Competitive Analysis: Image Maze Generator

Kai Wen Wang (Andrew ID: kaiwenw1)

Competitor 1:

<http://article.sapub.org/pdf/10.5923.j.computer.20130303.04.pdf>

Features:

* Displays image after player solves it → supports my purpose of exposing user to the appreciation of the nuances in the image.
  + I think displaying my image after solving would also be revealing.
* Takes the human way of picture-shape understanding
* Uses graph theory (and strategic placement of nodes/edges/walls) to generate and solve maze → a method I contemplated on using for maze generation.
* Connect contours (specifically cycles) of binary pictures to form maze.
  + Uses this to solve, and hence generate, the maze in an easy way.

Anti-Features:

* User doesn’t really see any inkling of the input image before they solve it.
* The maze is purely rectangular.
* The maze seems like any old regular maze at first glance. Only after solving it do we realize there is something hidden
  + This is interesting but isn’t particularly what I’m aiming for.
* Too complex a maze, won’t work well with my maze solver algorithm.

Competitor 2:

<http://www.hernan.moraldo.com.ar/mazegeneration.htm>

Features

* The maze is generated with a simple maze generation algorithm: DFS
  + However, the graph it is constructed on is being altered to look like the image.
* Thus, this provides for faster image processing, perhaps even <1s.
* Mazes of varying channel diameters to provide for an interesting user experience

Anti-Features

* You need to look far away, or you need to make the maze very complex, in order for the maze shown to be obviously derived from a image.
* Only one image example (penguin) was shown, therefore lacks a comprehensive example base for me to look at.

Competitor 3:

<http://www.cse.cuhk.edu.hk/~ttwong/papers/maze/maze.html>

Features:

* Creates maze from image outline
* Incorporates main image features into mazes.
* Very subtle incorporation so that the image appears ordinary yet there is an underlying image which inspired it, especially when looked at from afar.
* Uses novel methods, including mathematical reaction-diffusion method, in addition with cellular neural networks and evolutionary machine learning framework, to generate the maze.
  + This doesn’t require meticulous tweaking of several parameters by the user, it’s all self-learned by the machine.
* Two-Step Process: Finds stripes in the image (edges essentially), and makes graphs from them.
  + I will likely adopt the graph creation method.
* Places maze on the picture so it’s clear that they’re related. Contrasts with first competitor’s work.
* Uses loop removal algorithms.

Anti-Features:

* Very complicated algorithms and mathematics involved → graduate level work, and thus not suitable for the TP.
* Probably takes a long time to code up and train.
* Uniform maze diameter, which is good visually but detracts from possible game play (I’m talking about shrinking the player or ghosts when diameter shrinks)