Backlog for snake

Topics of learning:

-Usage of Git and GitHub

-Usage of sprite sheets, image cutting, tinting and rendering

1 Main menu (settings)

-Demo: Launch the game → menu with “Start Game” option.

-Notes: No working settings yet.

-Learning goal: Learn to design a basic menu and handle user input.

2. Render Playfield/Grid

-Demo: Starting the game shows a defined rectangular play area or grid.

-Notes: Making use of sprite sheets and an image cutter, we can create tiles from the sheet. Then we can order the tiles in a random but regulated manner to create a grid (different for each playthrough). Then by image tinting we can render a checkerboard-like pattern with edges.

-Learning goal: Learn to use sprite sheets: file handling, image cutting, image tinting and image rendering.

3. Snake Head Rendering

-Demo: When the game starts, a snake head is visible at a default position.

-Notes: No movement yet. (Behind the scenes, the entity stores each of its body part’s position.)

-Learning goal: Learn to use java’s built in painter to render a single game entity and position it visually on the playfield.

4. Snake Movement (tick)

-Demo: On game start, the snake head moves forward automatically each tick/frame.

-Notes: Uses a fixed direction initially.

-Learning goal: Learn to implement tick-based movement.

5. Direction Control (Arrow / AWSD Input)

-Demo: Pressing arrow keys changes the movement direction.

-Notes: Prevent instant 180-degree turns.

-Learning goal: Learn to handle real-time keyboard input and update movement direction, especial emphasis on one input per tick.

6. Eating Food & Growing

-Demo: Trigger growth by eating an apple and show body segments follow head.

-Notes: New body positions stored and rendered, not interfering with previous segments.

-Learning goal: Learn to handle data on the grid.

7. Food Spawning

-Demo: Random food item appears on the playfield when the game starts.

-Notes: Can't spawn on the snake.

-Learning goal: Learn to generate random positions while avoiding collisions with existing objects.

8. Wall Collision → Game Over

-Demo: Drive the snake into the boundary → game stops and game-over state is shown.

-Notes: Stops the game.

-Learning goal: Learn to detect boundary collisions and transition to a game-over state.

9. Self-Collision → Game Over

-Demo: Turn the snake onto itself → game ends.

-Notes: Same game-over handling.

-Learning goal: Learn to detect collisions between dynamic objects.

11. Score Display

-Demo: Eating food increments score, shown visibly in UI (top corner or overlay).

-Notes: When you eat an apple, your score increases by 1. When you die, your score is displayed.

-Learning goal: Learn to render a HUD and bind game state (score) to real-time visual output.

12. Settings

-Demo: Start the game with different settings, see if it’s fine.

-Notes: The settings are: Size of grid, speed of snake, fruit variants, colour of snake.

-Learning goal: Learn to implement configurable gameplay parameters and apply them.