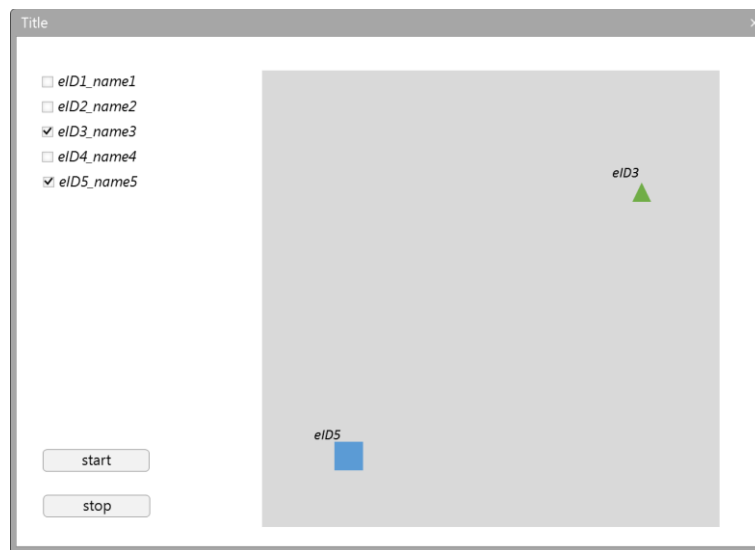# Developer candidate coding test

Coding Task

Part I

Using the attached JSON file, create a user interfaces displaying the entities in the file based on their properties (color, type, size) in their initial coordinates (X, Y).

On startup, all entities will be displayed and the user will have the option to hide / show entities from the display.

Example user interface, two entities shown.

Part II

When the start button is pressed, each of the selected entities will move (within the display grid) to a distance of 5 from current location at 5 sec intervals. The direction will be random (up, down, left, right). The entity cannot go back to previous location on next step.

Example:

Time = 0, eID1(X, Y) = (10, 10) >> Time = 5sec, eID1(X, Y) = (15, 10)

At time=10sec, location (10, 10) for eID1 is forbidden.

The movement of the entities will continue until stop button is pressed by user.

The n last locations of the displayed entities will be saved to a csv file, including the entity ID, name, X and Y. Value of *n* will be input by user.

Output example (*n*=2):

eID1, name1, 15, 10

eID1, name1, 10, 10

Notes
- The display grid is 100x100 (i.e. 0<=X<=100; 0<=Y<=100)
- There are three supported colors (red, green, blue)
- There are three supported entity types (circle, square, triangle)
- There are three supported entity sizes (small, medium, large)
- Max number of supported entities is 10
- JSON input file will be located at executable folder (testing input may vary).

Open Questions
(No code, written answers only)

Considering the coding task, how would you support the following requirements?

1. Display a line of movement for each moving entity, which shows the movement for the last *n* locations of the entity (user inputs value of *n*).
2. Support having three coordinates for each entity (i.e. (X, Y, Z) and how the display would be reflect this.

General Instructions:
1. The code must be written in JAVA.
2. Upload all source code & executable to a GitHub repository.
3. Upload document (txt, doc, etc.) with answers to open questions to GitHub repository.
4. Make sure code is readable / understandable.
5. If required, add document describing what you did (not a must).
6. Any assumptions made, beyond those stated above, must be described.