Class

CAStatic

extends JFrame implements Runnable, ActionListener. This class contains main. It coordinates all simulation and windowing events.

Fields

experiment *type: CAGridStatic*
the current grid of cells and its activities. This is a 1D experiment

savedvals *type: int[][][]*
an array to save values from the time steps of the 1D experiment.

runCount *type: int*
current experiment number

epsCount *type: int*
number of eps files that have been written

newframe *type: int*
blunt instrument for slowing the display

rand *type: Random*
random class instance

runner *type: volatile Thread*
the simulation thread

backImg1 *type: Image*
the visualisation of the experiment

backGr1 *type: Graphics*
some Java thing that takes the image and makes it part of the display or something

CApicture *type: CAImagePanel*
class used to display the changing image amidst the buttons

type: JButton
startBtn,writeBtn,paramsBtn,wrapBtn
some self explanatory buttons

msgBtn *type: JTextArea*
an area to hold text. Used to display the current parameters.

buttonHolder *type: JPanel*
a panel or window that holds the four buttons in a grid

scale *type: int*
scale factor both x and y for displaying results

iterations *type: int*
holds the current time step

gSize *type: int*
physical grid size both x and y

maxCellType *type: int*
number of different cell types. 0 is a non-cell, i.e. a space

maxit *type: int*
max number of iterations for a run

started *type: boolean*
has a run started?

palette *type: Colour*
palette of colours so that eps and display colours match

colorindices *type: int[]*

indices of the chosen colours

nnw *type: int*
used for colour repitition

javaColours *type: Color[]*
the colours in Java format

epsColours *type: double[][]*
the colours in eps rgb format

Constructors

CAStatic(int size)
sets up grid size and window size. initialises windows, buttons, and other variables.

Methods

setpalette() *Returns void*
sets up the Java and eps colours using the colour indices

saveCA() *Returns void*
save values from the current time step

outputEPS() *Returns void*
make a unique eps filename and call the eps printer

changeParameters() *Returns void*
changes experiment parameters

changeWrap() *Returns void*
toggle the toroidal wrapping

drawCA() *Returns void*
update the graphics image and flag a redraw

start() *Returns void*
start the thread and update the button status

stop() *Returns void*
stop the thread and update the button status

actionPerformed(ActionEvent e) *Returns void*
listens for button or other window activity and processes the requests

run() *Returns void*
run the experiment

postscriptPrint(String fileName) *Returns void*
pretty print the results as an eps file

initilise() *Returns void*
set up a fresh experiment

main(String args[]) *Returns void*
just kicks things off. can be given a fraction as an argument which will not be used in this program