Project for my favorite class

Me Myself

May 7, 2014

Abstract

My project for CG spring 2014 is the following: a square robot moves around in a 2D environment. The goal is to find a path from a start position to an end position that does not intersect the obstacles. The path is computed online, using the following heuristic: the space is divided using a regular grid and the algorithm A* is used to plan a path through the grid cells that goes from start to end. The abstract should briefly present the problem, and the key points.

1 Overview

Describe the project in more detail than in the abstract. Describe its architecture (its main components). For each component, describe briefly how it works. For example you can say that it uses the naive algorithm of checking every point with every point.

Here is the place to describe what improvements I did to the naive approach. Don't be shy. Whatever the project is, it has to be above 210 level. Balls that move on the screen and use naive intersection algorithms are not that interesting (beyond Lab 2 in Java). *Make your environment complex enough so that you need to consider and improve the speed of your algorithm.*

If you want to inlude a picture, here is how you can do it, I am showing examples of triangulations in Figure 1 and kd-trees in Figure 1.

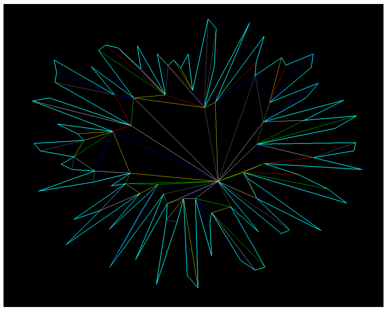

Figure 1: This is a triangulation.

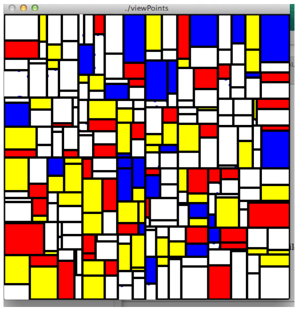

Figure 2: Cool, huh?

2 How to run it

To run the project, use

```
$plan  <filename>
$
```

3 Experimental evaluation

In this section write about the speed observed in practice. What are the bottlenecks in the project? Try to quantify somehow the complexity of the environment up to which the computation is fast enough. When does speed become a bottleneck?

If I did something to improve speed, here is the place to comment on the gains. For example, if I went through successive phases of A* with different heuristics, comment on how many states it expands from s to t in each case.

If I implemented some type of pre-processing to speed up intersection testing, here is the place to talk about the impact in practice.

Use actual numbers. Reflect.

4 Discussion

Here I would write lessons learnt from the project; perhaps things that I tried and did not work; things that made a big difference in speed. Ideas that opened up. Conclusions that I can draw, for e.g. regarding algorithms and practice and complexity and bottlenecks. If I had more time to work on the project, what would I do next.