# Bug Report

1. The tournament wasn’t ending because the score was not being updates
   1. Fixed it during the demo. Fix was that the updating the score was happening inside the if(tournament ended) logic.
2. The new round generated after a round completed had some tiles in the player hands already
   1. This was because the hands were not flushed before dealing the new round. Fixed it during the demo by flushing the hand before the round was initialized.

# Features Implemented

1. The computer uses AI logic, a version of best-first-search, to come up with its move and provides hint with the same logic
2. The UI has actual dominos instead of buttons with numbers

# Description of Data Structures and Classes

## Classes:

1. **layout** class to implement the layout
2. **hand** class to hold the hand of a player
3. **stock** class to hold the stock/boneyard for the round
4. **player** class in which playing strategies must be implemented
5. **human** class which inherits from player class and implements all the validation of input from the human player
6. **computer** class which inherits from player class and implements all the strategies used by the computer
7. **round** class which implements the entire round
8. **tournament** class which implements a tournament
9. **GlobalImports** header file the contains imports, declarations and static utility functions that is common to all the classes
10. **Longana:** class to initialize the players, tournament and load game from file.
11. **Domino:** Class to hold a domino. Includes the left and right pip as integer
12. **Move:** class to hold the player move. It consists of Domino and Side
13. **Side:** Enum to indicate the playing side
14. **RoundActivity:** Front-end class to run the round
15. **TournamentActivity:** Front-end class to run the Tournament

## Data Structures:

1. Vector of Domino objects for storing all the tiles
   1. Left and right for layout
   2. Implemented as a stack for boneyard
   3. hand

# Log:

10/18/17

* Basic setup for the Android front end and created the class for the model

2 hrs

10/19/17

* Implemented the round play logic and displayed in the view

3 hrs

10/29/17

* Wrote the hint logic
* Linked the hint logic with computer play logic
* Human can place tile on the view and the validations are working
* Computer can play with the hint strategy.
* Basic Game logic has been implemented. Round and Tournament still need to be hooked up

5 hrs

11/2/17

* Created the tournament activity and implemented the logic in the front end
* Linked the tournament and Round activity

2 hrs

11/5/17

* Deserializing the files done (Android File system is a pain)

2 hrs

11/9/17

* Serializing the game is done

2 hrs

11/11/17

* Implemented the image buttons

4 hrs

Total: 20 hrs

# Screenshots:

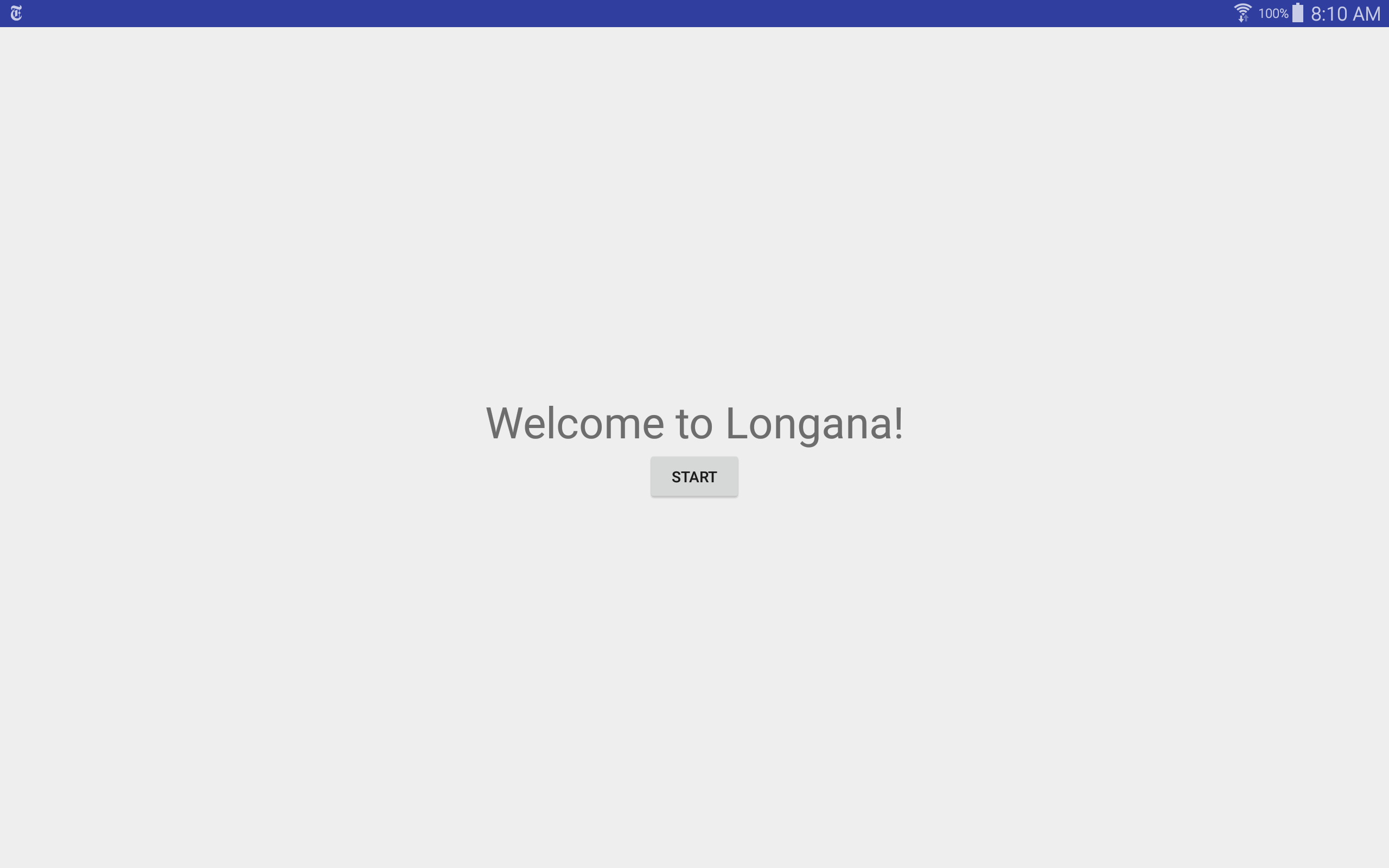


Figure 1: Start of the game

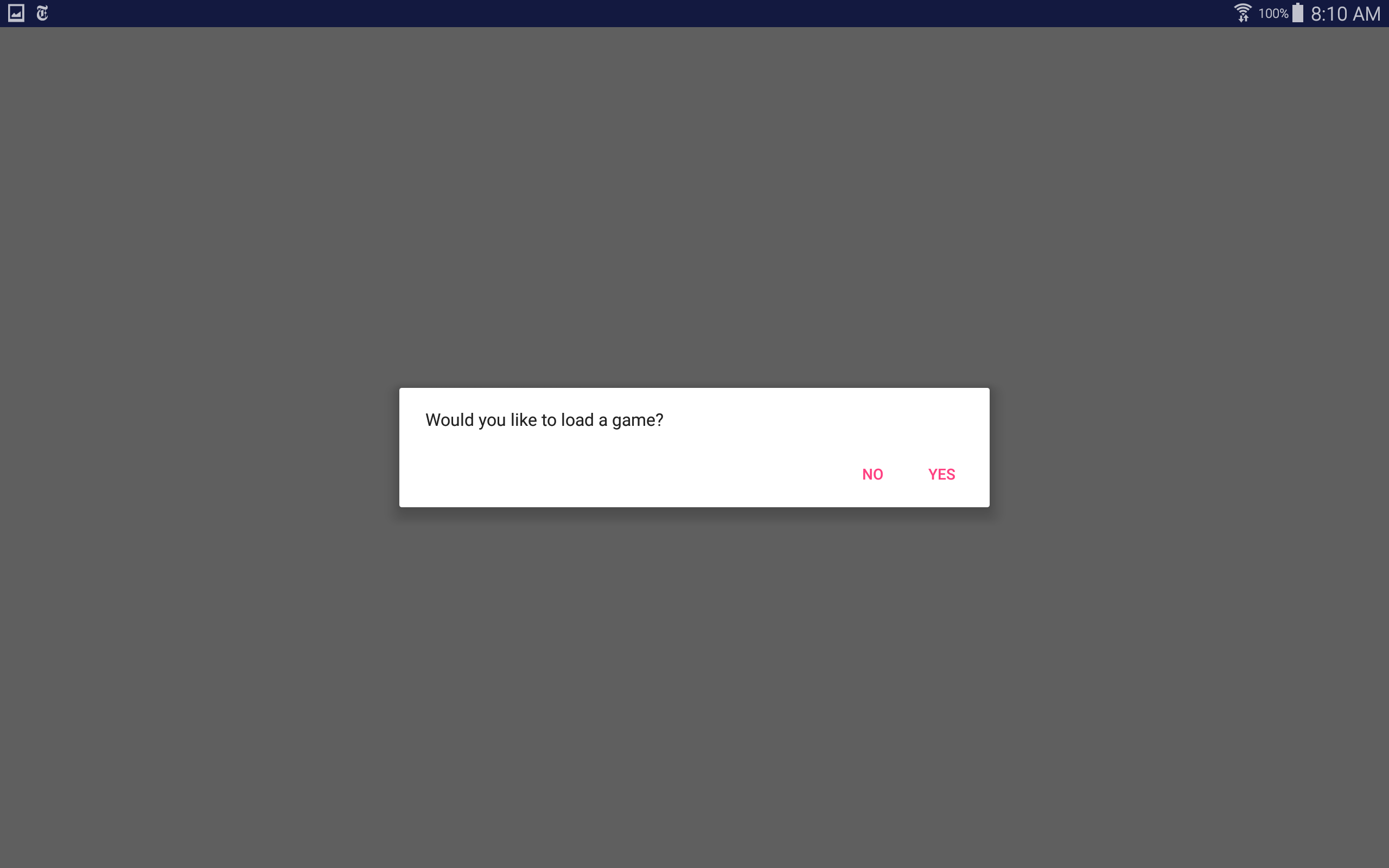


Figure 2: Asking user whether they want to load a game

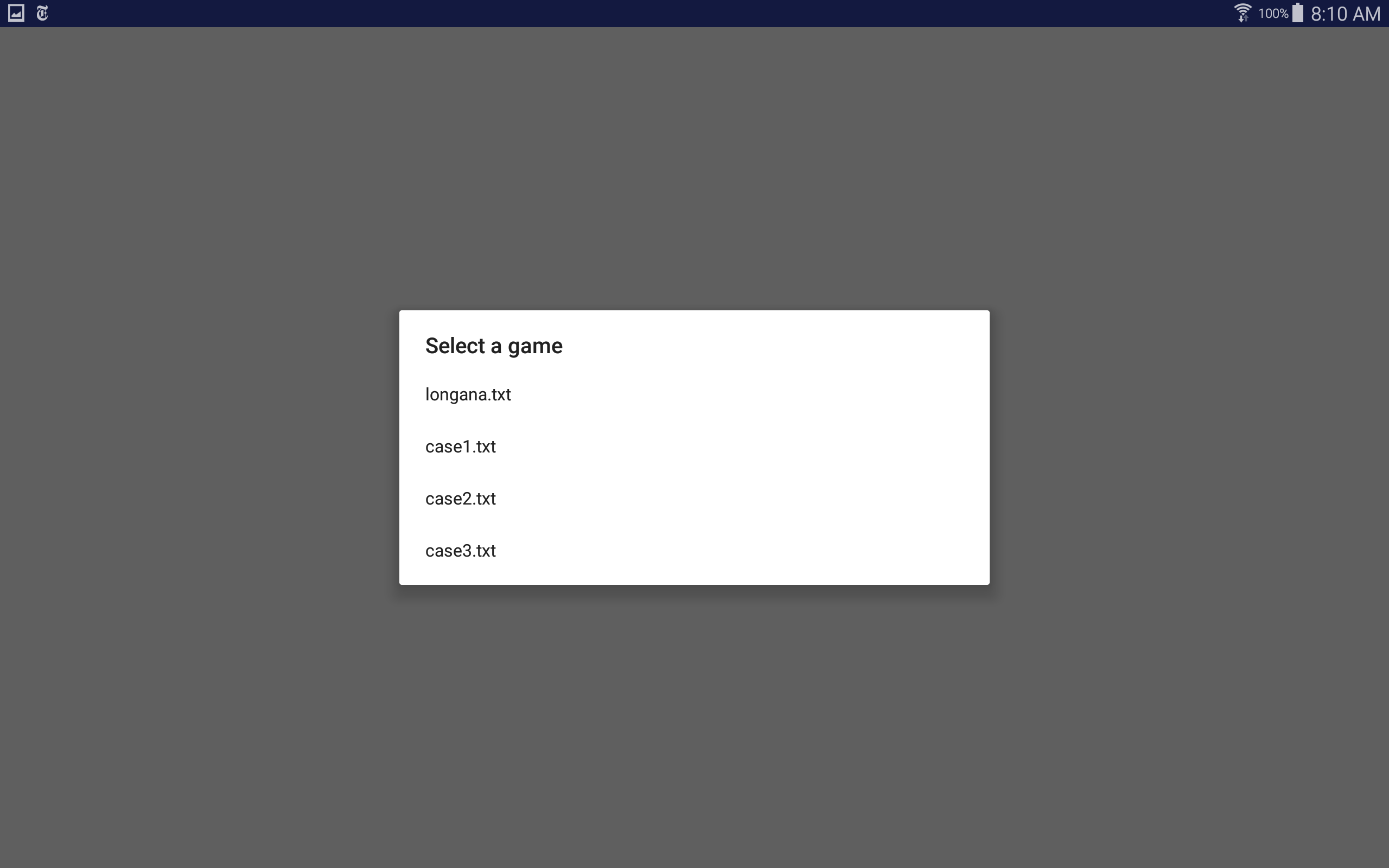


Figure 3: Letting the user choose from what file they want to load

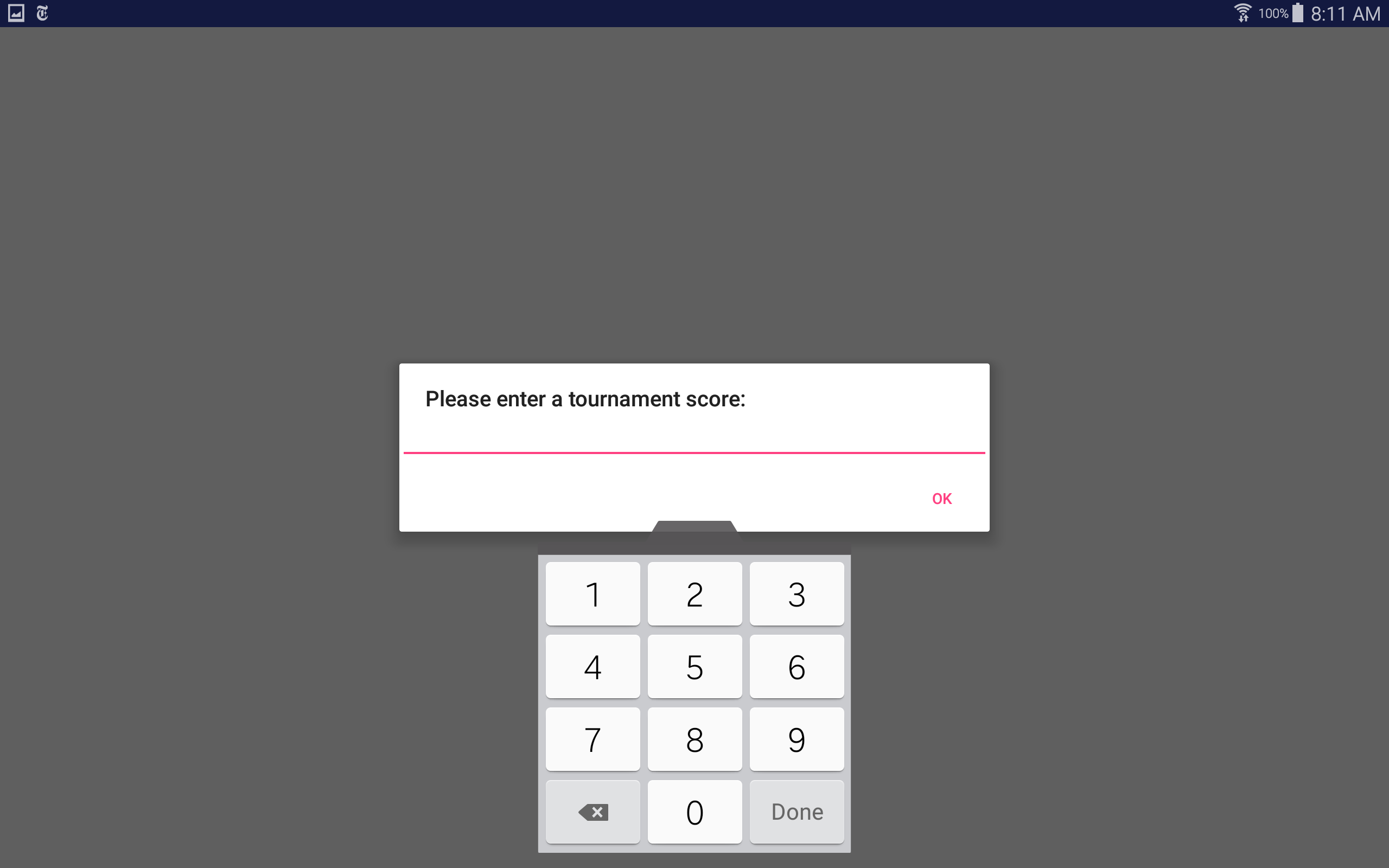


Figure 4: Asking for the tournament score in case of new game

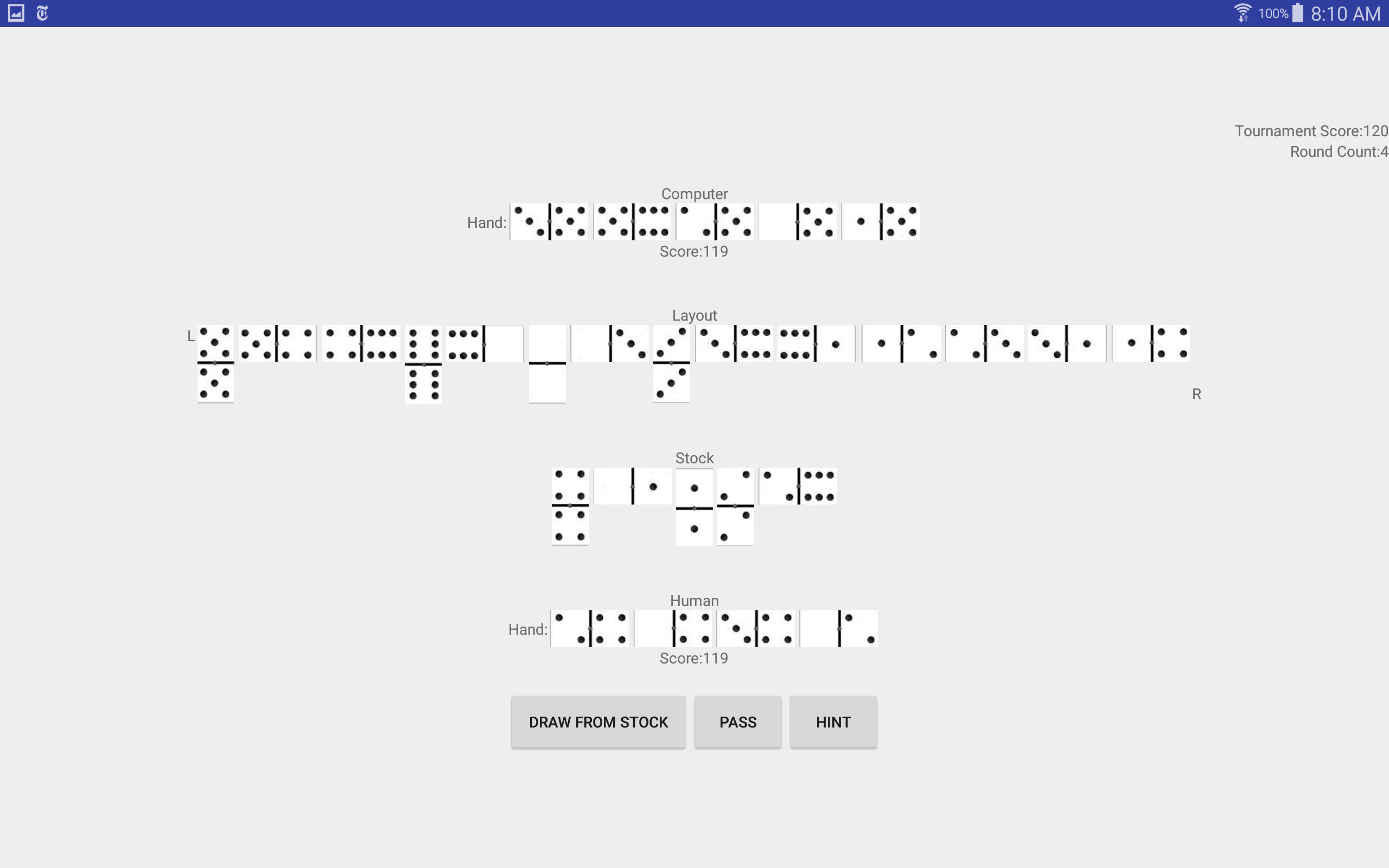


Figure 5: Round Activity

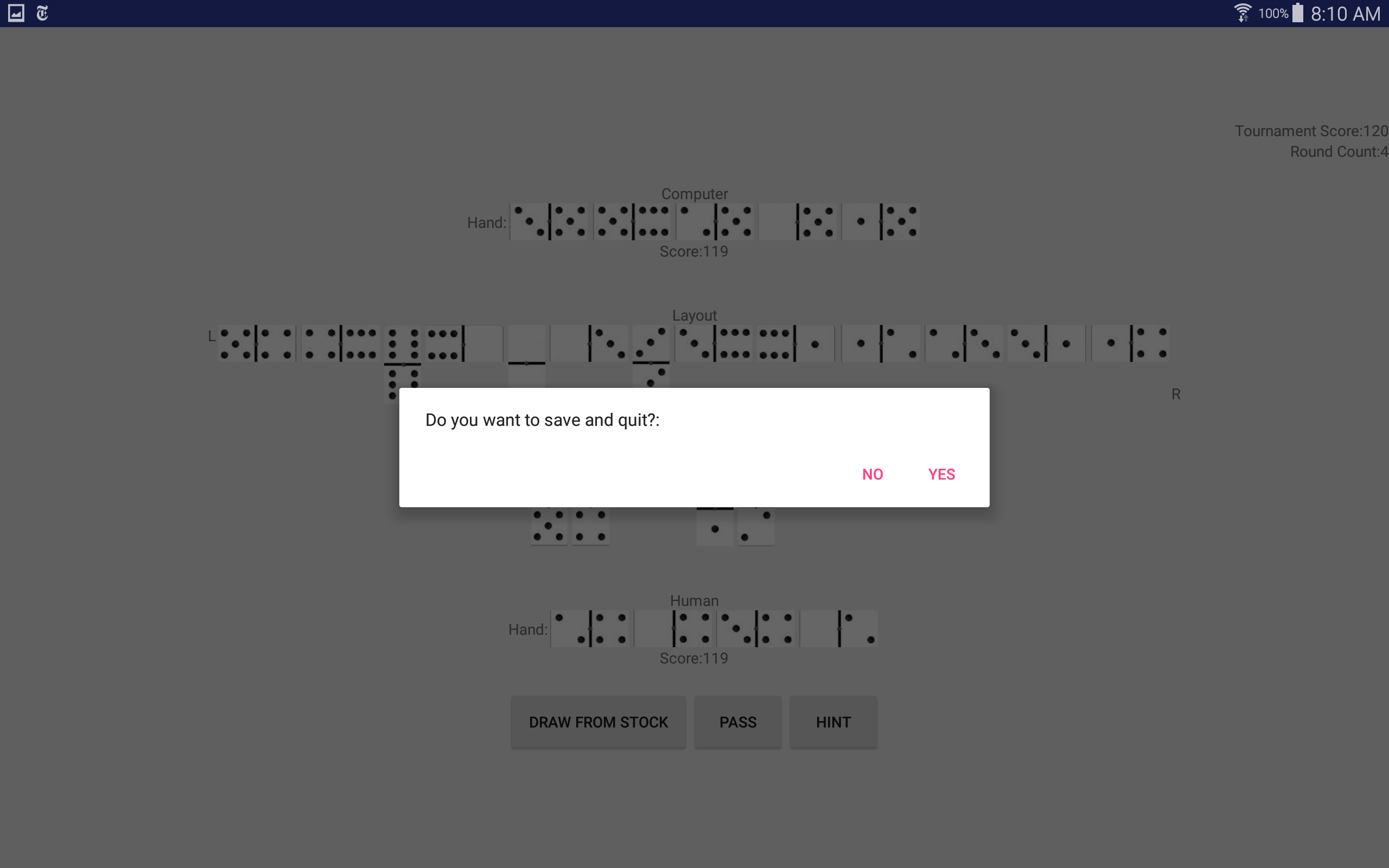


Figure 6: Asking user if they want to save and quit

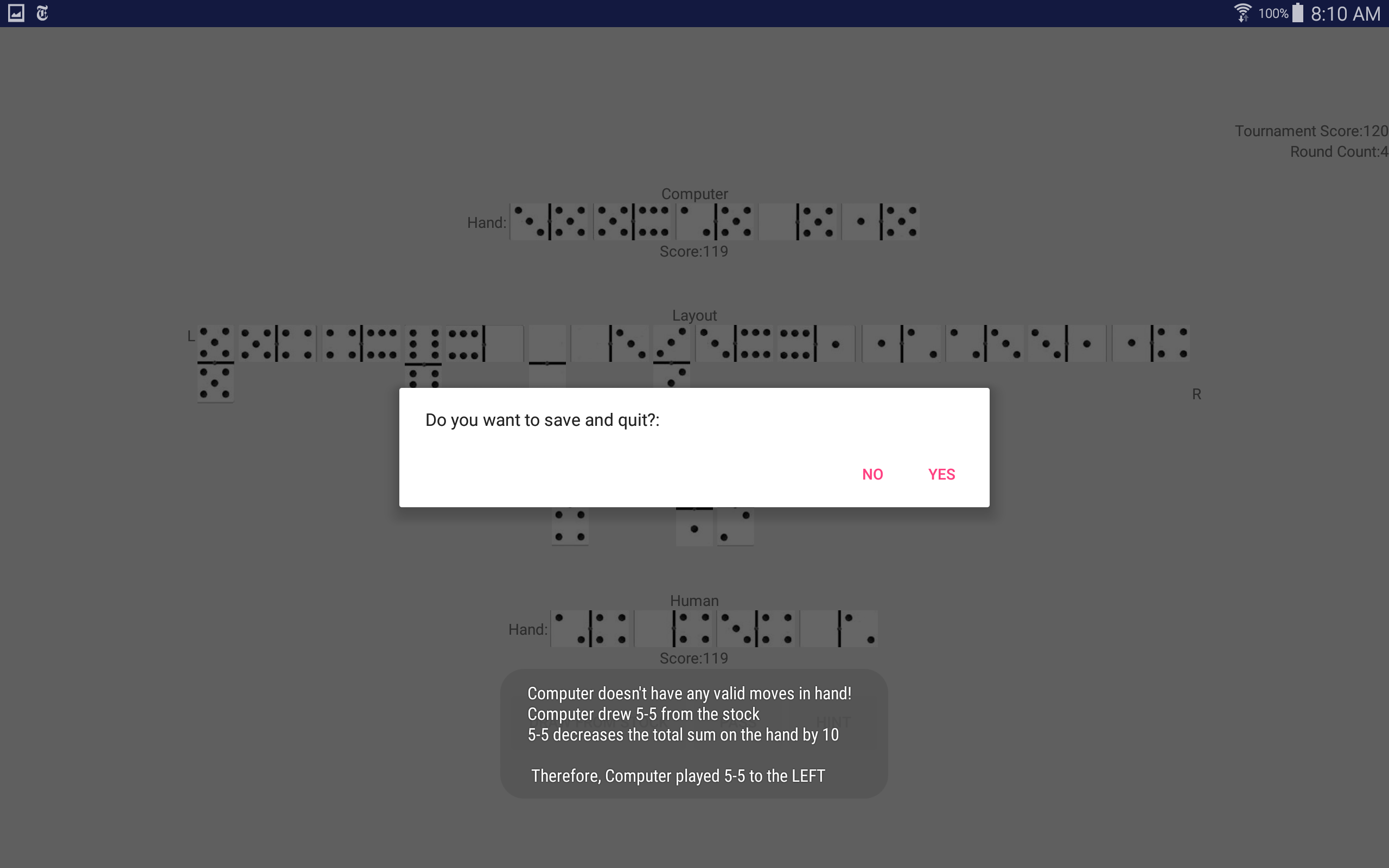


Figure 7: Computer player making the move

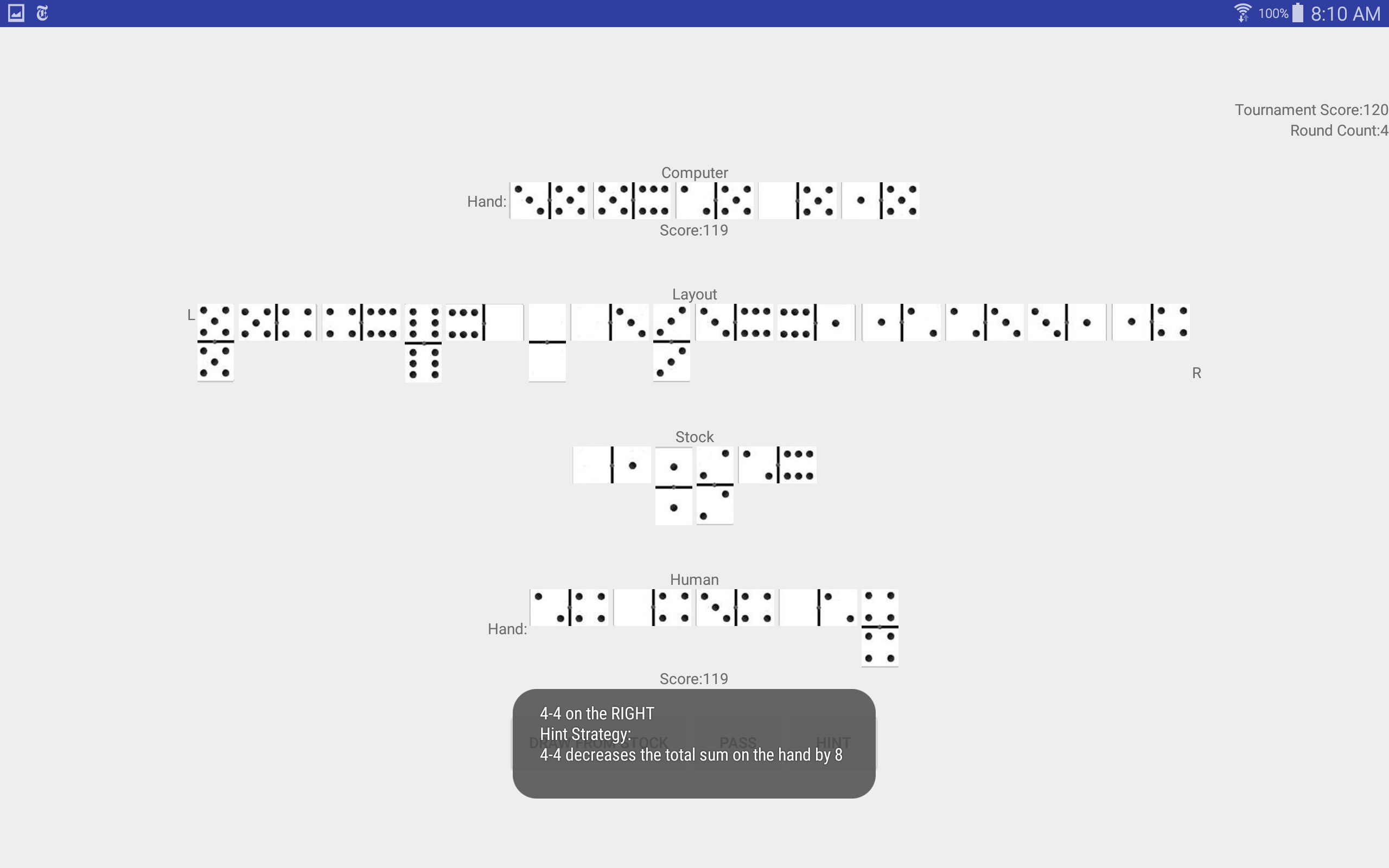


Figure 8: Human player using hint for suggestion