### SYSC 5104 METHODOLOGIES FOR DISCRETE EVENT MODELLING AND SIMULATION

#### Assignment 2: The Prisoner’s Dilemma Cellular Automata

### (2014 Fall)

# CHIDOZIE VINCENT ANALIKWU

(100967356)

Carleton University

**PART 1 – Model Description**

The prisoner's dilemma is a canonical example of a game analyzed in game theory that shows why two purely "rational" individuals might not cooperate, even if it appears that it is in their best interests to do so. It was originally framed by Merrill Flood and Melvin Dresher working at RAND in 1950. Albert W. Tucker formalized the game with prison sentence rewards and gave it the name "prisoner's dilemma" (Poundstone, 1992). More about Albert Tucker’s formalization can be read online at [http://en.wikipedia.org/wiki/Prisoner's\_dilemma]

Credits to the work done by (Nowak and May, 1992-1995) one can explore the concept of mutual help without going to prison. They suggested that spatial effects alone are sufficient to cause cooperative behavior. In their Cellular Automata (CA) model which this assignment will study, all of the sites of a two-dimensional lattice are occupied by players. The players interact with their nearest neighbor players in a pair-wise manner, over a number of time steps. The interaction strategies used by each player is determined as follows:

1. In a given time step, each player interacts with itself and with its eight nearest neighbors (i.e. the nine sites in the Moore neighborhood of the player) using either a cooperative strategy or an uncooperative strategy.
2. The total payoff to each player resulting from the nine interactions is determined, and each player adapts for the next time step the strategy of the player in its Moore neighborhood (including itself) who received the biggest payoff.

A 2-D Cellular Automata model will be developed and simulation carried out using CD++ tool with wrapped around borders to study this model.

**PART 2 – Formal Specification**

The formal specifications <X, Y, I, S, θ, N, d, τ, δint, δext, λ, ta> for the atomic Cell-Devs Prisoner Dilemma CA is defined as follows:

X = {

} // Input external event

Y = {

} // Output external event

I = <45, 0, {}, {}> // Model’s modular interface

with the neighborhood size  = 45,

there are no input/output ports in this model

S = {[\*11\*\*], [\*12\*\*], [\*21\*\*], [\*11\*\*] } // Possible states for a given cell

Where [\*11\*\*] = denotes a player cooperating now and cooperated in the previous step [Color – Blue]

[\*12\*\*] = denotes a player cooperating now and defected in the previous step [Color – Green]

[\*21\*\*] = denotes a player defecting now and cooperated in the previous step [Color – Yellow]

[\*22\*\*] = denotes a player defecting now and defected in the previous step [Color – Red]

The first \* before the number is used to keep track of current stage using two numbers {1,2}. {1} – indicates that the payoff of each player between itself and its eight neighbors be calculated. {2} – indicates that each cell should compare its current total payoff value between itself and eight neighbors and change its strategy to that of the cell with the highest payoff.

The last two \*\* after the number is used to keep the record of the total payoff for each player.

θ= { (s, phase, σqueue, σ)} //cells state variable

where s∈ S //defined above

phase = {1,2} as defined above {1} – Payoff should be calculated. {2} Strategy should be updated.

σqueue = {(vi, σi) / i ∈ N, vi ∈[0, infinity), σi ∈ R0+]

and σ ∈ R0+ //uses transport delay

N = {

} // set of input events

d = 100ms //delay for all individual cells

τ = local computing function which will be discussed in section 2.2

δint://internal transition function that is defined by CD++ automatically

δext://external transition function that is defined by CD++ automatically

λ = { is the output function

ta(passive) = INFINITY

ta(active) = d

**PART 2.1 – Neighborhood List**

|  |  |  |
| --- | --- | --- |
| (-1,1) | (0,1) | (1,1) |
| (-1,0) | (0,0) | (1,0) |
| (-1,-1) | (0,-1) | (1,-1) |

The neighborhoods of each cell of the Prisoner Dilemma CA consist of the 9 sites of the Moore neighborhood (i.e. itself and its eight neighboring cells) as shown in the diagram above.

**PART 2.2 –Detail Description and Implementation**

The cells are first initialized based on the case for studies. In this simulation the cells are initialized with one defector in center (whom we assumed also defected in the previous step [22]) surrounded by 8 defectors (whom we assume cooperated in the previous step [21]) who are then surrounded by cooperators (whom we also assume here cooperated in the previous step [11]) as shown below:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 11000 | 11000 | 11000 | 11000 | 11000 | 11000 | 11000 |
| 11000 | 11000 | 11000 | 11000 | 11000 | 11000 | 11000 |
| 11000 | 11000 | 12100 | 12100 | 12100 | 11000 | 11000 |
| 11000 | 11000 | 12100 | 12200 | 12100 | 11000 | 11000 |
| 11000 | 11000 | 12100 | 12100 | 12100 | 11000 | 11000 |
| 11000 | 11000 | 11000 | 11000 | 11000 | 11000 | 11000 |
| 11000 | 11000 | 11000 | 11000 | 11000 | 11000 | 11000 |

The meaning of the numberings and color coding have been explained in the formal specifications above.

**PART 2.2.1 – Playing the game**

The prisoner dilemma game is played over a number of rounds (i.e. time steps), in each of which every prisoner (site) on the lattice interacts with itself and with the eight nearest neighbor sites in the Moore Neighborhood.

Each round of interaction proceeds as follows:

**First Stage:-** Each prisoner interacts with the prisoners in its Moore Neighborhood in a pair-wise manner. And for each interaction a payoff value is computed as follows:

|  |  |  |  |
| --- | --- | --- | --- |
|  | OPPONENT’S STRATEGY | | |
| PLAYER’S STRATEGY |  | CCOOPERATE | DEFECT |
| CCOOPERATE | 1 | 0 |
| DEFECT | b | 0 |
|  |  |  |  |
| The Payoff Matrix | | | |

If the player and its neighbor both cooperate, the player gets a point

[Cooperate + Cooperate = 1]

If the player and its neighbor both defect, the player gets nothing.

[Defect + Defect = 0]

If the player defects and its neighbor cooperates, the player gets b points (b>1). In this simulation our b values is chosen to be 1.85. Other values of b can be chosen to simulate interesting results.

[Defect + Cooperate = b]

If the player cooperates and its neighbor defects, the player gets nothing.

[Cooperate + Defect = 0]

Note that other values corresponding to these parameters can be chosen. The values are used to simplify the simulation.

After all interaction the total payoff is calculated and the game moves to the next stage. A sample computation of the payoff matrix for stage 1 is shown below:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 21108.0 | 21108.0 | 21108.0 | 21108.0 | 21108.0 | 21108.0 | 21108.0 |
| 21108.0 | 21107.0 | 21106.0 | 21105.0 | 21106.0 | 21107.0 | 21108.0 |
| 21108.0 | 21106.0 | 22109.2 | 22105.6 | 22109.2 | 21106.0 | 21108.0 |
| 21108.0 | 21105.0 | 22105.6 | 22200 | 22105.6 | 21105.0 | 21108.0 |
| 21108.0 | 21106.0 | 22109.2 | 22105.6 | 22109.2 | 21106.0 | 21108.0 |
| 21108.0 | 21107.0 | 21106.0 | 21105.0 | 21106.0 | 21107.0 | 21108.0 |
| 21108.0 | 21108.0 | 21108.0 | 21108.0 | 21108.0 | 21108.0 | 21108.0 |

The reason for change from 1 to 2 as initial value of each cell is to keep of track of each stage of the game.

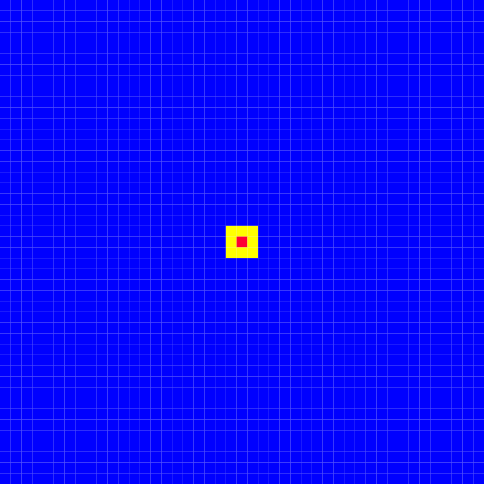
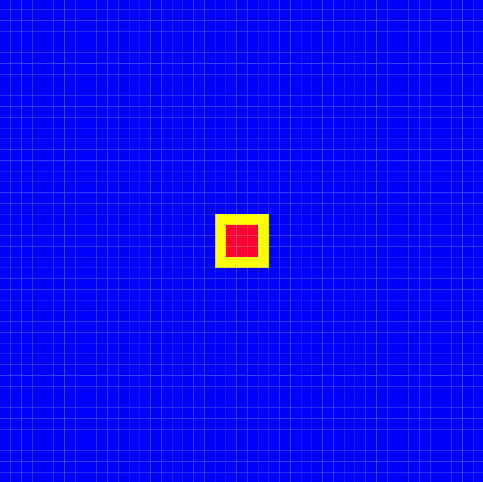
**Second Stage:-** Each prisoner now compares its total payoff value with itself and all the other prisoners in its Moore Neighborhood looking for the prisoner that is the most successful in that round. The prisoner will then update strategy to the one used by its neighbor to achieve the highest score.

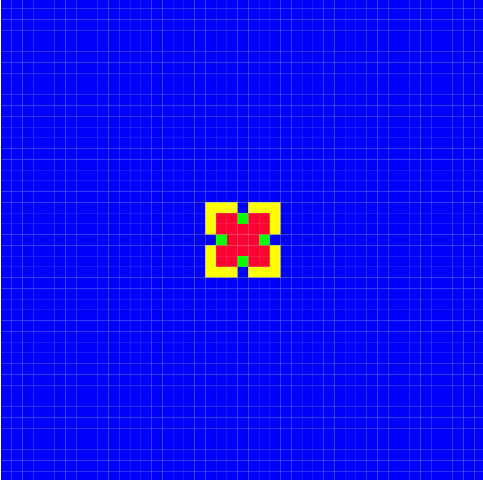
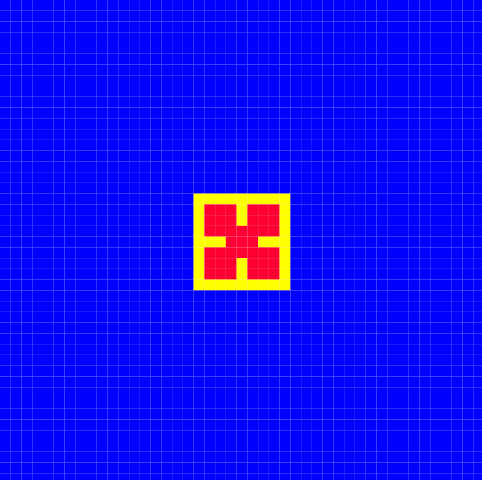
For example the cell that was colored with green (just for explanation purpose) has a value of [“2”11”05.0”] where 2 indicates the current stage, 11 indicates that in the previous and current round he was cooperating and 5 is his total payoff value. The prisoner now looks around his neighborhood and discovers that the highest scorer is “22109.2” with a total payoff value of 9.2. The prisoner updates his strategy to that which was used by that prisoner to achieve that highest score which in this case is 2 (defecting). So his strategy for the next round will be “21” (previously cooperated and now set to defect). After all strategies have been updated the prisoners go in for the next round. A sample of strategy updated for stage 2 is shown below:

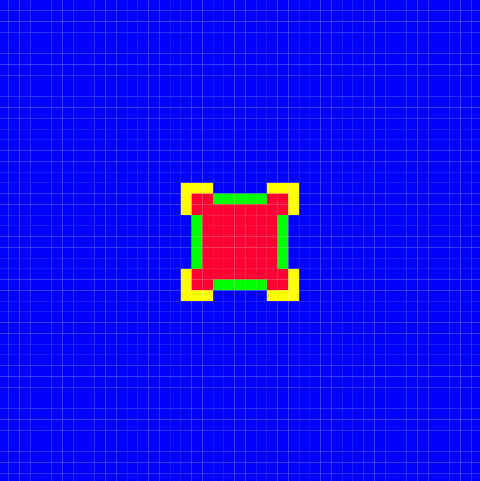
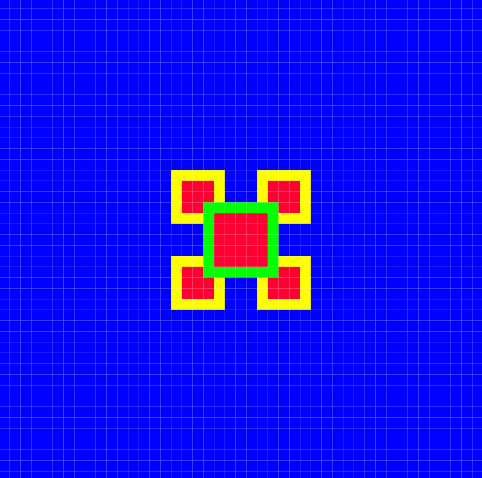
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 11000 | 11000 | 11000 | 11000 | 11000 | 11000 | 11000 |
| 11000 | 12100 | 12100 | 12100 | 12100 | 12100 | 11000 |
| 11000 | 12100 | 12200 | 12200 | 12200 | 12100 | 11000 |
| 11000 | 12100 | 12200 | 12200 | 12200 | 12100 | 11000 |
| 11000 | 12100 | 12200 | 12200 | 12200 | 12100 | 11000 |
| 11000 | 12100 | 12100 | 12100 | 12100 | 12100 | 11000 |
| 11000 | 11000 | 11000 | 11000 | 11000 | 11000 | 11000 |

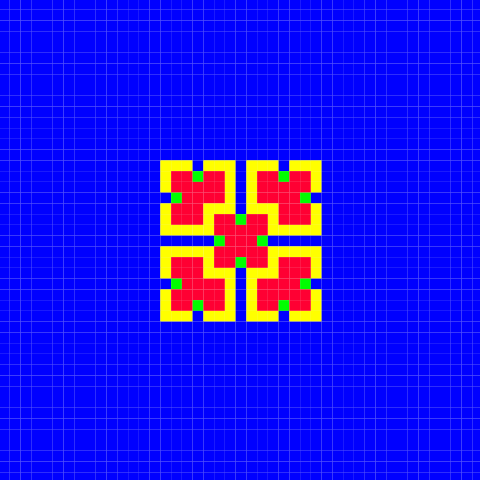
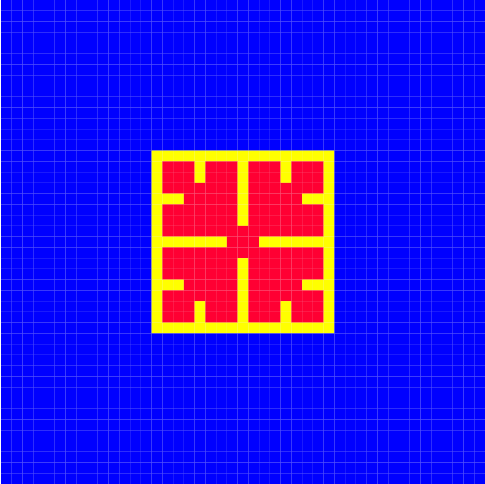
**PART 2.3 – Simulation Result and Conclusion**

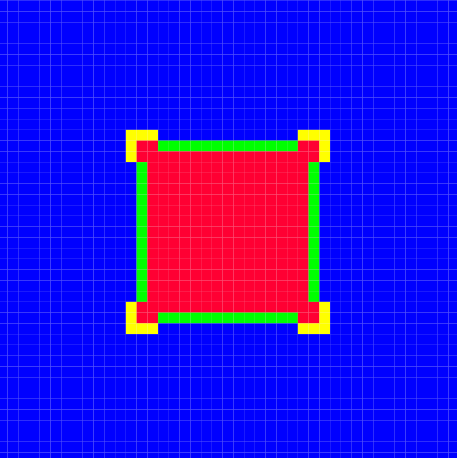
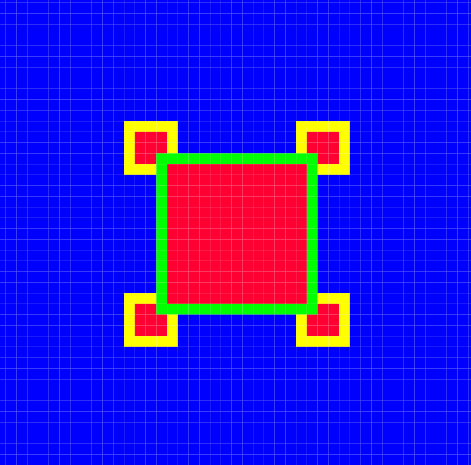
The simulation was carried out for a b value of 1.85. Other interesting result will be achieved for a different b values (b > 1).

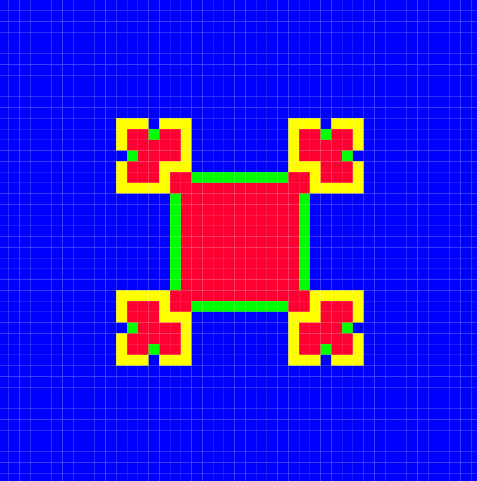
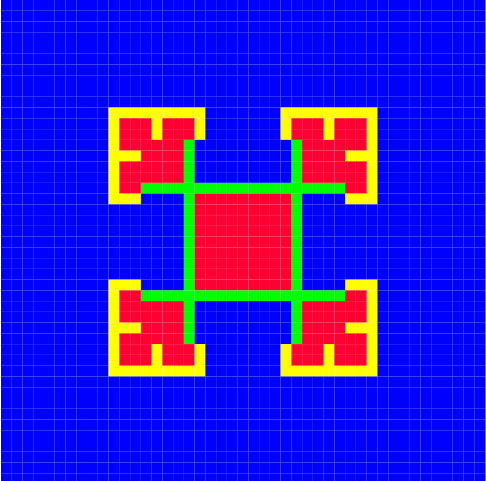
** **

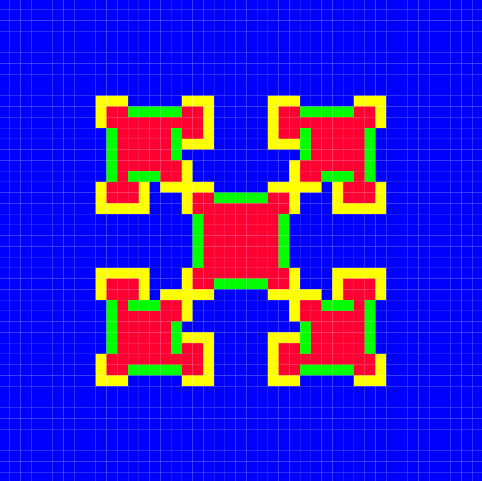
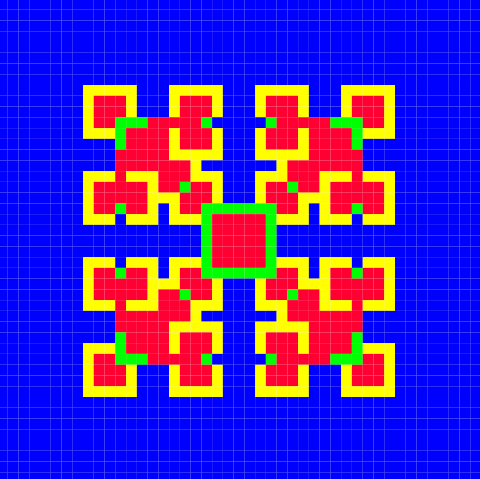
** **

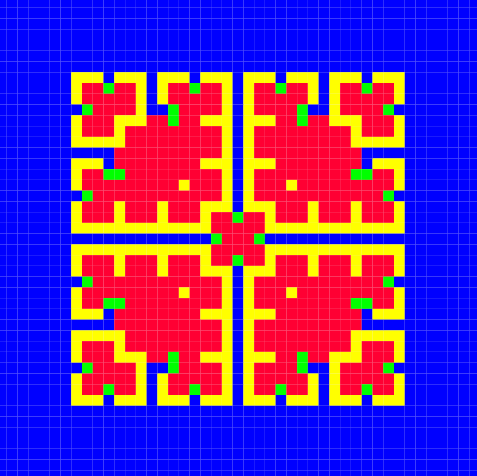
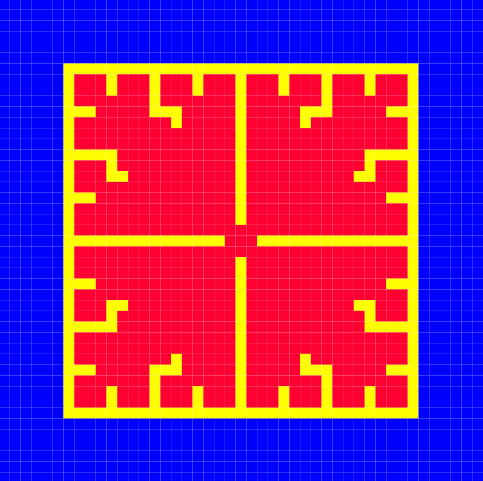
** **

** **

** **

** **

** **

** **

**Conclusion**

One of the enduring problems of evolutionary theory, from Darwin's time to today, is the evolution of altruism: cooperative behaviour may benefit the group, but it is inherently unstable to exploitation by cheats, who achieve greater reproductive success than other members of the group by enjoying the benefits of cooperation without paying the associated costs.

The prisoner’s decisions as simulated in this model highlight the deference between what is best from an individual’s point of view and from that of a collective. This conflict endangers almost every form of cooperation, including trade and mutual aid. The reward for mutual cooperation is higher than the punishment for mutual defection, but a one-sided defection yields a temptation greater than the reward, leaving the exploited cooperator with a loser’s payoff that is even worse than the punishment. This implies that the best move is always to defect, irrespective of the opposing player’s move. The logic leads inexorably to mutual defection. Most people feel uneasy with this conclusion. They do often cooperate, in fact, motivated by feelings of solidarity or seflessness. In business dealings, defection is also relatively rare, perhaps from the pressure of society. Yet such concerns should not affect a game that encapsulates life in a strictly Darwinian sense, where every form of payoff is ultimately converted into a single currency: offspring.