

Assignment Specification 3

Contains:

Part 1:

Section 1 – Design Structure

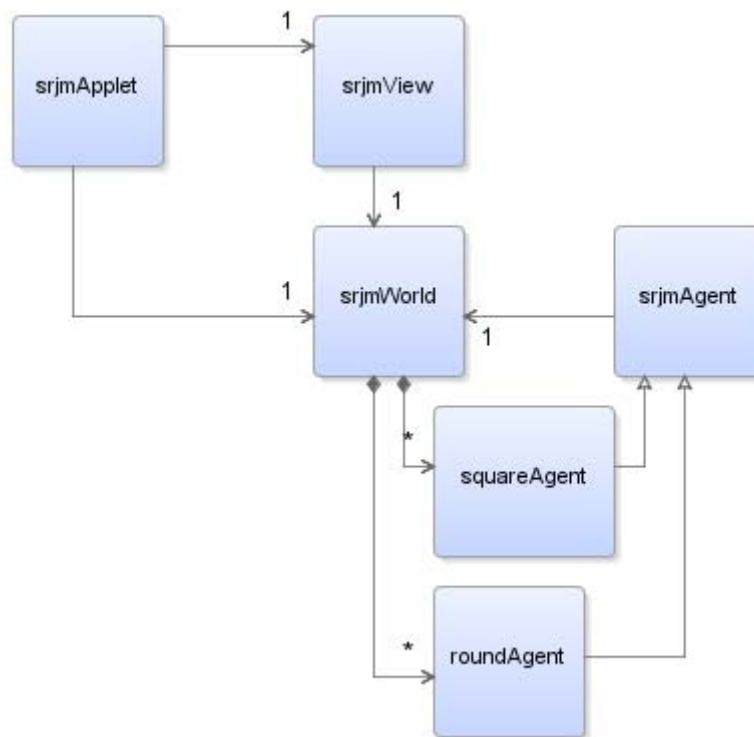
Section 2 – Class Descriptions

Section 3 – Agent Rule Descriptions

Part 2:

Printed Java Classes

Section 1



srjmApplet is the main java class that initialises and displays the program by arranging the interface, providing buttons and controls for the user to modify the setup, and by creating and displaying the srjmView and srjmWorld classes.

srjmView is a class that extends panel and can therefore be added as a panel directly to the interface in srjmApplet. Its purpose is to take numerical positions of cells and agents from srjmWorld and draw them appropriately as a coloured grid, so the two classes are closely linked.

srjmWorld, the central class to this program, creates any number of squareAgents and roundAgents and an array-like structure of cells that can be of two states (on or off). It provides srjmView with the information to draw and on a timer event from srjmApplet, cycles through each agent triggering their 'move' function.

srjmAgent has two subclasses, squareAgent and roundAgent that follow different rules and are created directly by srjmWorld rather than the main srjmAgent class itself.

Section 2

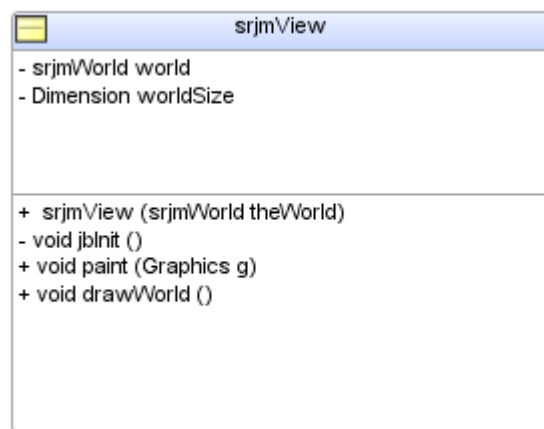
srjmApplet

Arranges the graphical user interface of the applet and uses the values from the four textfields to initialise a new world via the generateWorld() function when the start button is clicked by the user. Using a border layout the controls of the applet are arranged from top to bottom with srjmView as a panel in the centre, input values at the top and the buttons and speed scrollbar at the bottom.



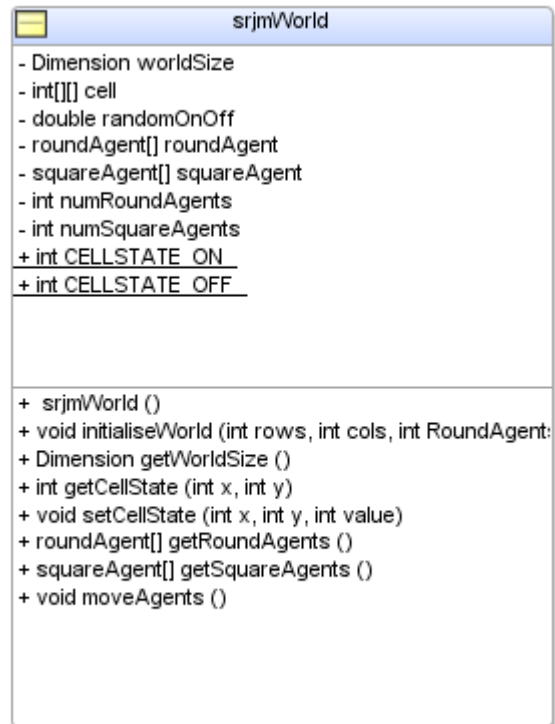
srjmView

This class takes information from srjmWorld and draws it as the visible map on the central panel of the applet. To achieve this firstly it uses a nested For loop in order to draw individual cells in srjmWorld, using a different colour for the two states, On and Off. Then secondly it cycles through each agent created by srjmWorld and draws a differently coloured and shaped rectangle in the corresponding position.



srjmWorld

Using initial values passed from the applet to initialiseWorld(), srjmWorld creates and manages the world's activity. It holds each cell of the world in a 2D array (cell[x][y]) and has two separate arrays holding either type of agent. Values are passed and set by other classes via the set and get functions.

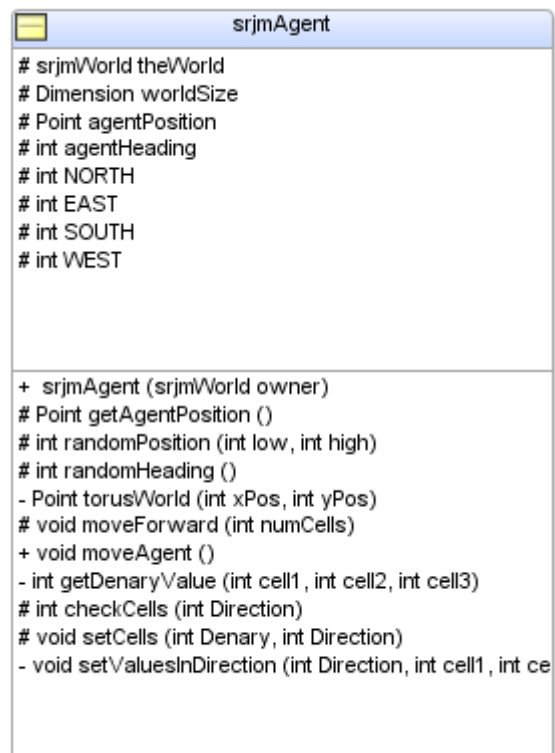


srjmAgent

This class is the basis for both of the subclasses Round and Square agents. Following each update of the world it will change position and possibly the state of nearby cells according to a simple set of rules, defined more comprehensively in the two subclasses.

The torusWorld is a special function that has had to be made to compensate for more than one task, direction and distance due to the flexibility in the distance an agent can be made to travel and check cell states. It keeps an x,y coordinate within the bounds of the world's size for getting and setting cell states and keeping the agent's position true to a torus shaped world.

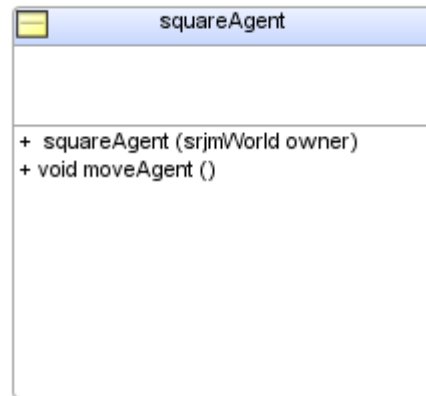
The getDenaryValue function is an incredibly simple method of converting the three cell states into a binary to denary number for which I am sure there are far more effective methods. This therefore applies to the other functions I have based on this one.



Section 3

Square Agent Rules

The square agent checks the three cells in each direction and the first it finds to be a configuration of 0 it changes to 7. It has a larger possibility for a random direction change.



Round Agent Rules

The round agent checks North and South for a configuration of 3 and sets the cells to 4 and East and West for a configuration of 4 and sets to 3. Again, it changes the first configuration it can find, and has a normal possibility for a random direction change.

