

Scientific Computing - Assignment 2

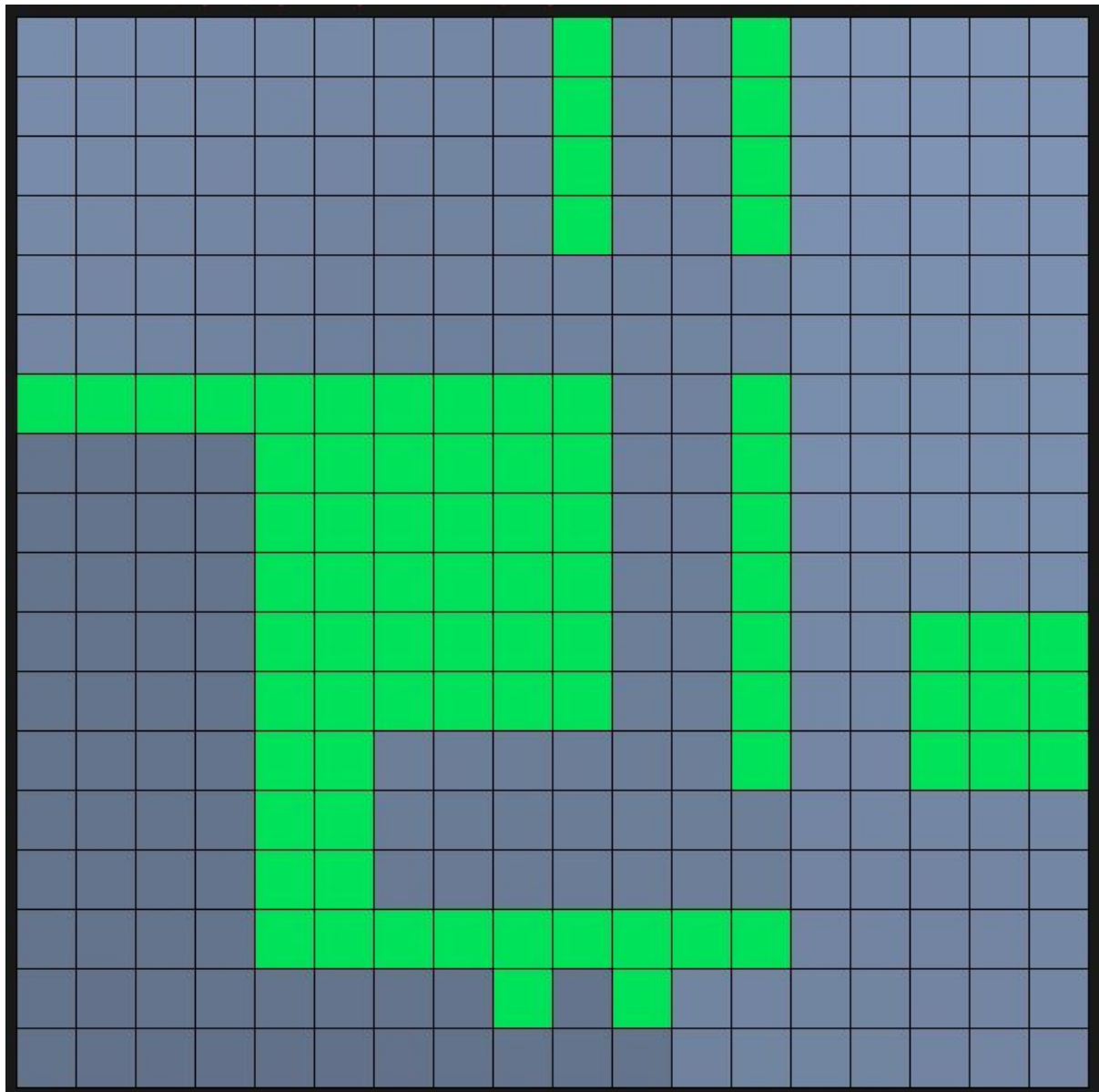
By Chloe Parbst

Roskilde University, Computer Science

5th term

Supervisor: John Gallagher

24th of November 2016



Introduction

For the second assignment, we could take a look at either the cluster analysis algorithm or cellular automata. I chose cellular automata.

We could either work on the Bears and Fishes or heatflow examples from the course, or make something ourselves.

I decided to simulate water flow by using cellular automata.

Solution

I wanted to make a water flow, based on cellular automata. My idea was, that some water sources could be created, which then eventually even out. I also added some terrain, which the water can't pass through, so the water would be forced to take another route.

The program is based on a world, which consists of some cells. These cells can have some different states: Water, ground, and void.

Each cell got a strength value, but based on their state, this value changes. Strength represents how many water units, that are present within the cell.

A ground cell is non-manipulatable by the water cells, and will not count as a neighbour to the water cell, since it won't be manipulated, and shouldn't manipulate the water back.

The void cells are empty cells, which can be manipulated by water cells. A cell is void, if it contains neither ground or water. The void cells strength is always 0.

Water cells are controlled by the cellular automata. A water cell will check how many neighbours it has, which are either void or water, with less strength than itself. The water from a cell drains, based on the number of neighbours. So if they don't have any neighbours, they won't drain, but otherwise the drain speed is increased by one for each neighbour. The neighbours' strength increases by one. This makes sure, that the water drained always ought to be equal to the water sent to the neighbours, and no additional magic water appears. The water also only spreads to other water cells, if the neighbouring water cell got a lower value, since the water should flow from a higher value towards a lower value.

The cellular automata only checks horizontal and vertical cells. This is a decision I made, since I was a bit inspired by Minecraft's way of handling water through cellular automata, which also doesn't check diagonally.

When a water cell transfers water to a neighbouring void cell, the void cell changes into a water cell. This cell then checks for neighbours with a lower value than itself, and drains towards them, while still being filled from the neighbours with a higher value.

The water is displayed in watery colour, with an alpha based on the strength value. So there more water there is in a water cell, the more colour it gets.