Cellular Automata
Cellular Potts Model
Assignment 4, Natural Computing

Josip Koprčina, S1062758
Reza Shokrzad, S1056369
Pratik Kulkarni, S1051211

In this report we discuss how we implemented obstacles and then added migrating cells to observe the obstacle's effect on those cells. Having understood that an *obstacle* is a cell that is immobile, we first figured out that such cells could be implemented by adding a high value of Lambda_V. This restricts their mobility.

We considered different scenarios for the size of 'obstacle' cells, and we decided that having multiple small cells would make for a more interesting obstacle for the migrating cells. With a canvas of size 256x256, the following figure demonstrates our obstacle cells:

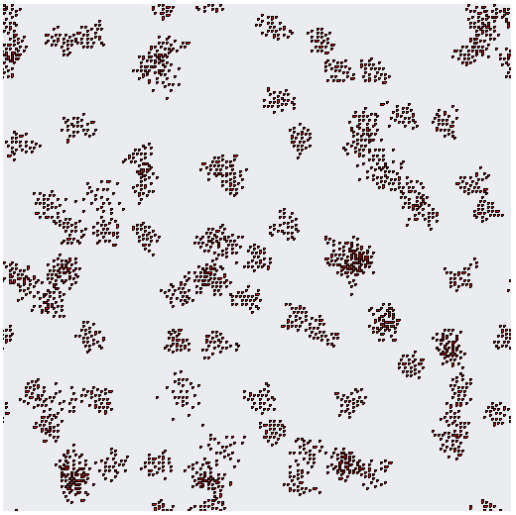

Figure 1: Obstacle Cells

We then added a more active cell kind that is larger and has a high value of MAX_ACT to facilitate migration of the cells. The following figure shows 20 of these cells:

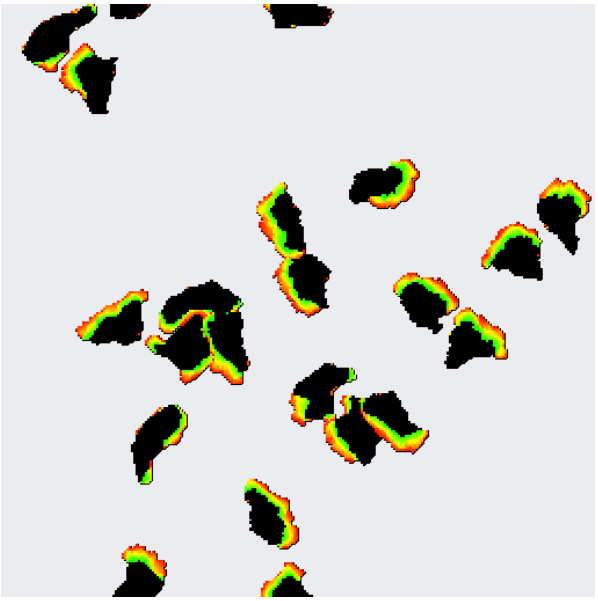

Figure 2: Active Migrating Cells

Combining both the Active and Obstacle cells resulted in the following figure:

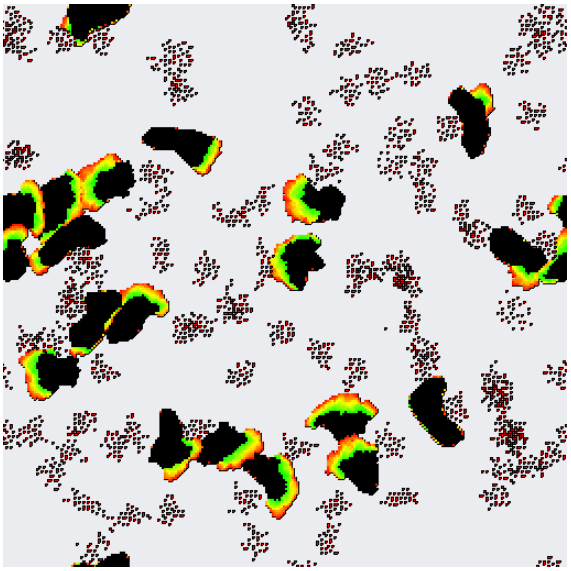

Figure 3: Active Cells with Obstacles

We expected the migrating cells to be locked into either positions or orbits, but it was surprisingly able to move past the obstacles with ease. This clearly happens because of the high Activity constraint. Lowering this constraint seemed to slow the process of migration, but not end it.

We had a suspicion on the torus characteristic of the canvas in that it could facilitate migration of the cells because they could jump around from the edges, but having tested the same without the torus setting we found out that it does not have any effect.

Further, we decided to place obstacles in a straight vertical line for a more robust plan to block the cells from moving. The following figure demonstrates the obstacle:

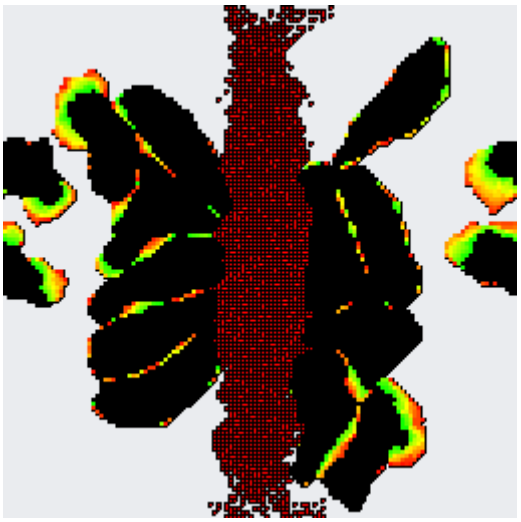

Figure 4: Vertical Obstacle

This wall of obstacle cells was indeed impenetrable as it left no space for the migrating cells to slip through.

Having run this simulation, we conclude that the potts model is indeed able to avoid obstacles given a high enough Max_act parameter value, unless there is some deliberate effort involved for it to not do so as shown in Figure 4. We also noticed that two active cells can also act as barriers to themselves if they are moving in opposing directions.

For our simulation, we used the following configuration for our Cellular Automata:

```
let config = {
   ndim : 2,
   field_size : [256,256],
   conf : {
       torus : [true,true],
       seed : 1,
       T : 15,
       J : [ [NaN, 12, 6], [12, 6, 16], [6, 16, 6] ],
       LAMBDA_V : [0,20,200],
       V : [0,425,25],

       LAMBDA_P: [0,2, 2],
       P : [0,260, 260],

       LAMBDA_ACT : [0, 300, 0],
       MAX_ACT : [0, 30, 0],
       ACT_MEAN : "geometric"
   }
```

And the following simsettings:

```
simsettings : {
   NRCELLS : [20, 95],
   BURNIN : 500,
   RUNTIME : 1000,
   RUNTIME_BROWSER : 20000,
   CANVASCOLOR : "eaecef",
   CELLCOLOR : ["000000","FF0000"],
   ACTCOLOR : [true,false],
   SHOWBORDERS : [true,true],
   zoom : 2,
   SAVEIMG : true,
   IMGFRAMERATE : 1,
   SAVEPATH : "output/img/Obstacle1"
   EXPNAME : "Obstacle1",
   STATSOUT : { browser: false, node: true },
   LOGRATE : 10
},
```