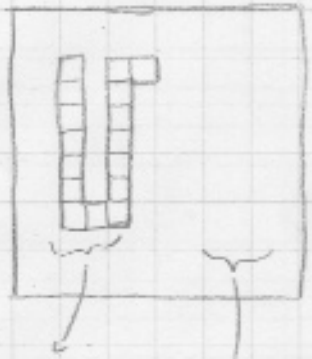
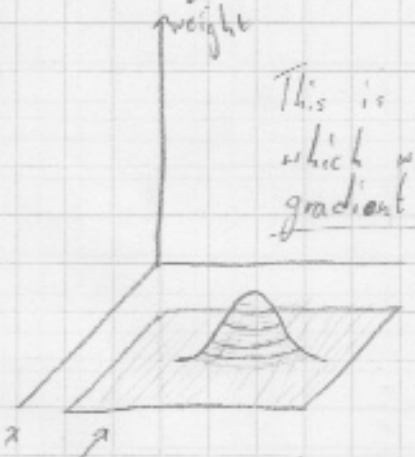
So why am I doing this? The biggest problem w/ the original snake I made is that if it cannot find a path, then it's just told to go forward. This is bad because it just crashes into the wall.

So here's my idea...



more weight in this area as opposed to this

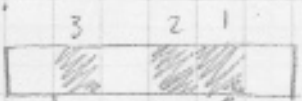
if the snake has a weight where, where the snake is has more weight, then we can just tell the snake to go to places w/ the least weight.



This is my original thought, which why did research about gradient descent.

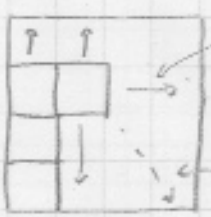
Currently (1/8/19), I have this, kinda. The thing about this graph is that it peaks at a single point, meaning it has a single point where the weight is the greatest. This means the snake points need to add together if they are close together. If you look through my psuedocode, you see I ignore snake points, instead of recursively adding them. Using the graph as an explanation, this would mean that the function would plateau instead of peaking.

i.e.



These snake points are close together, so should be added.

Furthermore, the gradients only happen in the x & y direction instead of radially. This will still work, but not good enough.



gradients only go this way but not this way

So, the algorithm needs to be reworked to account for these two things, but for now works as a version 1.0.