	Name	How to demo	Notes
1.	Project initialization.	Project compiles and runs with no errors.	Create a new java project, add main class and methods. Set up Git.
2.	Main window setup.	Displays the initial game interface layout.	Set up the primary game window, defining its size and layout
3.	Main panel creation.	Panel appears inside the window.	Add the game board.
4.	Create a static layout.	Open the layout in a browser to see the static cafe scene with customers and a waitress.	Create game backdrop of a cafe, creating the customers and waitress.
5.	Clickable tables.	Clicking a table prints its number in the console.	
6.	Generate different food items.	Screen displays a list of all available food items.	Create a menu class and add different types of menu dishes.
7.	Display random orders.	Every few seconds, a table shows a dish request from the menu. A dish icon\text appears next to the table.	Generate random orders for tables over time. Customers appear next to a table when ordering - each table has one customer.
8.	Define paths to tables.	Print correct coordinates for each table path.	One path per table.
9.	Movement by click.	Waitress moves to a table position by click.	Only one movement can be active at a time. Clicking on different tables creates a list of future movements.
10.	Select which dish to carry	Player clicks on the dish that they want to carry	Player can only select one dish at a time
11.	Dish serving correctness.	If the waiter has the right dish - order disappears and points increase.	Wrong dish = lose points.
12.	Scoring display.	Score increases/ decreases according to the dish serving.	Add a score label and update it dynamically.
13.	Customer timer.	User loses points when he does not serve customers in time.	Timer for when a dish is not served on time.
14.	Star system setup.	Each time a player reaches a specified score, they earn one star.	Total attainable stars are five.
15.	Mini game popup.	Display a popup whenever a new star is earned.	Popup appears between stars.

16.	Create a memory game.	Game pops up whenever it is triggered and the player has to play a memory game of remembering a sequence of menu items.	Remembering the sequence awards the player with a new menu item.
17.	Return to main game.	After the mini game ends, return to the main window and resume.	Maintain score and state of the game.
18.	Winning condition.	After each mini game, check if the user reached 5 stars. If so - the user wins the game. Else - continue the game.	
19.	UI improvements.	Game improves visually.	Replace shapes with icons/ images + add sound effects.

Main topics

Git - Version control – The project was fully developed and managed using Git. Each of us had her own branch – to develop different parts of the game. After testing new feature locally, changes were merged into the main branch through a pull request. This allowed us to collaborate efficiently without overwriting each other's work. A detailed README file on GitHub also explains how to set up and play the game.

Git repository -> https://github.com/ShiraYos/CaffeGame.git

UX – We applied UX features that made the game clear, engaging and enjoyable. Players receive immediate visual and audio feedback for their actions – for example, when serving a customer, earning points, or progressing through levels. The layout is minimal and intuitive, allowing players to understand gameplay naturally without long instructions. In addition, every time the player wins a mini game – he gets a new item to the menu, maintaining motivation while playing.