Evaluation

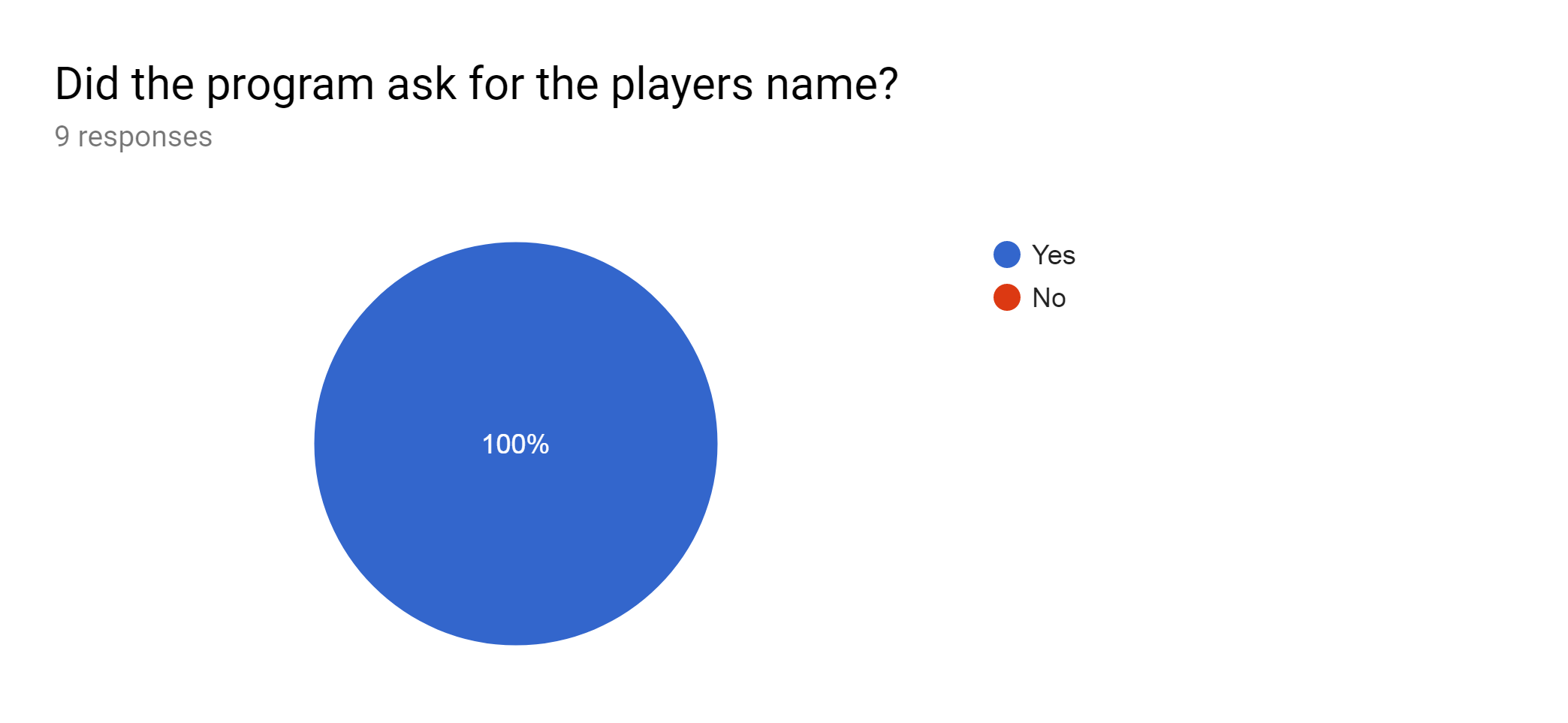
* Requirements Specification Evaluation

| Stage of Requirement specification | Point | Completed? | If not, why? |
| --- | --- | --- | --- |
| Functional Requirements | Read in data from csv file | Yes |  |
|  | Validate inputs by using buttons | Yes |  |
|  | Efficient use of 2D Array | Yes |  |
|  | Ask users for their name | Yes |  |
|  | Display user interface with all correct information, updating every turn | Yes |  |
|  | Ask current player if they want to buy property | Yes |  |
|  | Roll dice button work correctly and for doubles | Yes |  |
|  | Give the active player £200 for each time they pass GO and correct the position | Yes |  |
|  | End turn button work correctly and change the turn to the other player | Yes |  |
|  | Allow users to buy houses/hotels and properties | Yes |  |
|  | Chance and community chest cards work and display a message | Yes |  |
|  | Jail function work correctly, charing fine and GO TO JAIL tile changing the position without collecting money for passing GO | Yes |  |
|  | Free parking tile give the user the total calculated from all fines and charges from income tax, super tax and chance/community chest cards and resetting | Yes |  |
|  | Train stations work and charge correct rent based on how many stations are owned by the owner | Yes |  |
|  | Charge correct rent on each property based on how many houses or if a hotel is present | Yes |  |
|  | Super tax and income tax work correctly taking away the money and adding it to the free parking total | Yes |  |
|  | Display the winner and loser at the end of the game | Yes |  |
| End user requirements | Between 10 and 15 cards (for chance and community chest) | Yes |  |
|  | 2 player game | Yes |  |
|  | Use of buttons | Yes |  |
|  | Using two dice | Yes |  |
|  | Using the £ sign | Yes |  |
| Scope and Boundaries | Roll dice and move around board | Yes |  |
|  | Buy properties/hotels/stations/houses | Yes |  |
|  | Calculate rent costs | Yes |  |
|  | Automated UI which updates information as the users play | Yes |  |
|  | Working action tiles like income tax, GO TO JAIL and free parking etc | Yes |  |
|  | Chance and community chest system | Yes |  |
|  | Calculate rent for stations based on how many owned | Yes |  |
|  | Program be able to detect when a full street is owned | Yes |  |
|  | Only be able to play up to 2 players | Yes |  |
|  | Trading abilities between players,with cards and money when they can’t afford rent etc | No | Due to time constraints |
|  | Auctions for properties when the active user does not want to buy them | No | Due to time constraints |

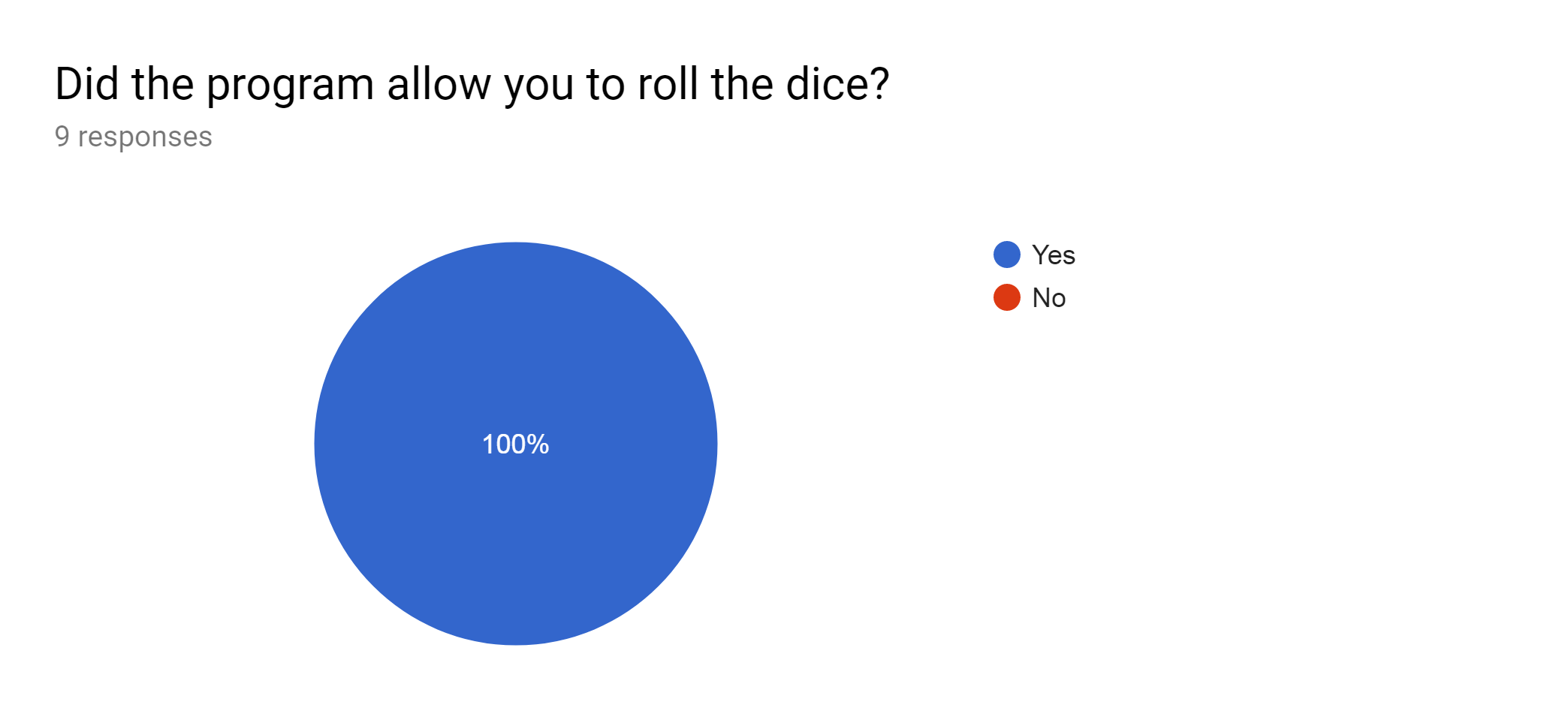
The final testing:

| Testing | Successful/unsuccessful | Evidence |
| --- | --- | --- |
| Entering names | Successful |  |
| Display position before and after roll | Successful |  |
| Roll Dice | Successful |  |
| End Turn | Successful |  |
| Buy Property | Successful |  |
| Read in all data | Successful |  |
| Charge Rent | Successful | Before balance £1500:    After balance £1450: |
| Chance and Community Chest | Successful | Community Chest:    Chance: |
| Rolling a double | Successful | As you can see the roll dice button is still active after rolling the dice ^ |
| Free Parking | Successful |  |
| Jail | Successful |  |
| Super Tax and Income Tax | Successful |  |
| Passing GO | Successful |  |
| Display Winner | Successful |  |

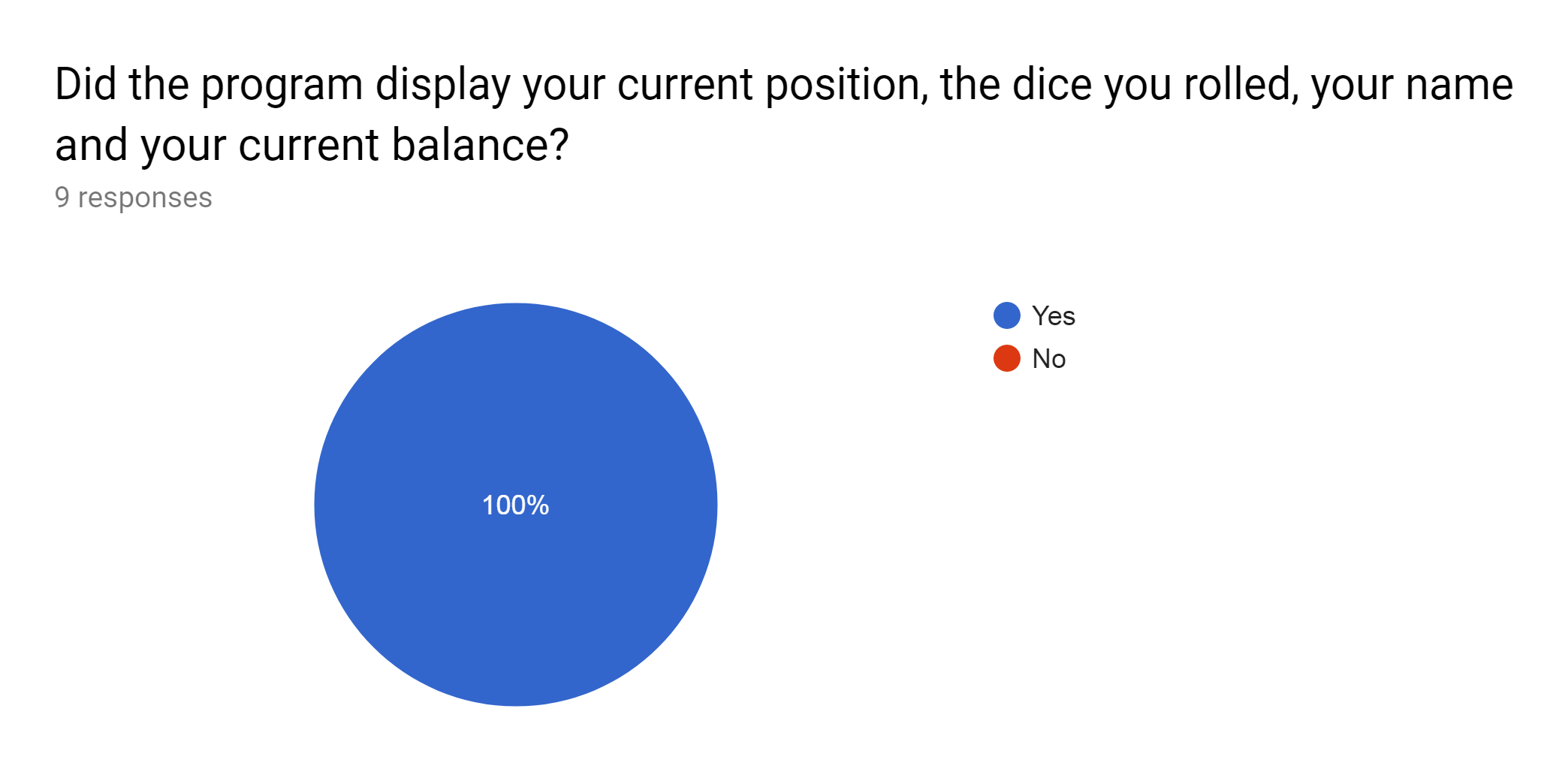
The end-user testing:



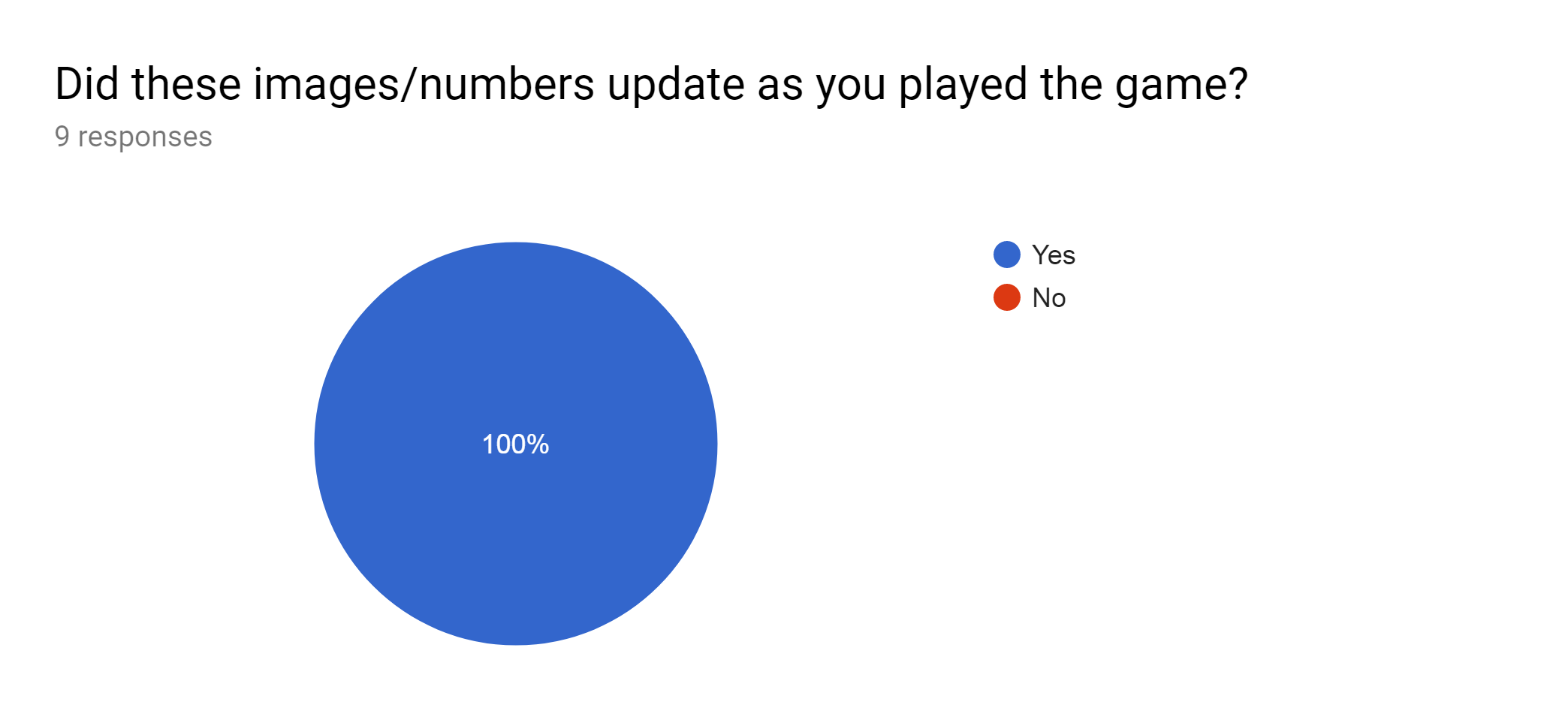
My program successfully asked for the users name



My program successfully allowed the user to roll the dice



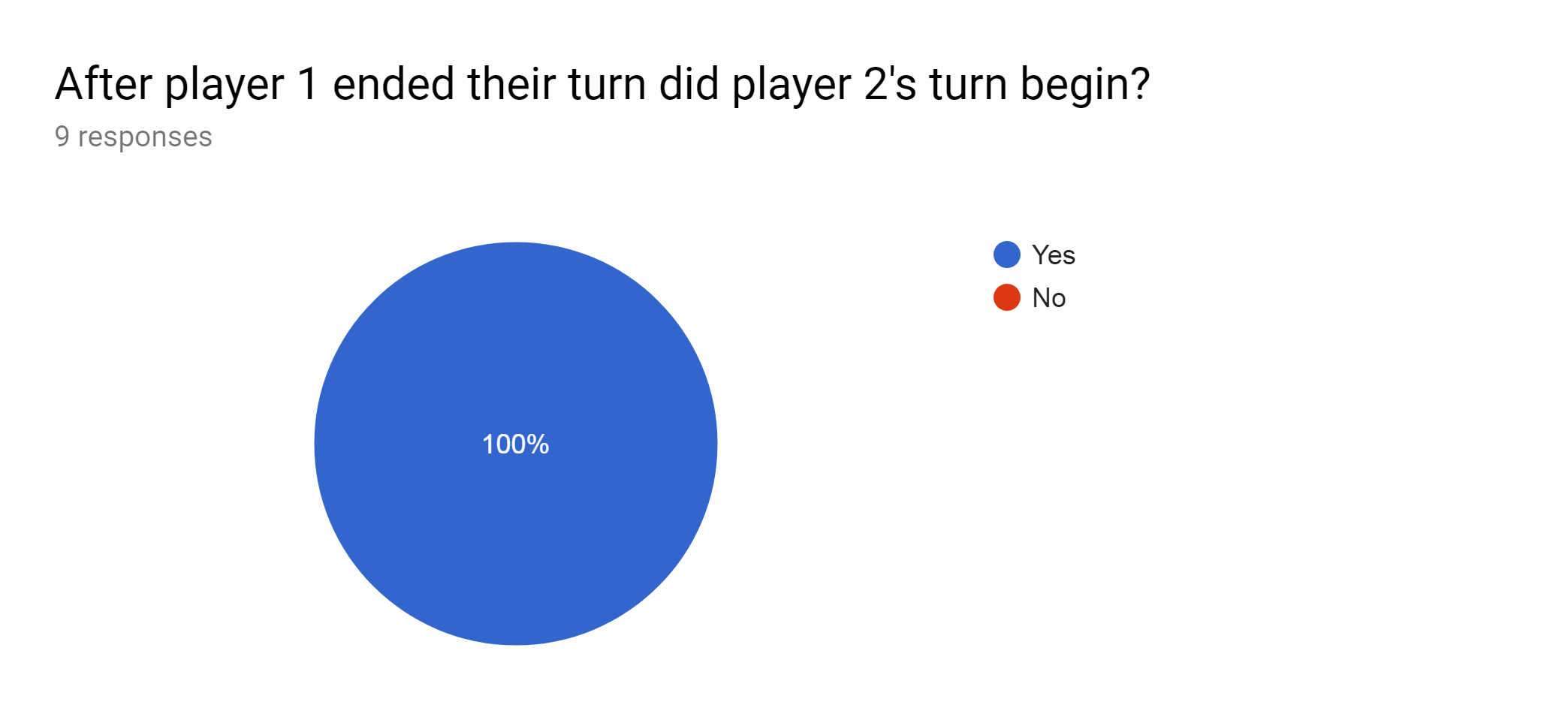
My program successfully displayed the user interface I desired



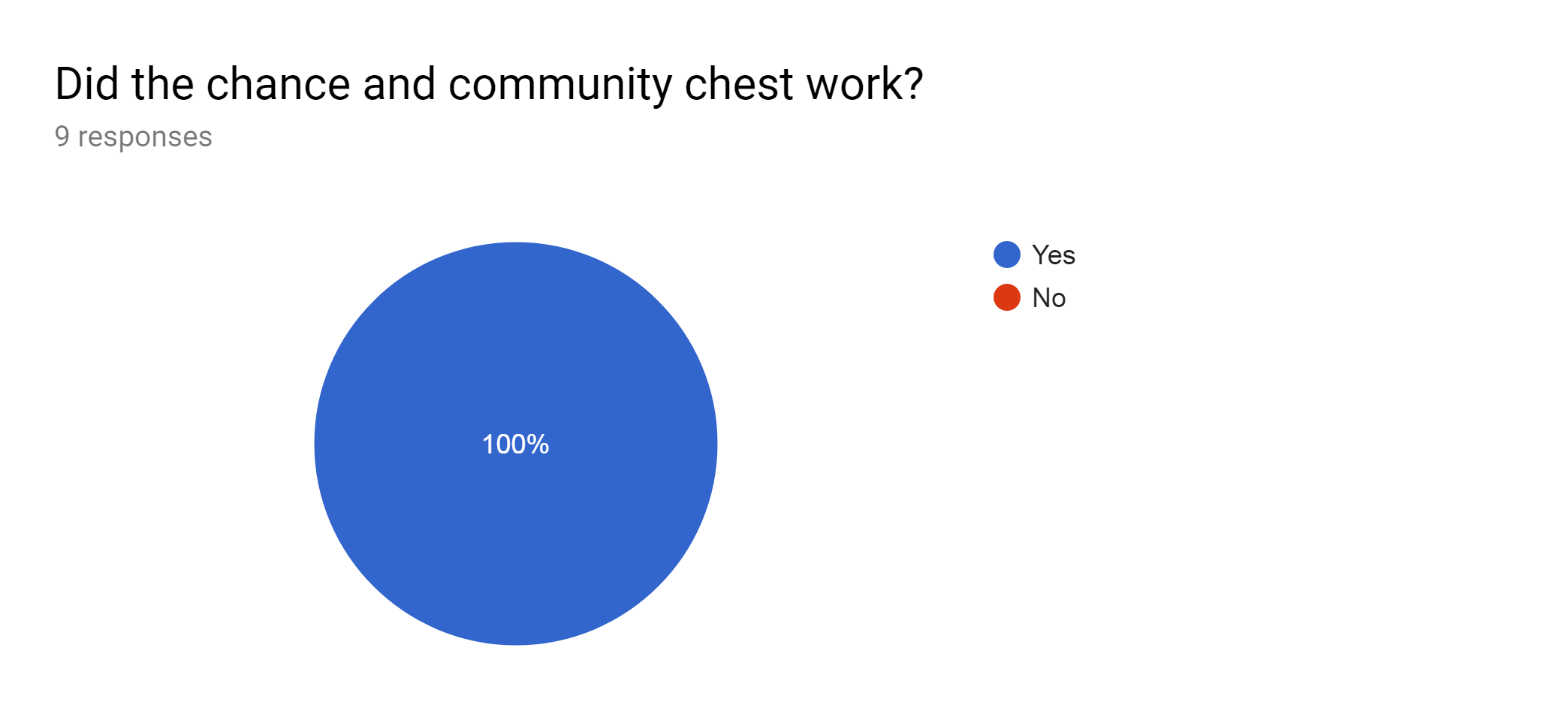
My program successfully updated the user interface after each move/turn



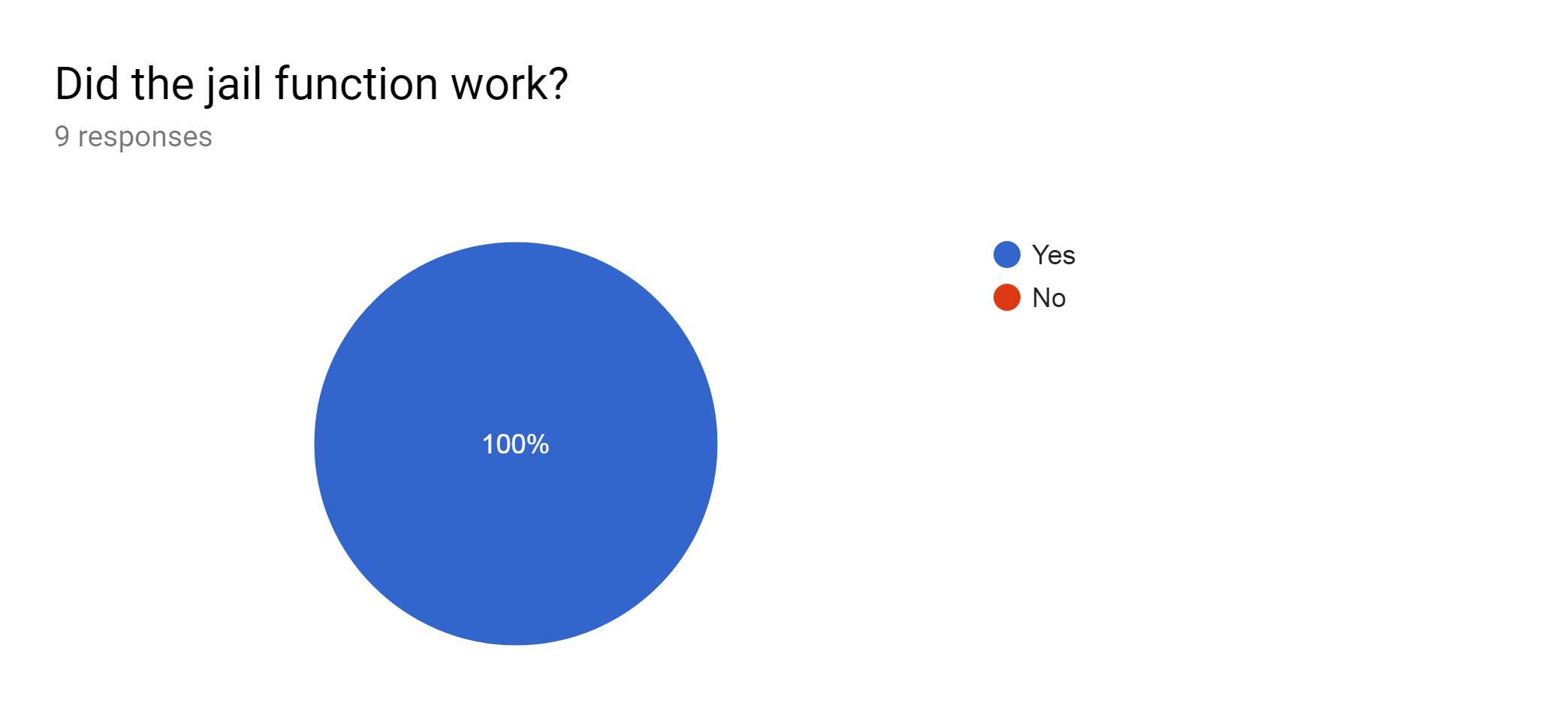
My program successfully asked if they wanted to buy the property



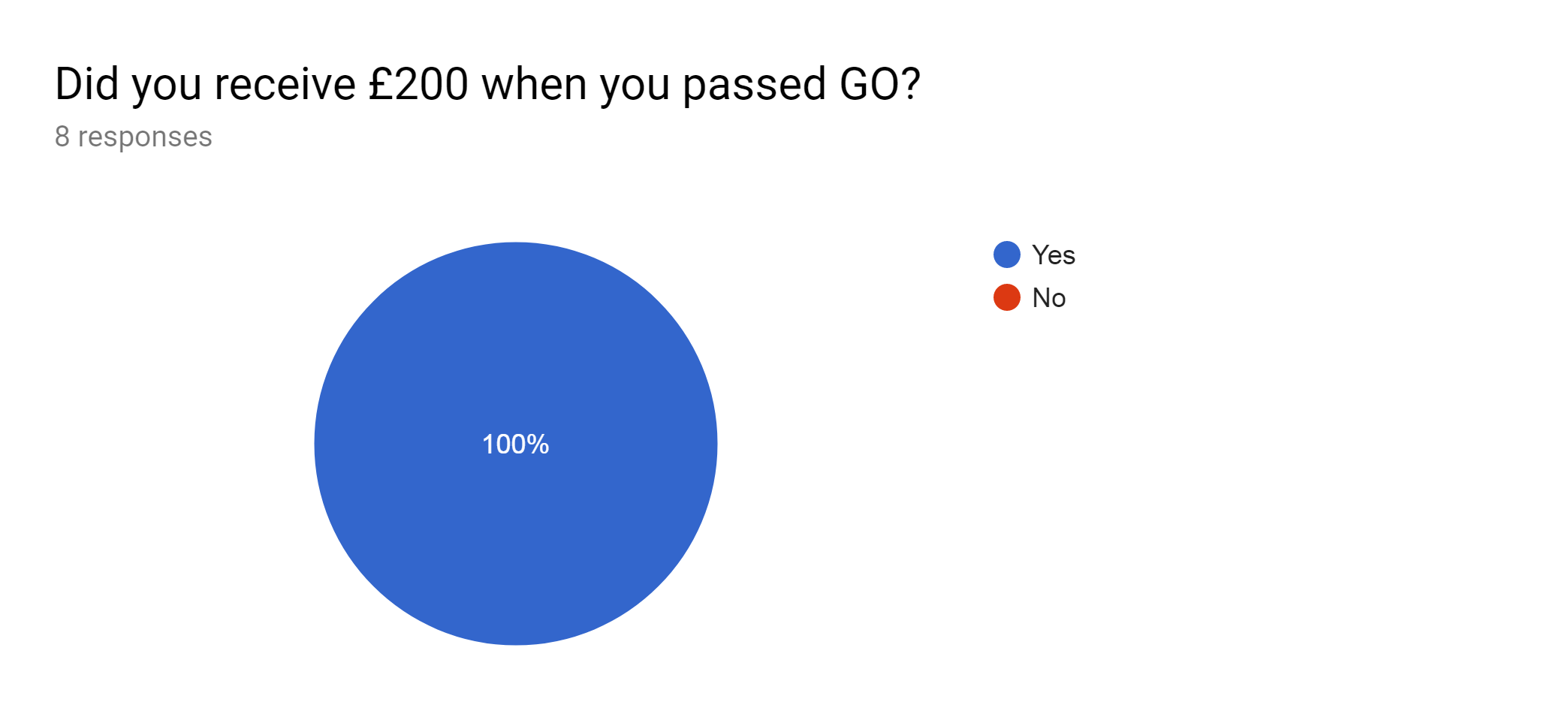
My program successfully terminated the active players turn and started the next players one



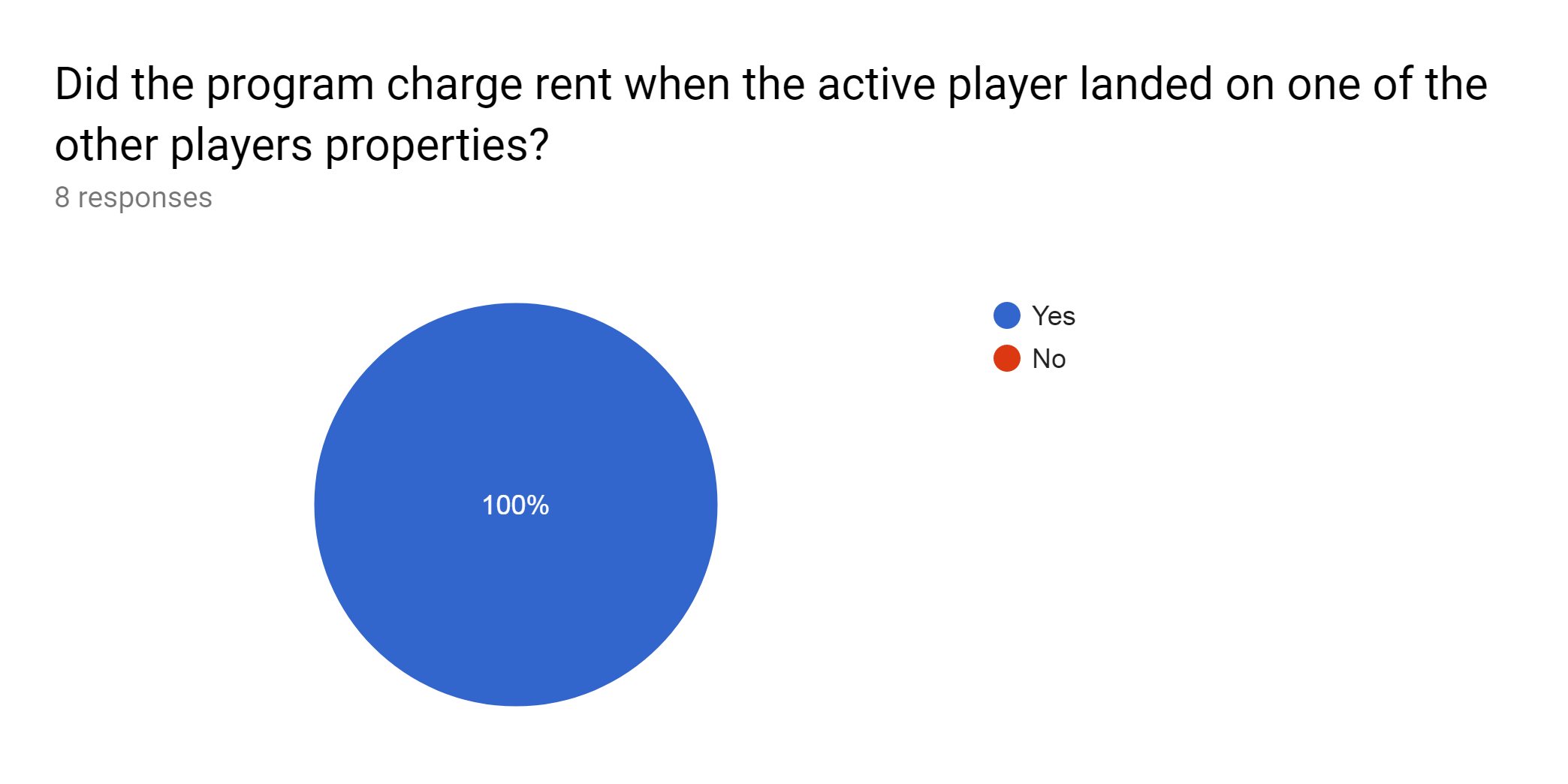
It is obvious that the chance and community cards work correctly



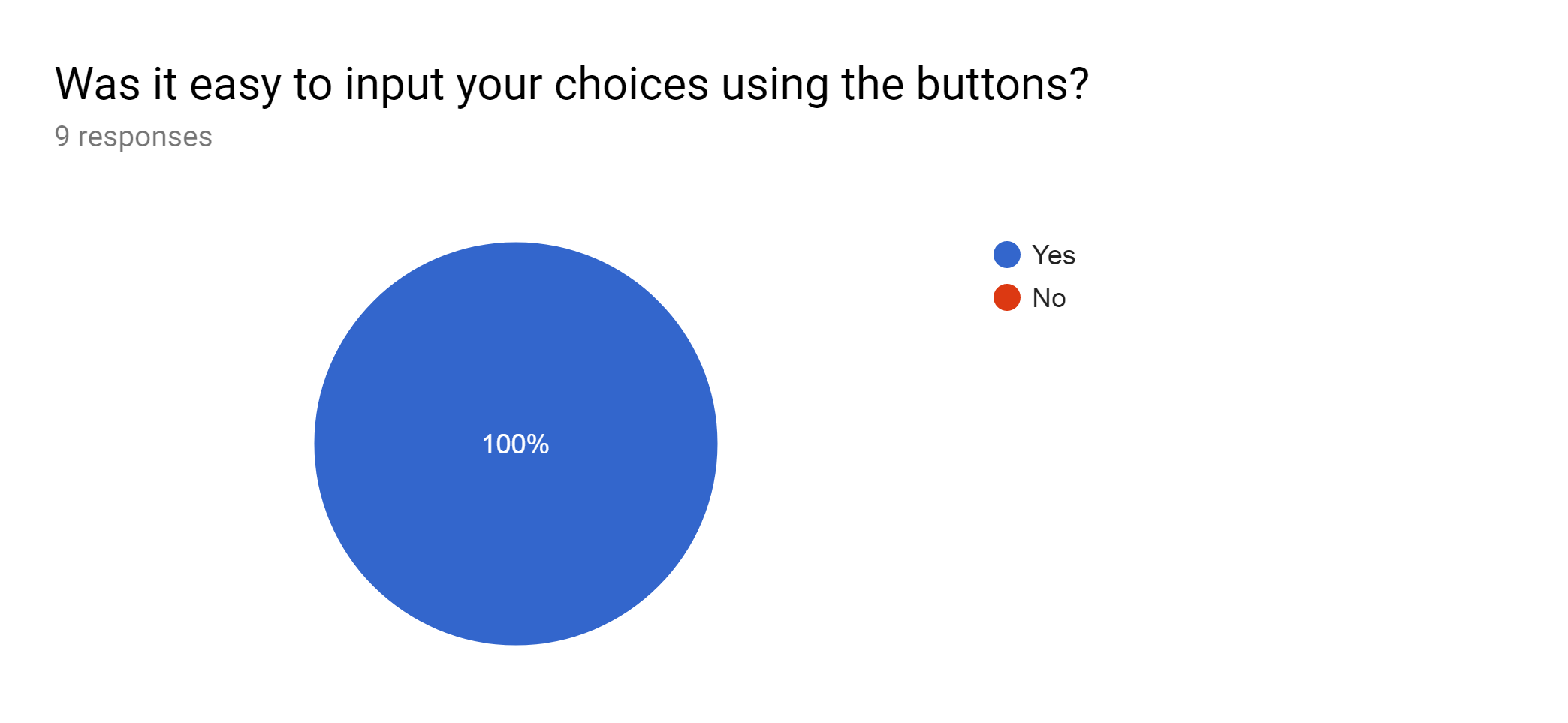
The jail system worked in my program



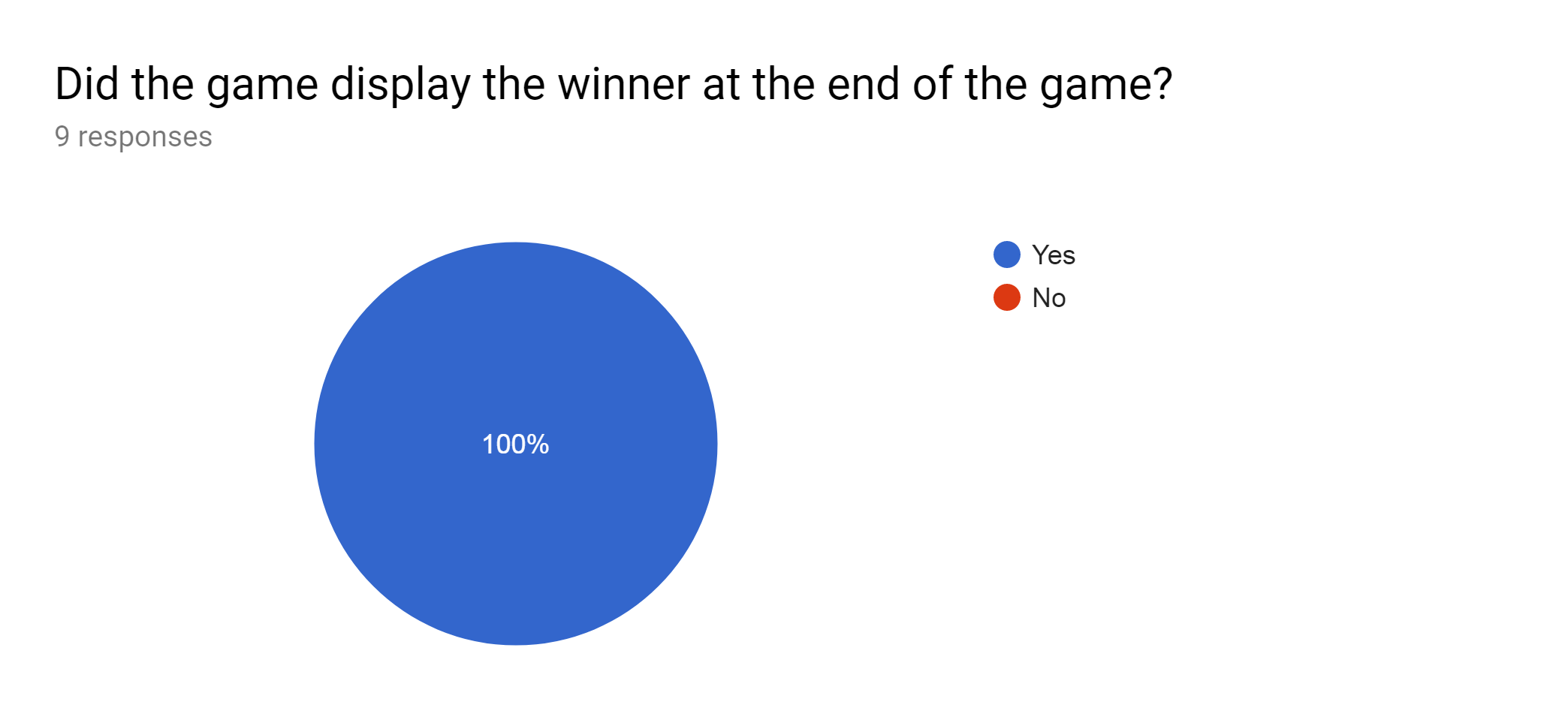
My program successfully gave the user £200 when they passed go



My program successfully charged the correct rent and gave it to the owner



I have successfully designed a user friendly interface which is easy to use

My program s

My program successfully display the winner and loser at the end of the game.

Development Process:

| Stage of Development | Point | Completed? | If not, why? |
| --- | --- | --- | --- |
| Programming | Inputting all of the necessary data from 3 separate csv files | Yes |  |
|  | Creating 2D Array, for each player in the game | Yes |  |
|  | Making method to check the position they have landed on and complete the necessary protocol. | Yes |  |
|  | Create main loop to keep game going until there is a winner | Yes |  |
|  | Create method to calculate rent which should be charged for a certain property | Yes |  |
|  | Buy houses/hotel method to check if they own street etc | Yes |  |
|  | Setting up methods for each tile they could land on i.e. jail, free parking, chance and utilities etc | Yes |  |
|  | Creating inherited class with user interface | Yes |  |
|  | Creating class in super class which creates the UI each time it is required with updated data each time | Yes |  |
|  | Making my buttons work so that the call classes in the super class, even though they buttons only exist in the inherited class | Yes |  |
|  | Testing UI to see if the methods all work together and everything updates successfully | Yes |  |

Further Developments:

I would add the function to be able to trade properties between player in exchange for money. Also for the properties to be auctioned if the current player doesn’t want to buy them. I would alter my game so that up to 8 people are able to play it.

Valid and Relevant conclusion:

After many setbacks and problems with the user interface in java, I have completed my project and I am very happy with the outcome. I believe I have created a solution to the problem I proposed at the start of my project. My program has met the functional requirements that I set out, also my end-user requirements have all been achieved, and my scope and boundaries have all been reached and accomplished.

Evaluation of development process

| Testing | Difficulties | How I overcame them |
| --- | --- | --- |
| Read in all of the data for my game | Too much data for one file to input | Used 3 separate csv files and 3 extra classes |
| Making a user interface from scratch | Too difficult just using code | Used an interface JFrame designer program |
| Implement the user interface into my program | Had to be in another class but connected to the main class | Used an extended class inside the master class and made a method to refresh the UI every time and creating a new instance of the variables each time to pass the parameters into the UI class. |
| Optimising buttons | Making the buttons disabled and enabled when they shouldn’t be able to be clicked or have already been used | Used boolean variables in the master class which would be turned to false and true when certain actions were available and not. For example after the user rolls the dice and doesn’t roll a double disable the roll dice button for the rest of their turn. Another example would be when they are on a tile they cannot buy the appropriate booleans are set to false and the “Buy Houses/Hotel” and “Buy Property” buttons are disabled. |