JavaFX Project

- Door Dash -

Nicolás Serrano

Taryn Cail

Zach Davis

CS 1083

Dr Jeff McNally

Winter 2024

**Game Description**

The game DoorDash involves five sets of doors, each door hiding an outcome behind itself. You start the game with 0 coins and as you progress through the doors you can increase or decrease your coins which will correlate to a VB’s prize.

***How to Play***

At the start of each round the player selects a door, once chosen, the door will reveal what’s behind it, the options are -1 coin, 0 coins , +1 coin, +2 coins.

Once the player makes it through all five rounds of doors, they can choose to cash out their jar of coins and end the game, or they can forfeit their earnings and attempt to win higher prizes in the Lightning Round.

During the lighting round there is a set of four doors, behind two doors is a Lose Everything option, behind another is a Keep your Current Jar, and behind the last Double’s your Coins.

***Rules***

* You can not change your answer after declaring it.
* The player must progress through all five rounds of doors.

***Payout Scheme***

| **Coin Amount** | **VB’s Prize** |
| --- | --- |
| Coin < 0, 1, 2, 3 | 0 VB’s |
| 4, 5, 6 | 500 VB’s |
| 7, 8 | 1000 VB’s |
| 9, 10 | 1500 VB’s |
| 10 < coins | 2000VB’s |

| **Lightning Round Prizes** | Lose Everything | Lose Everything | Keep your Current Pot | Double your Coins |
| --- | --- | --- | --- | --- |

***Probability Analysis***

Each round the player has a 25% chance to lose coins, a 25% chance to remain at their current amount of coins and a 50% chance to gain coins.

In the Lightning Round the player has a 50% chance to lose everything, a 25% chance to keep their current amount and a 25% chance to double their coins and win a higher prize.

***Concept/Mechanics Behind the Game***

The doors in the game will be JavaFX objects that will act like buttons, when clicked a cut scene of the door opening will be played and the outcome beneath will be revealed.

* If the player gains coins there will be a cut scene of the corresponding amount of coins dropping into the jar.
* If the player loses coins there will be a cut scene of the corresponding amount of coins coming out of the jar.
* If the player does not lose or gain coins there will be a cut scene of just the jar sitting there.

On the side of the game, there will be a few things including…

* A scoresheet indicating how many VB’s each amount of coins corresponds to. (Similar to the table in Payout Scheme)
* A tally above the jar indicating how many coins the player currently has.
* An ‘End Game’ button which will declare the player’s final VB’s prize based on their current amount of coins and end the program.
* A ‘Reset’ button which will empty the jar, reset the coin's tally to zero and reset the doors randomly generating new options behind each door.
* A ‘Reveal’ button so that once the player is done playing, will reveal the options behind the other doors and the pathway that the player took.

There will also be a background image which will serve as a backdrop for our game.

***Extra Credit***

We have no prior knowledge of a similar game to this, besides its resemblance to the Price is Right and Fall Guys, Taryn’s ten-year-old sister, Alysa, aided in the preliminary development of the idea.

**ChatGPT Experience**

Overall our team had a very pleasant experience with ChatGPT, we took the approach of getting ChatGPT to create the basic initial layout and then create small individual pieces of code that we implemented ourselves. We gave very clear and direct instructions that allowed the AI to work at maximum capacity with fewer errors. There was some disconnect when it came to designing the doors and buttons behind it because ChatGPT did not really understand what we meant by laying the door images on top of the buttons but still being able to click the buttons underneath. We had to do our own research for that and discover we could instead put the buttons on top but make them transparent.

**Challenges**

There were a few changes our team faced including:

* Having the door images and door buttons line up and be able to click the button and randomise the prizes without connecting the buttons and door images in order to ensure randomization. We accomplished this by putting both the door buttons and door images into separate arrays and then stacking the door images on top of the buttons. In order to still be able to press the buttons we had to set the images to pass the mouse event through it to the buttons underneath it.
* We struggled with having the lightning round window launch properly and take the jar image array along with it as well as function in the new window. We fixed this by making the jar image array a static variable outside of any class so that it could be accessed by both classes.
* We struggled with actually creating the jar image array and having it properly move through the different images, this was an easy fix by creating a few methods that interacted with the jar image array.
* An issue that came up temporarily was when we were trying to have the lightning round launch, we wanted there to be the sound of thunder while the new window launched, this idea was scratched because it became too difficult to implement and kept crashing the program.
* A major change we made to our game was the layout itself, we originally wanted our game to be played in the bottom left corner of the screen but opted to lay everything out in a Border Pane for its easy to use features and clean layout.
* We struggled with implementing sound into our game because the Media Player continued to crash our game. Even when sound was embedded into the video files it still didn’t work and we decided to not implement that part of the game.

**Future**

A few future developments for our game would be:

* To create a website for it, that allowed people all across the world to play.
* Create more images of doors slightly turned each time and play them on a timed interval one after another to create the visual effect of the door opening when a door button is clicked.
* Add an array of door opening sounds and randomly play one when a door is chosen, or assign a sound to a door.
* Change the layout of the game to eliminate blank space.
* Add sound to the video animations when a door is clicked.

**Conclusion**

In conclusion, creating and developing this game expanded our horizons of Java programming and allowed us to be creative and learn about the use of AI in a working environment. We learned how to organise, develop and create a fully functional game which was an amazing experience.