Crowd dynamics simulation - a multi-agent system based on CA

Preparations

- Download and extract Pedestrians.zip (http://home.agh.edu.pl/~porzycki/Pedestrians.zip) .
- Import and run project

Cell types in CA

- 0 - floor
- 1 - wall
- 2 - exit
- 3 - pedestrian

Neighborhood

In class **Board** in method **initialize()** initialize neighbors for every cells. Prepare two versions using:

- Moore neighborhood,
- von Neuman neighborhood.

Do not initialize neighbors for the border cells.

Static potential field (2pkt)

In class **Board** in method **calculateField()**:

- Create list of **Points** for which static field should be recalculated. Please note that initially the value of **staticField** is set to 100000.

```
ArrayList<Point> toCheck = new ArrayList<Point>();
```

- For each cell of type 2 (exit), set its **staticField** to 0. Each neighbor of such cell should be added to **toCheck** list.
- Until list **toCheck** is empty:

1. verify if first element on **toCheck** list changes its **staticField** (use method **calcStaticField()** for this cell)
2. if it is true, add all neighbors of this cell to **toCheck** list.
3. remove first element from the list.

In class **Point**:

- Implement method **calcStaticField()**. If this cell **staticField** is larger than smallest value of naighbours **staticField** +1, set cell static field to this value. Return **true** if you change the value of **staticField**, otherwise return **false**.

Run your application. Set some exit. Push button „Calc Field”. Analyse the influence of used neighborhood to the shape of static floor field.

Crowd dynamics - naive implementation (2pkt)

In class **Point** in method **move()**:

- Check if there is a pedestrian in given cell:

```
isPedestrian == true
```

- If so, move the pedestrian to the not occupied, neighbor cell with smallest static floor field.

Run simulation. Observe what artifacts appears. Try to find reasons of this errors.

Crowd dynamics - improvements (2pkt)

Main reasons behind errors in previous point are:

- Agents that reach exit should be removed from the simulation. If pedestrian enters cell type 2 (exit) do not set its variable **isPedestrian** on true.
- No synchronization of cells. One should note, that agents moving down, and right can reach destination in one iteration. In order to fix this issue, create in class **Point** variable

```
boolean blocked = false;
```

If cell is blocked, agent on this cell can't make a move. Cell is blocked if some pedestrian enters it. Remember to unblock all cells at the beginning of each iteration.

Further improvements

- Improve calculation method for static floor field.
- Add repulsion force between pedestrians and walls (add it by modification of static floor field close to the walls).
- Random order of pedestrians movement.
- … your idea.