100Doors

The **100Doors** project is based on the logic of a door maze, conceived by Jaoiher Boulila, Mihai Branga-Peicu and Anis Slimani.
Positioned at the center, a set of doors surrounds the player, their number determined by the chosen difficulty level.
The player must guess the correct door combination, while a progress bar tracks the game's advancement.
Each door has specific characteristics and can be opened using various features.
Once the player correctly guesses the combination, they win the game.

How to play?

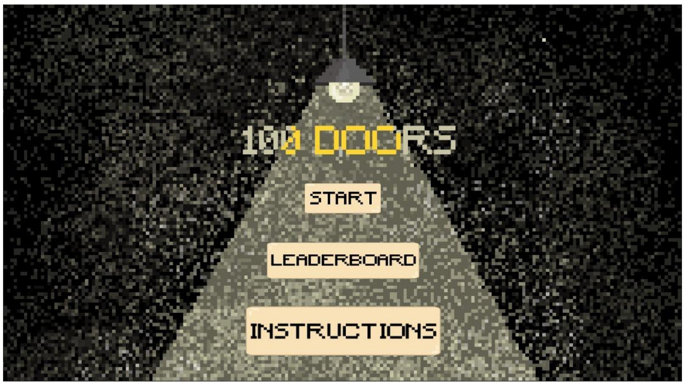
Setup and Launch:

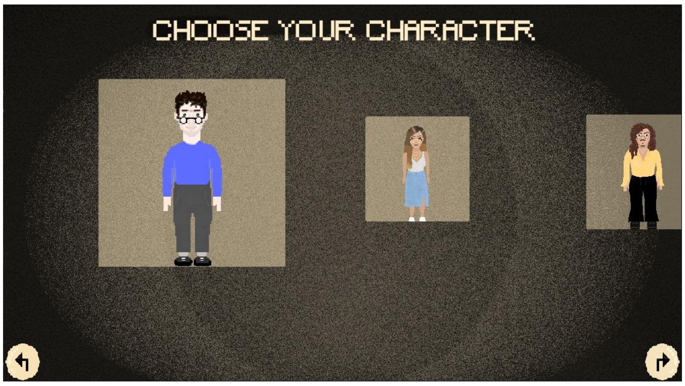
- Open IntelliJIDEA.
- Navigate to File.
- Open the OneHundredDoors.java file.
- Click the play button to launch the game.

Starting the Game:

-Click **New Game**.
-Choose your Character
-Choose your difficulty level by clicking on **EASY**, **MEDIUM**, or **HARD**.

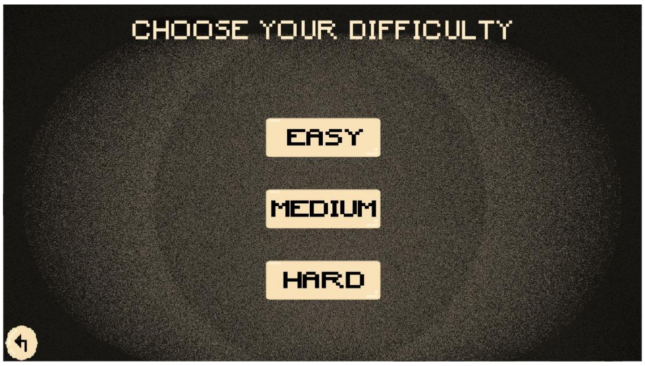
Screenshoot

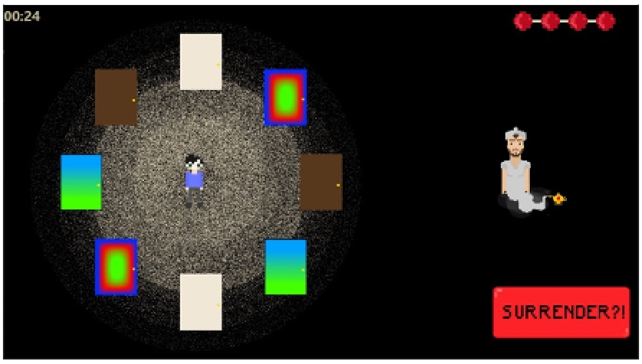




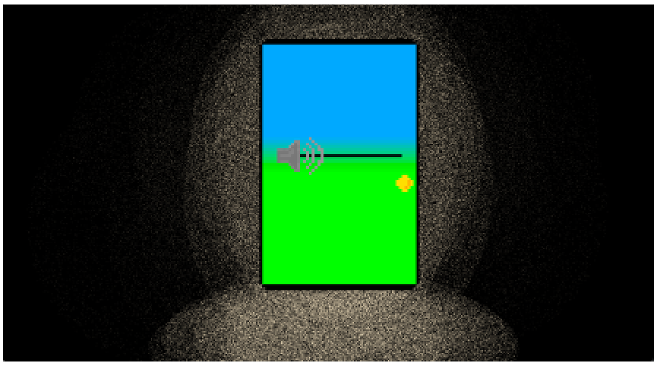
Once the game is initialized the player starts at the win screen. They can click on the leaderboard to get the previous high scores (Scoring system not implemented), and on instructions to get information about how to play.

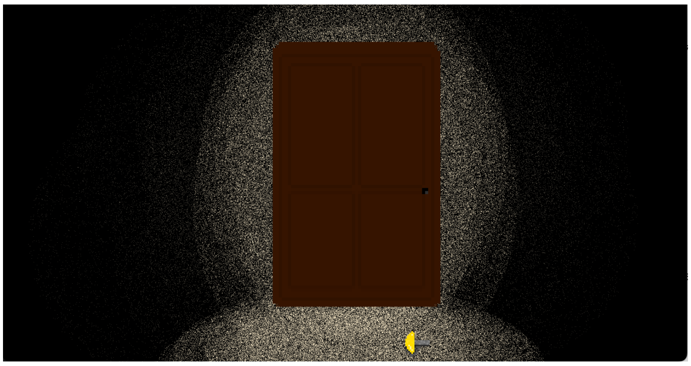
Next the player is led to the choose your character screen where they have to click and swipe to choose between Miha, Jo and Ani.



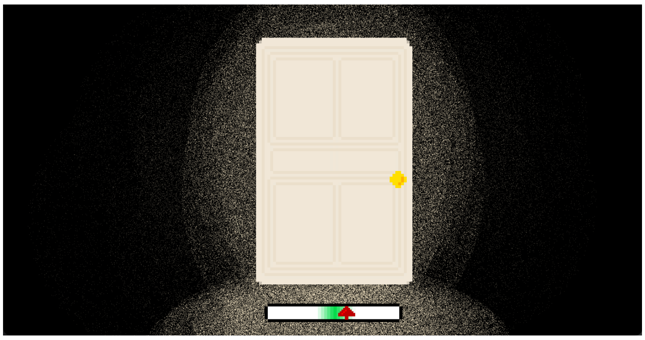


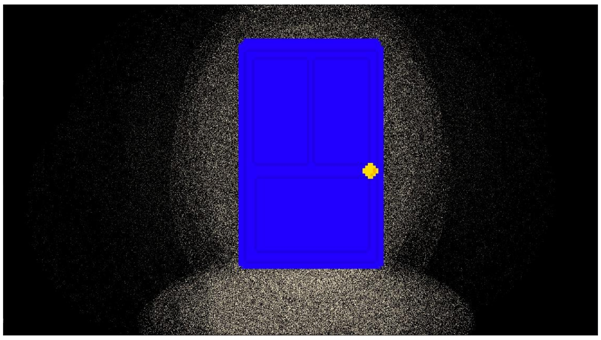
The player can choose between 3 levels: easy, medium and hard. In easy mode the player has to find the combination of 4 doors to open between 8 doors, in medium between 12 and in hard between 16. There is a progress bar that indicates how many correct doors they have opened. There is a surrender button which is the only way for the player to quit the game. If the player guesses a wrong door they get returned to this screen, with all the progress re-initialized. There is also the magical genie Mehdi, which when clicked gives the player hints. The first hint makes one unnecessary door disappear, while the second hint creates a halo effect around the door the player is supposed to open.



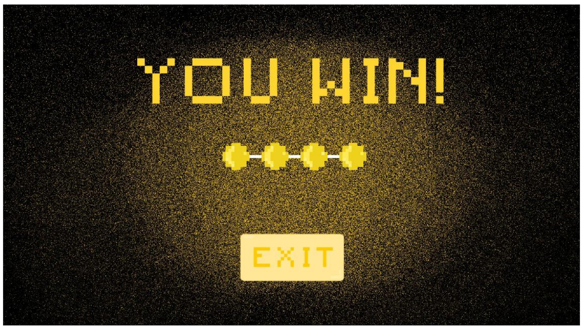


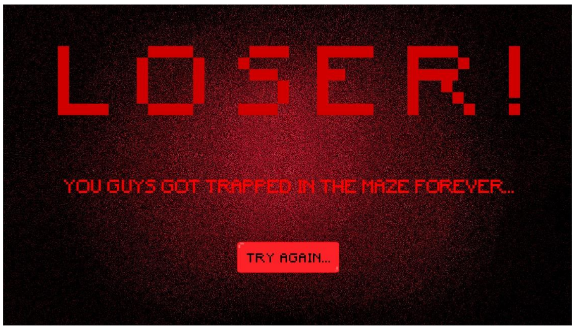
The sound door opens once the player screams loud enough, once they reach the limit the door unlocks. In the drag and drop door, the player clicks and drags the door handle to the hole in the door to open it.





The timed button, where the player clicks the Enter button to open the door, meanwhile in the color door, they have to show the camera an object that is either red, green or blue depending on the door that appears on the screen.





YOU GUYS GOT TRAPPED IN THE MAZE FOREVER...

Once the player completes the combination and the progress bar is green, the player gets led to the win screen, with an exit button that leads them to the start screen. When the player surrenders they get led to the lose screen. Surrendering is the only way to see this screen.

What was implemented?

- **Basic Mouse Interaction:** Players can navigate between scenes and ask for help.
- **Timer:** A timer is implemented.
- **Mouse Press Event:** The event occurs only when the mouse is pressed.
- **Surrender:** Users can surrender the game by pressing the surrender button.
- **Character Selection:** Users can choose their character by **swapping** between options and selecting their desired character.
- **Hints:** Users can request hints by clicking the “genius”. Two hints are available.
- **Game End Screen:** After completing the game, a screen announces victory.
- **Different Functionalities Doors:** Each door has different functionalities:
  - **DragAndDropDoor**: The user must click, drag, and drop the door handle onto the lock to open the door and progress in the game.
  - **SoundDoor**: The user must shout to open the door and continue in the game.
  - **ColourDoor**: The user must show the camera the color orange, blue, or green, depending on the door they have encountered, to open it.
  - **BarDoor**: The user must stop the arrow in the center, which moves along a bar, using keyboard input (Enter) to open the door and continue in the game.

What was coded?

- **HomeView** manages the main screen, where players can access game instructions, view statistics, and initiate a new game. The "home.css" files control the colors, backgrounds, and fonts used on this screen.

-The **CharacterSelectionScreen** class oversees the character selection screen, enabling users to switch between characters for their choice. The "characterSelection.css" files control the colors, backgrounds, and fonts used on this screen.

-The **ChooseLevelView** class handles the view where players can select the game's difficulty level. The "chooseLevel.css" files control the colors, backgrounds, and fonts used on this screen.

-**GameModel** and **GameView** are the core components of the code, governing the view and model of the entire game, encompassing the visual aspects and the underlying logic of 100Doors. The "game.css" files control the colors, backgrounds, and fonts used on this screen.

-**DoorModel** and **DoorView** manage the doors and their various types, along with their respective logic. The "door.css" files control the colors, backgrounds, and fonts used on this screen.

-**SoundRecognizer** allows for the detection of sound intensity received from the microphone. This is crucial for the SoundDoor, where sound intensity is detected and calculated.

-**WebCamCapture** manages the ColorDoor, enabling the detection of different colors through the webcam.

What needs to be improved:

- Despite having a few types of doors with interesting interactions, the possibility is endless for more types of doors and more types of interactions