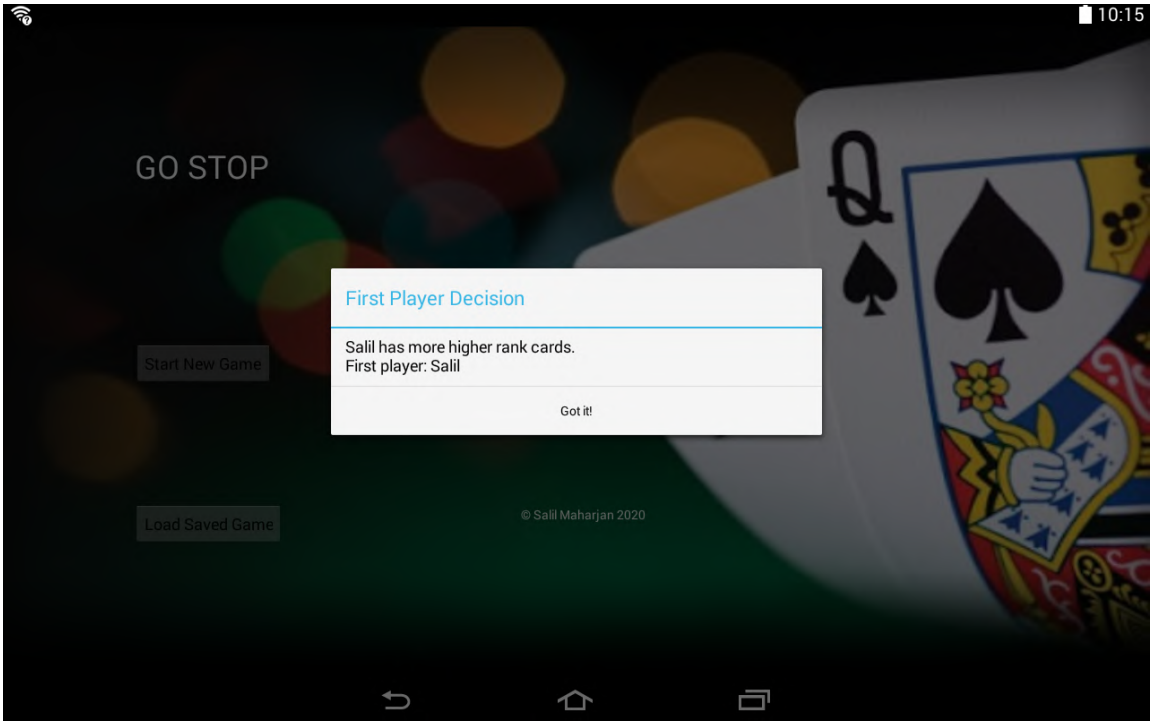
Screenshot 10: Computer's stock play turn.



Screenshots 11: Saving the game.

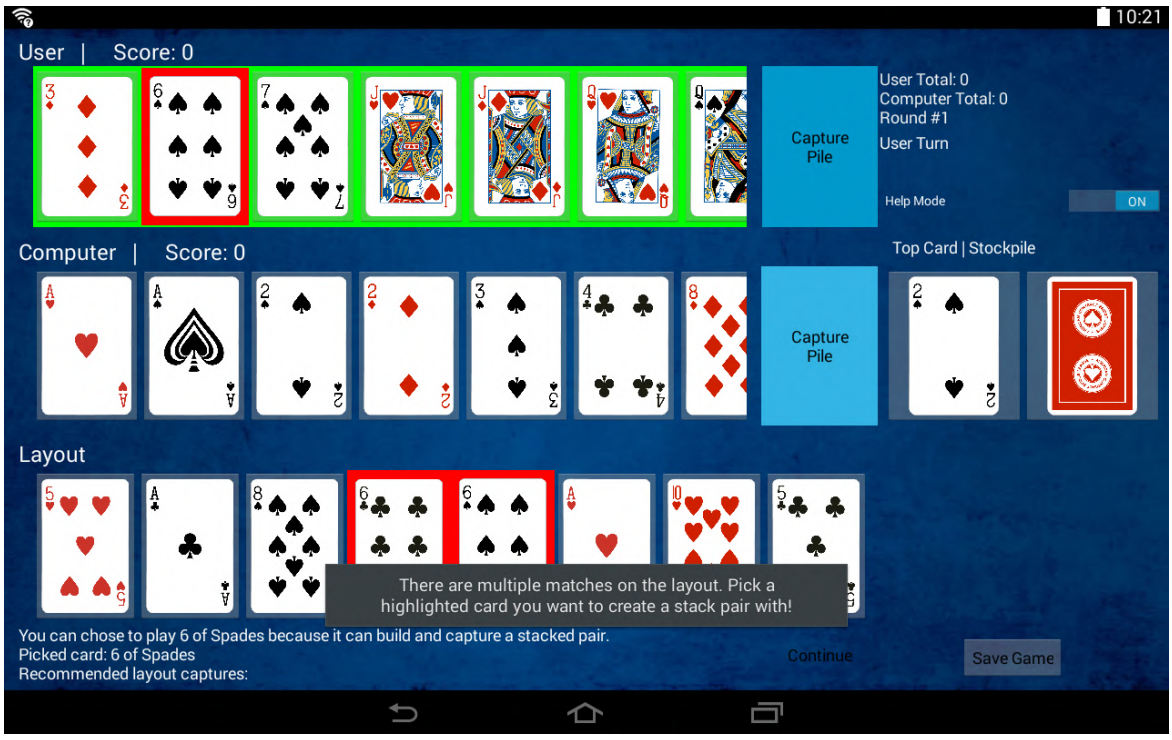


Screenshots 11: Loading saved game.

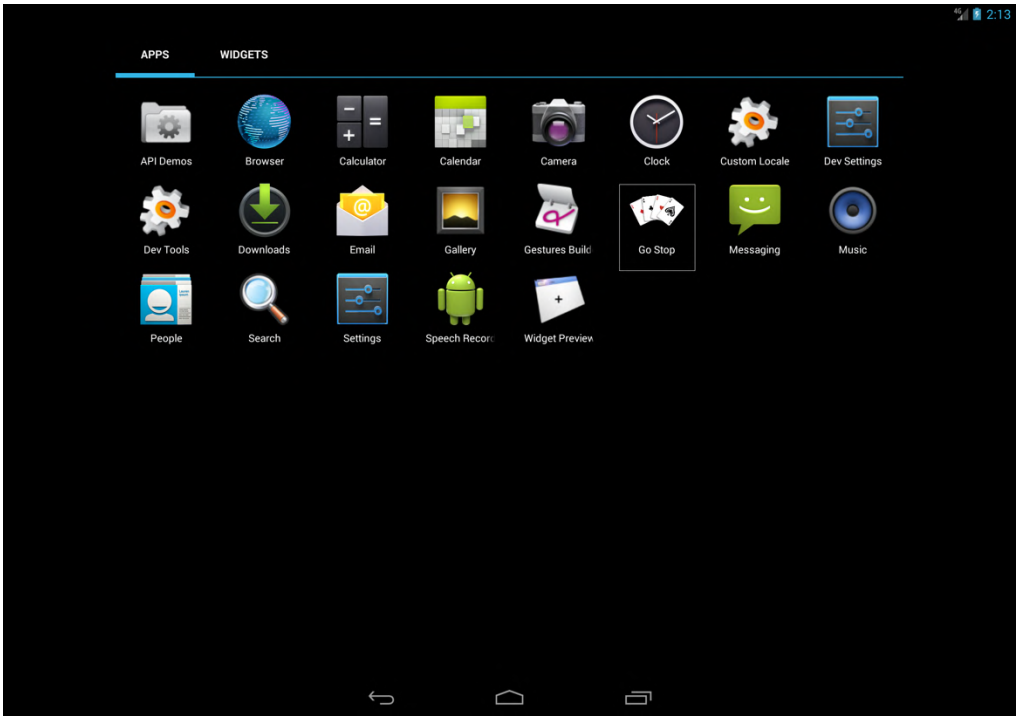


Screenshots 12: Custom name input

Screenshots 13: Custom name and board configuration.



Screenshots 14: User Option highlighted to pick card to stack with.



Screenshots 15: Custom application icon

Thank you