# Settlers of Catan: Documentation

Risto Helinko
risto.helinko@student.tut.fi
211661

## How to use

The program doesn't take in command line parameters, so after compiling just run it (./a.out or however). There's only one .cc file, so compiling is simple.

It will then ask the number of test cases, and after that the tile indices for those cases.

As output it just gives the resource ID for the tile in question.

## The solution

Solution is based on arithmetic operations on the indices. This way the data structure searches are only done by index, and therefore are constant time. The actual building of the board is done in linear time, which I don't think could be improved.

The main idea is that if we keep track of certain things, we can quite easily deduce the neighbours of the tile that's being added. First of all, for every tile added previously, the corresponding resource ID is saved in a vector. Also, every time a tile is added, the corresponding resource ID count is incremented, so we can keep track of how much there is any resource. This way we don't need to calculate it separately.

The task then comes to find out the indices of neighbours, so that the possible resource IDs can be deduced.

The formula for neighbours depends on a few things:

- index of the current tile
- current *ring*
- current *side*
- whether it's a *corner* tile
- whether it's the first or last of the ring

The concepts of ring, side and corner are explained by figure 1. Ring 0 is just the tile indexed with 1. Ring 1 is tiles 2 through 7, and so forth. On ring 2, for example, index 8 would be the first tile and index 19 would be the last.

On ring 4, the second side is highlighted with red. Note that index 45 is also part of that side, but it's considered a corner tile, and has only two neighbours (at the time of adding) instead of three.

So all of these are kept track of, to eliminate laborous calculation, even though it could be done.

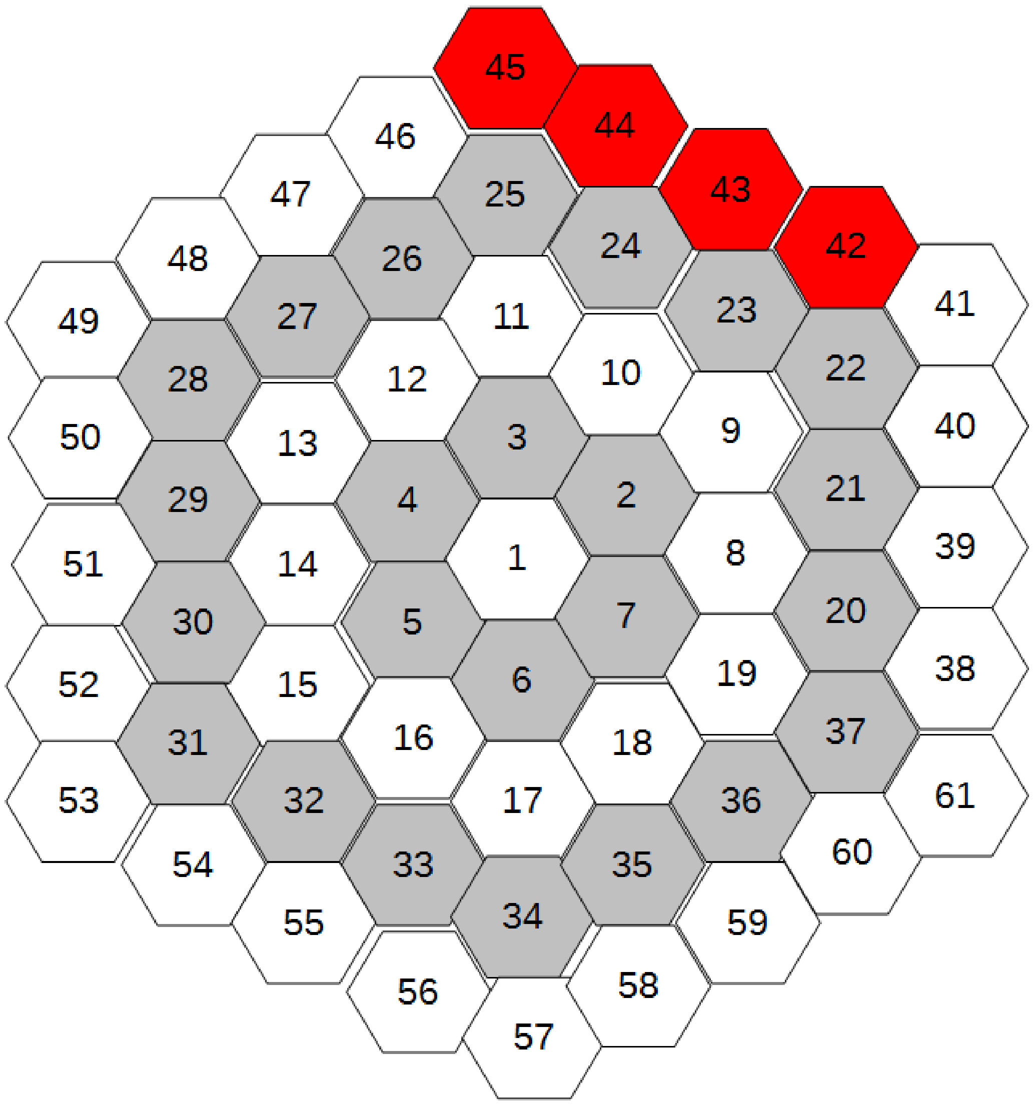

*Figure 1: First 4 rings of tiles with their indices*