Team 4 - Visibility Checks

Wang, Mao, Sebastian G.

November 17, 2015

1 Problem Description

Given

- A graph $G = (V, E)$ describing a landscape with obstacles modeled through simple polygons with n vertices in total
- A straight-line drawing of the landscape $\Gamma : V \to \mathbb{R}^2$
- The position $P \in \mathbb{R}^2$ (*Pacman*)
- The positions $(Q_i \in \mathbb{R}^2)_{i \in \{1,...,m\}}$ of m ghosts
- The visibility radius r

Find an efficient way to determine if P is visible from Q_i for each $i \in \{1, ..., m\}$, using "reasonable" preprocessing time (e.g. about a minute on typical inputs).

For positions $A, B \in \mathbb{R}^2$, A is *visible* from B iff $|\overline{AB}| \leq r$ and the segment $\overline{AB}$ has no intersection with any polygon in Γ.

2 Suggested Approach

We took inspiration in ray tracing and settled for a solution involving a spatial data structure like a BVH tree, quadtree or kd-tree. This is to bring down the asymptotic complexity of visibility tests to $\mathcal{O}(m \log n)$, but comes at the cost of having to build such that data structure at preprocessing time, which will happen in $\Omega(n \log n)$.

The outline of the algorithm is as follows:

1. Triangulate the input polygons (or split them in any other kind of convex polygon) for efficient intersection tests
2. Build the spatial data structure on this (convex) polygon soup
3. Perform the m visibility checks through intersection tests with the data structure in $\mathcal{O}(\log n)$ each