université PARIS-SACLAY

Find your puzzle peace

Project work
Programming of interactive systems | Université Paris-Saclay
By: Ida Carlsson, Katja Kordes, Angeliki Fokou

General Info

This project is a group project where the main goal is to create an interactive maze with different interactions, we were encouraged to be creative when implementing the maze.

Technologies

- java version 18
- javafx-sdk-19

Setup

Our github link: https://github.com/afokou/MazeGame.git

To be able to run this project (step by step here, how to run the program)

- Open eclipse (used version **2022-09 (4.25.0)**)
- Right click on the project → properties → Java Build Path → Libraries → Module Path → Add External JARS → Add the jars in your javafx-sdk folder
- Right click on the project → Run As → Run Configurations → (x) Arguments → unclick "Use of the X-start….." → *add in VM arguments → --module-path <path to workspace>/javafx-sdk-<version>/lib --add-modules=ALL-MODULE-PATH*
- Run the program

Separation of tasks within the group

To effectively manage this project, we choose to divide the project into different tasks and subtasks. We started with making our storyboard which gave us clarity as to which tasks that were necessary to reach the end goal. We then talked about our different strengths within the group and divided the tasks equally. Even if we divided the tasks everyone was involved within every process of this project and to ensure quality we always gave each other feedback and explained our thought process. Because of that, throughout the whole project we all have worked together on all the tasks, but had different focus areas which is specified below.

Ida:
Ida created the first version of the game as a base to start implementing the different functions, the first fxml files and made the functions for transitioning through each page. Ida then worked on the interactions functions and specifically worked on the drag and drop function handle and the interaction with the dog. Ida worked with the finishing touches and made all the buttons work and error handling.

Katja:
Katja made the initial sketches of the game idea and has mainly focused on the graphics part of the prototype and final project (by designing the icons and characters), on creating the

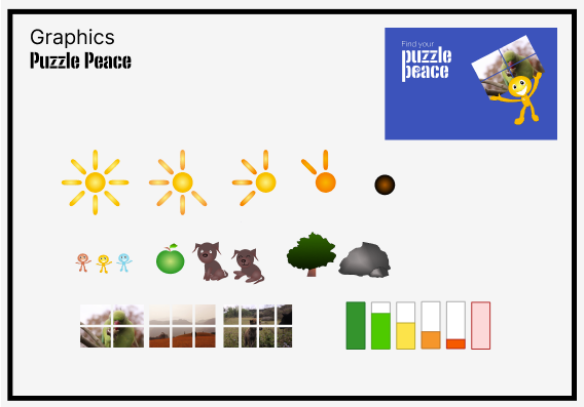
different fxml files and on adding the styling of the game. Furthermore, Katja has mainly focused on the keyboard input with detecting that the user presses a certain combinations of keys in 10 seconds time. This interaction also included allowing to restart the input. Further, she also worked with the error handling of the project.

Angeliki:
Angeliki made the second version of the game more interactive and implemented the basic functionality of the game. Firstly, she created the character movement functionality, as well as the obstacle detection. Angeliki worked on the obstacle interactions and the brightness reduction functionality.

Everyone of us has worked on every function because we had to incorporate each other functions in each other. The last final touches we all did and went through the code together and tested.

Information about the game

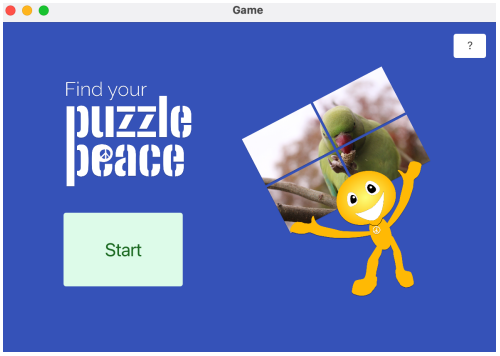
Graphics



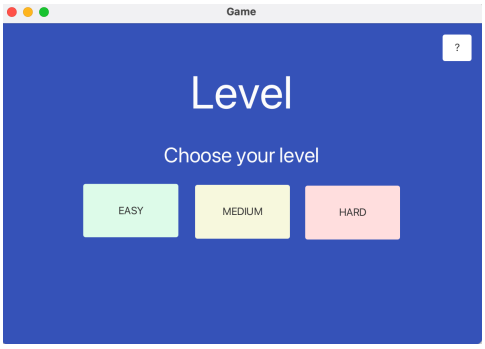
Implemented functionalities

Basic functionality

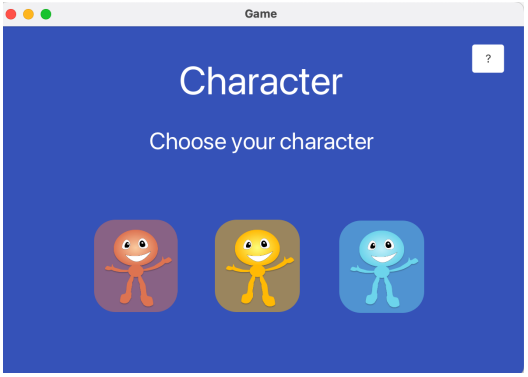
- The home screen of the game shows the ‘Start’ button.



When clicking on this, the user will be redirected to a levels page, where the user can choose one of the 3 difficulty levels, Easy, Medium or Hard.



After choosing the level of their preference, the user will be redirected to the Character screen, where they will be able to select the character of their choice.



- Depending on the user's choice of level, the gameview will differ. There will be more obstacles the harder level the user choses.

- The easy level has an initial time of 150 seconds, and for the user to win, 4 images are required to be collected. It also includes two dogs. When the user approaches the dog, they are asked to input a key combination in 10seconds, after which they can receive additional time.
- The medium level has an initial time of 100 seconds, and for the user to win, 6 images are required to be collected.
- The hard level has an initial time of 60 seconds, and for the user to win, 8 images are required to be collected.

- The user can move the character by keyboard input, using the up, down, left and right arrows.

- When the user encounters a puzzle piece or a food object, it will automatically move to the grid at the bottom of the page.

State indicator

- The "sun" 

at the bottom of the game view, shows the remaining time
    - During the game the light will start to fade and that way the user will have an indication that the time is almost over.

- The “barometer” 

at the end of the screen shows how much energy the user has left.
  The energy level reduces as the time goes down.
  To gain more energy, the user needs to **drag and drop** any collected foods from the inventory to the energy bar. This will reset the current energy level to the initial state.

Winning the game

The game is won when the user has collected all required images for each leven, within the given time

Losing the game

The game is lost if the user has not managed to collect all the images within the time given.

Advanced functions

Gain time

The user can gain extra time to continue playing in the game with two ways:

1. In all levels, at any time during the game, by dragging and dropping food that has been collected in the inventory, to the energy bar. This resets the time to the initial state.
2. In the easy level, when the user encounters a dog. In this case, the user will be given 10 seconds to memorize a letter combination and **TYPE** it on their **keyboard**, and, if done correctly, the time will reset back to the initial state.

Memorizing and inputing a keyboard combination

The user is given the opportunity to earn extra time in the easy mode. After approaching a dog on screen, the user is asked to memorize a keyboard input combination, close the window and type it out. They have ten seconds to type it, otherwise the dog will be gone without the user receiving extra time. If the user mistyped, they have the option to type “X” and restart typing the combination. However, the time limit of 10 seconds from the start remains.

Limitations:

Our devices did not allow the recording of sound in Eclipse, so we decided to focus on other interactions instead.