Spatial Games and the Prisoner's Dilemma

Carl Kingsford (`carlk@cs.cmu.edu`)

1. Assignment

1.1 The Prisoner's Dilemma

The Prisoner's Dilemma refers to the following situation: Two criminals who committed a crime together are being interrogated by the police in separate rooms. Each prisoner has a choice to either *cooperate* with his partner in crime and stay silent, or to be a *defector* and testify against his partner to gain favor with the police. The outcome of the interrogation depends on the choices made by the two prisoners:

- If both prisoners cooperate with each other and stay silent, then they each only go to prison, but for a short time (because there is little evidence against them).
- If both prisoners testify against each other ("D" strategies), then they both get long sentences.
- If one prisoner stays silent, but the other testifies against his partner, then the one who testifies gets a big reward: immunity from prosecution, and the person who stayed silent gets a long sentence.

We can model these outcomes in the following way: Each prisoner is either adopting a strategy of cooperation ("C") with his partner or defection ("D") from his partner.

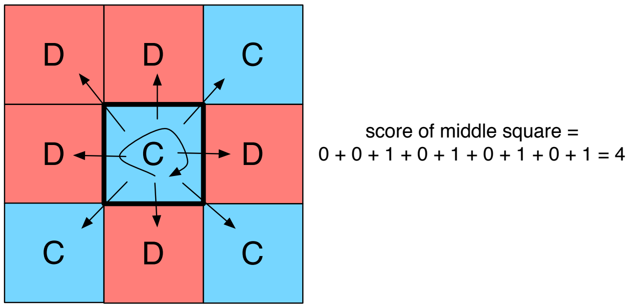
Prisoner 1	**Prisoner 2**	**Outcome**
C	C	both get 1 point
D	D	both get 0 points
D	C	prisoner 1 gets b points and prisoner 2 gets 0 points
C	D	prisoner 1 gets 0 points and prisoner 2 gets b points

That is: 0 points for a long sentence, 1 point for a short sentence, and b points for immunity. Here $b > 1$ is a parameter that says how much reward a prisoner gets for being the only defector.

The Prisoner's Dilemma is widely studied as a model for real policy situations. Pollution is a good example: if no one pollutes (that is everyone adopts a "C" strategy), everyone does well but everyone loses some short-term economic benefit. If everyone pollutes (all adopt a "D" strategy), we all lose. But if you live in the only country that pollutes (the only "D" country), you get the economic benefit without the global cost (a big reward for you).

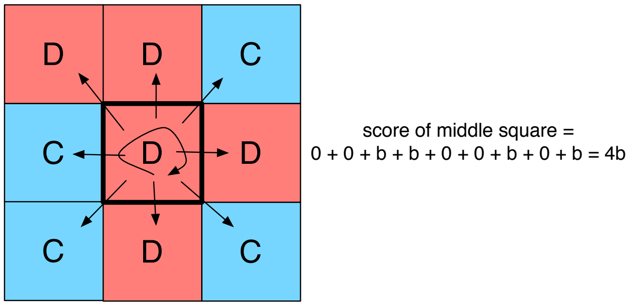
1.2 Spatial Games

This assignment adds a twist: we assume we have n^2 prisoners arranged in cells in a prison in an n-by-n grid. The prisoners are each either C-prisoners (cooperators with their fellow prisoners) or D-prisoners (defectors who inform to the police). The D-prisoners testify against the prisoners in the cells that neighbor theirs (and their own cell!) and the C-prisoners always stay silent. The reward to a prisoner is the sum of the points according to the scheme above. For example, consider this C-cell and its neighbors:



The center cell plays 9 prisoner dilemma games, and gets a total of 4 points: 0 points for interacting with his D neighbors, and 4 points for interacting with his C neighbors (including himself).[1]

Another example:



The center cell gets $4b$ points: 0 points for interacting with the other defectors, and b points for interacting with each of its C neighbors.

After calculating all the scores, each prisoner gets to change strategies: they adopt the strategy (C or D) of the prisoner in their neighboring cells (including their own) with the most points. After each round the point are reset to 0. Prisoners on the boundary of the field will play fewer games since they will have fewer neighbors.

So, the overall procedure is:

1. prisoners are either C or D prisoners.
2. prisoners get points according to their neighbors using the point system in the table above.
3. after all the scores are calculated, the prisoners adopt new strategies (either C or D) according to the best strategy in their neighborhood in the previous round.
4. this procedure is repeated for a given number of steps.

These "spatial games" were first explored in:

> Nowak and May, Evolutionary games and spatial chaos, *Nature* **359**:826–829 (1992).

---

[1] We include self-interactions to match the original paper on these spatial games; they don't really affect the qualitative behavior of the system. You can imagine that each cell represents a group of prisoners who all behave the same way.

1.3 What to do

In this assignment you will write a program to simulate this group of prisoners arranged in a grid. You will write a program that can be run with the following command line:

```
./spatial FILENAME b STEPS
```

where FILENAME gives the file that contains the initial assignments of C or D strategies (in a format described below), STEPS is an integer that tells how many steps to run your program for, and *b* is the reward a D cell gets against a C cell as described above.

Some of the program has been written for you, and the basic structure of the program is already decided. The template file `spatial.go` contains the program structure. Your job is to fill in the functions that haven't yet been written.

The main tasks are to write functions to (a) read in the initial field, and (b) update the scores and strategies according to the rules above. See `spatial.go` for instructions.

1.4 Format of the initial field file

The FILENAME command line parameter will be the name of a file that contains the field dimensions and the initial type for each of the cells. The first line of the file will contain the number of rows and columns, e.g.:

```
10 15
```

means there are 10 rows, each having 15 columns.

Each subsequent line is a string of Cs and Ds of length equal to the number of columns. There will be a line for each row. Together, these rows give the initial strategies for each cell. For example:

```
CCCCCCCCCCCCCCC
CCCCCCCCCCCDCCC
CCCCCCCCCCCCCCC
CCCCDCCCDDCCCCC
CCCCCCCCDDCCCCC
CCCCCCCCCCCCCCC
CCCCCCCCCCDCCCC
CCCCDCCCCCCCCCC
CCCCCDCCCCDCCCC
CCCCCDCCCCCCCCC
```

The assignment contains several example files: `f99.txt`, `f100.txt`, `rand200-10.txt`, `rand200-50.txt`, and `smallfield.txt`.

1.5 Tips on how to start

First, install the template provided and make sure you can:

```
go build
./spatial
```

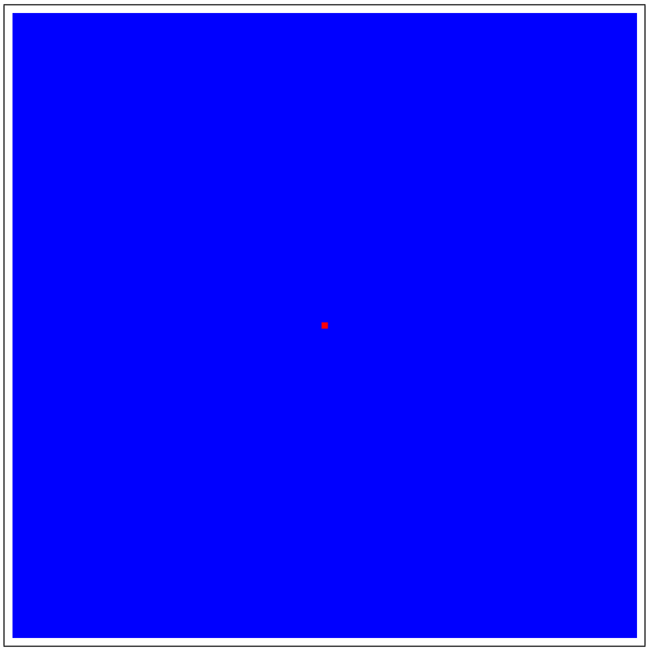
(On Windows you may have to type `.\spatial`.) You should get an error message about not enough command line arguments. This ensures you are set up to begin coding. **Do this today.**

Next, read through the template code in `spatial.go` and understand the structure of the program and how it is organized into functions. Particularly understand what the program calls a `field` and `Cell`.

Next, write the `readFieldFromFile`, and `drawField` functions. Compile and test your program by reading the starting field `f100.txt` using 0 steps of evolution using the command:

```
./spatial f99.txt 1.85 0
```

This will ensure that you are reading the field and writing the picture correctly. The field should look like this:

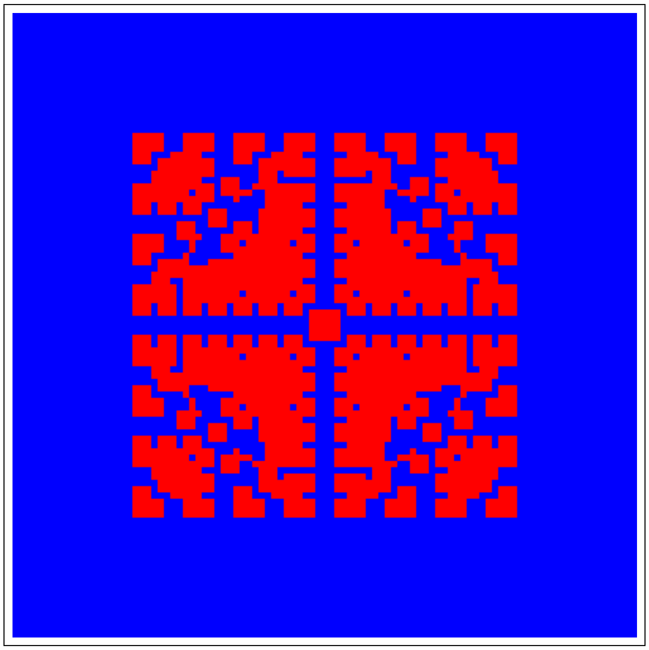


before you do any evolution steps.

Next, write `updateScores` and `updateStrategies`. Now test your program for more steps. If you run the command

```
./spatial f99.txt 1.85 30
```

You should get the picture:

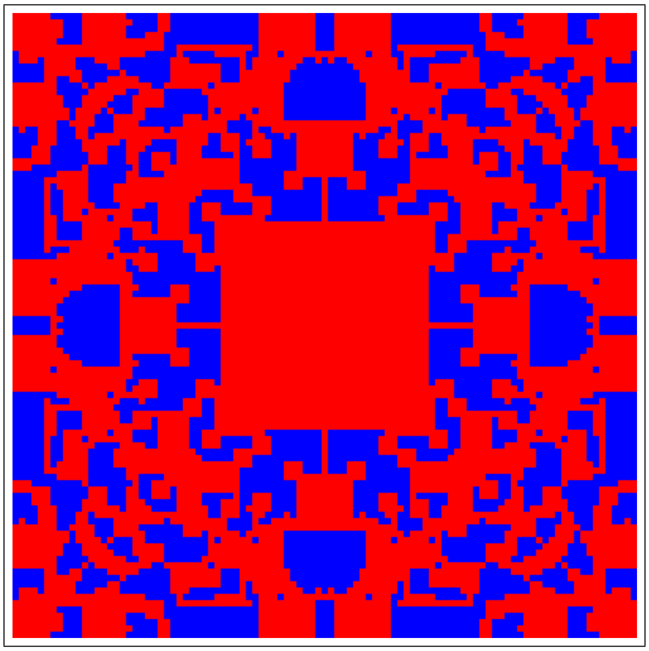


1.6 Explore your program

1. What happens if you execute the command:

```
./spatial f99.txt 1.85 80
```

You should get a picture that looks like:



Try lots of other numbers of steps and see how the pattern changes.

2. What about

```
./spatial f99.txt  1.8 200
```

What's going on here? Compare with $b = 1.81$.

3. Compare these two runs

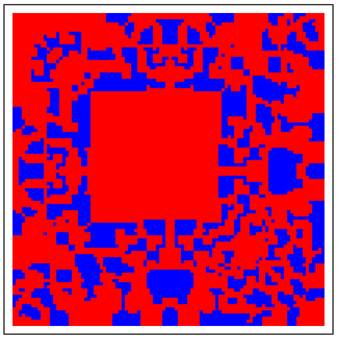
```
./spatial f99.txt  1.999999 80
./spatial f99.txt  2.0 80
```

Why are they so different?

4. The initial field f99.txt is a single D cell in the center of a 99-by-99 field of C cells. The initial field f100.txt is the same, except the field dimensions are 100 by 100. If you run

```
./spatial f100.txt 1.85 80
```

You get:

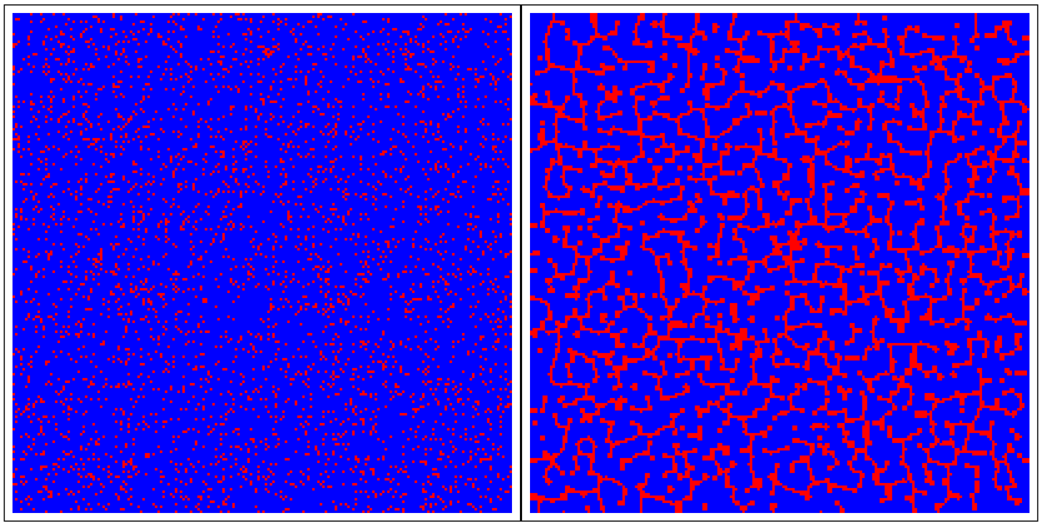


Notice the result is not symmetric. Why is this so different from what you get with f99.txt?

5. Compare these two runs:

```
./spatial rand200-10.txt 1.75 0
./spatial rand200-10.txt 1.75 200
```

You should get:



1.7 Learning outcomes

After completing this assignment, you should have:

- practiced reading a partially completed program and completing the missing pieces,
- learned how to read data from a file,
- understand how to use struct types,
- learned about the Prisoner's dilemma and spatial games,
- gotten additional practice drawing images,
- gotten additional practice with loops and if statements.

1.8 Extra Credit!

Implement the color scheme for drawing the field that is described in the Nowak and May paper cited above (where there are 4 colors depending on the current and *previous* state of a cell).