

# Martian Squirrel City Simulator

## Approach

### Calculating Where to Build

This is the most time expensive job because of the large number of places available to build the city. I decided to go around and check to cost to build a city by only checking the price to level a plain 500 by 500 in certain spots. When checking the cost to level a spot, instead of assuming the cheapest elevation as the average of all the elevations, I checked the each elevation from +/-10 from the elevation to see which elevation results in the cheapest plain. This strategy is really effective as it helped me find a much cheaper city than if I just found the average elevation of a spot. The cheapest spot to build a plain was chosen to be the place to build the actual city.

### Erosion Method

A good recursive flattening method would be optimal for this part. However, using a simple recursive method that goes around and increments all the dirt around it and goes onto that dirt will go into an infinite loop by trying to flatten into a circle. This will result in a stack overflow and crash the program. The way my program did it was to increment the perimeter of the city by one. Incrementing the initial dirt will cause everything to be stable and cannot crash because it is extremely simple to find an erosion safe values for increments of one. My program keeps incrementing the perimeter by one until it reaches the required elevation.

### Multi-Threading

Since a majority of the time is taken by trying to find the place to build a city, 8 searching threads were launched to scan the terrain simultaneously. 8 threads are optimal because that is the number of cores my processor has. I ran tests with 1, 5, 8, 10, 50, and 100 threads, but 8 turned out to be optimal on time. Each threads scans in a horizontal line one by one. No two threads repeat the same line. When one thread finds an optimal city, it saves it to the optimal city variable which is synchronized to allow only one thread to access it at once.

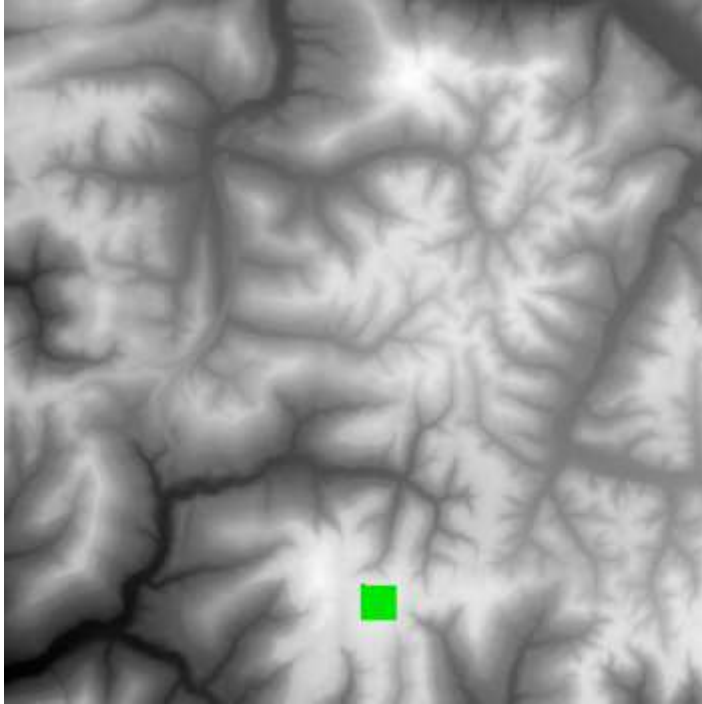
## Results

On the basic cost map, my program produced to following city:

Location: 5080, 8320

Total Cost: 377244

Total Run Time: 241.13 sec



## Future Ideas

### Better Erosion method

In the future, I can come up with a better erosion method rather than slowly incrementing the dirt by one. A better erosion method will increment the dirt to the desired value fully and then find the optimal erosion pattern.