

# Outline

Monday, February 4, 2019

11:40 PM

## 1. Game levels

- a. Diagrams in draw.io
- b. Improvement based on that?
- c. Considering doing 3D geometry

## 2. Algorithms

### a. The main idea:

- Discretely: Try to separate the canvas into countless squares (so that we can apply discrete algorithms as we listed). For each point, find its neighbours with a certain size of step and decide its next move.
  - An appropriate amount of neighbours to do?
- Sounds a bit like hill climbing problem, calculating the distance from the destination can be seen as a function in hill-climbing problem
  - Can take a considerable amount a random points to improve the result
- Local beam search?
  - Challenge for the stack memory

### b. The visibility issue

- Try to make the algorithm itself visible
  - Shining dots and lines while running?

### c. Find shortest path

- Dijkstra algorithm (discrete)
- A\* algorithm (discrete)
- TPS (discrete)
- RRT
  - Harder to deal with when we get to the "make human-like" part since it is very hard to find the points-distribubtion pattern --> hard to apply shape-formulas to it
  - You never know the pattern
  - If we are doing this, then we need a more general make-human-like idea
- (Even consider doing the find-path part **continuously?**)
- Cut-edge point connecting
  - More random shapes involved (from level 5 in our design)
  - Couvax or the other?
  - How we are gonna detect the shape by lines?

## 3. Make the path human-like

- a. Firstly we need to solve the visibility issue
- b. Take a couple steps back and make the zig-zag path smoothly
  - i. For a 90 degree angle, re-distribute the point in a more smooth way
  - ii. Try to not walk 100% along the edge of a shape, instead, find the middle-ish points/path between two edges/shapes

