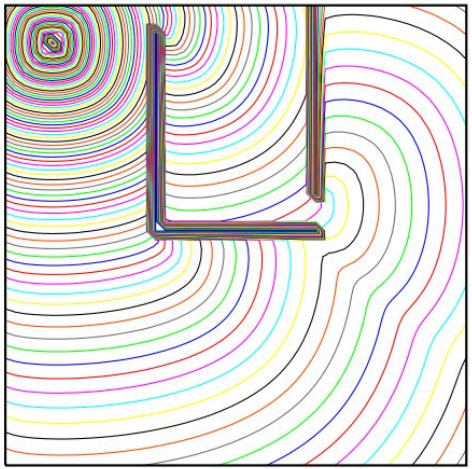
Smart Snek

A neural map implementation

What is neural mapping?

- Provided that the robot has located both the **obstacles** and its **target**, it can construct a neural map.
- The map looks like this:



How is such a map constructed?

1. An estimate of the grid/map is created
2. Starting from the most desirable point, we calculate the "desirability" of every single spot on the grid/map.
3. The robot will start building its path starting from its current location, and then gradually it will find its way by choosing the neighbouring block with the highest desirability.
4. The path will ultimately end to the final - most desirable point, which is its target.

My implementation

- Snake shaped robot.
- Square grid, n x n dimensions.
- Randomly generated obstacles.
- Randomly generated food (objective).
- Radar with both sonic (sonar) and x-ray scanner.
- Generated pseudo-colored neural map.

Navigation

Objectives:

- Eat food
- Avoid walls

How?

- Track food and walls with the sensors
- Once food is found, construct and follow the map

Navigation (2)

When following the map, the robot selects its next move based on 2 criteria:

- If there is a wall, avoid it.
- Select the square with the highest desirability.

When there is no map (still looking for food) :

- Move in a certain direction for 10 steps, then turn to a random direction.
- Avoid walls!

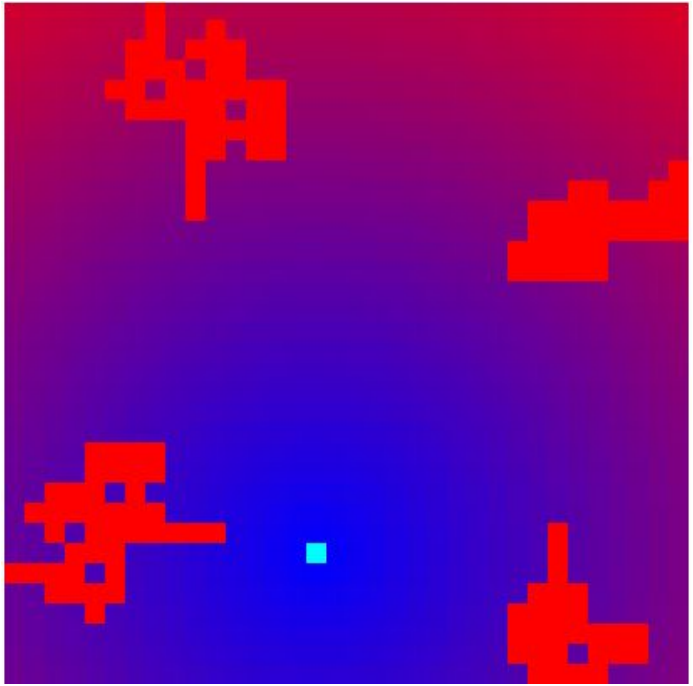
Neural map

- Bright red: 0 utility, obstacle
- Bright blue: 254 utility, desirable spot
- Blue to purple to red: 1-253 utility
- Cyan: 255 utility, food

The utility is calculated based on the distance of the pixel from the objective (food)

Higher utility => Higher desirability

Color = (R, G, B) = (|u - 255|, 0, u)



Thank you!

Future versions will include exciting features like:

- Additive White Gaussian or Meyer noise.
- Energy (battery) levels, where the food is actually the charger.
- Better, wave-like neural map implementation. (example)

Contact me: achatziioannou1@isc.tuc.gr