\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

+++++++++++++++++++++++++\_The Amazing Maze\_+++++++++++++++++++++++++++++

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

...could build a simple animation as game intro (for fun)...

Imports to add:

java.awt.ArrayList //solution points and tiles

java.awt.BorderFactory //optional highlight functionality - for .createLineBorder()

(https://docs.oracle.com/javase/7/docs/api/javax/swing/BorderFactory.html)

java.awt.MouseEvent //mouseListener

java.awt.MouseAdapter

(https://docs.oracle.com/javase/7/docs/api/java/awt/event/MouseAdapter.html)

java.awt.Collections //for shuffle()

initial state consists of:

.

.

.

container

JPanel

high customizability

tile side bars

central 4x4 grid

menu (collapsable...)

Set Up game

drawMaze()

save as jpeg/png

...could do resizing here...

(...get local screen dimensions, apply a ratio....)

store array of solution points

could add this to ImageSplitter functionality

run through ImageSplitter()

store array of tiles

Collections.shuffle(Tiles)

Set up puzzle

loop through i > numTiles

tile = get Tiles[i]

pic = get Pic[i]

tile.add(pic)

tile.setBorder()

tile.addActionListener(new ClickAction())//for mouseListener

sidePanel.add(tile)

mouselistener

//tile border change on mouse hover/migrate

@Override

mouseEntered()

setBorder(BorderFactory.createLineBorder(Color.yellow)) //highlight on hover

@Override

mouseExited()

setBorder(BorderFactory.createLineBorder(Color.gray)) //return to gray - if gray is original border

maybe we could override here to achieve "snap"

something like:

@Override

mouseReleased()

if outside central grid

return to last location

else

evaluate which cell in the grid is closest to center of the tile

(needs alot more thought)