#### Predator-Prey Dynamics

#### Using Cell-DEVS

#### Winter 2016

#### Assignment 2

Talal Bin Qamar 100731129

Carleton University

**Conceptual Model**

A predator-prey model will be simulated using Cell-DEVS. The goal will be to simulate sharks and fish based on Alexander K. Dewdney model on the fictional planet of Wa-Tor1 (torus shape area which means the grid is looped).

The rules are as follows:-

* 20 by 20 grid
* Each cell will either have shark , fish or empty water
* Neighborhood will be the 4 adjacent squares including the current occupied square
* Movement for fish is random , after a certain time period fish will reproduce
* Shark will be deprived of energy after a step , upon reaching 0 energy it will die
* Shark will eat fish if it moves to a cell that had fish and gain energy
* Once a shark has survived a number of steps it will be able to reproduce
* Shark can also move randomly to adjacent square if empty

Running different initial populations of fish and sharks to see if extinction occurs or a perfect balance is maintained.

**Changes Made for implementation**

Movement of fish and shark was instead changed to movement by precedence. First choice would be up , second choice right , third choice down and fourth choice left. (clockwise)

Fish breeding and shark dying were too complicated to implement for the scope of the assignment and thus were left out.A fixed number of sharks were added (6) and the simulation was run with about 150 fish in a 20 by 20 grid.

**Cell DEVS Specification:-**

The cell devs specification for the model used is as follows :-

< *X, Y, S*, *N*, *type*, *d, τ*, *δ*int, *δ*ext, *λ*, *D* >

X = { i / i є Z }

Y = { i / i є Z }

S = {0 , 1, 2} // 0 is water , 1 is fish and 2 is shark

N =

wator(-2,0)

wator(-1,-1) wator(-1,0) wator(-1,1)

wator(0,-1) wator(0,0) wator(0,1) wator(0,2)

wator(1,-1) wator(1,0) wator(1,1)

wator(2,0)

(Neighborhood complications arose so as to avoid collisions and duplication between cells.)

Type = Inertial (Predator prey models are best simulated using inertial delays)

d = {.1s}

*τ* :- // All the rules from .ma File

rule : 1 100 { (0,0) = 1 and (-1,0) != 0 and (0,1) != 0 and (1,0) != 0 and (0,-1) != 0 }

rule : 2 80 { (0,0) = 2 and (-1,0) = 2 and (0,1) = 2 and (1,0) = 2 and (0,-1) = 2 }

// predator rules

rule : 0 80 { (0,0) = 2 and (-1,0) = 1 }

rule : 0 80 { (0,0) = 2 and (0,1) = 1 }

rule : 0 80 { (0,0) = 2 and (1,0) = 1 }

rule : 0 80 { (0,0) = 2 and (0,-1) = 1 }

rule : 2 80 { (0,0) = 1 and (1,0) = 2 }

rule : 2 80 { (0,0) = 1 and (0,-1) = 2 and (-1,-1) = 0}

rule : 2 80 { (0,0) = 1 and (-1,0) = 2 and ((-2,0) = 0 and (-1,1) = 0) }

rule : 2 80 { (0,0) = 1 and (0,1) = 2 and ((0,2) = 0 and (-1,1) = 0 and (1,1) = 0 )}

rule : 0 100 { (0,0) = 1 and ((0,1) = 0 or (1,0) = 0 or (0,-1) = 0 or (-1,0) = 0 )}

// fish movement rules

rule : 1 100 { (0,0) = 0 and (1,0) = 1}

rule : 1 100 { (0,0) = 0 and ((0,-1) = 1 and (-1,-1) = 1)}

rule : 1 100 { (0,0) = 0 and ((-1,0) = 1 and (-2,0) = 1 and (-1,1) = 1) }

rule : 1 100 { (0,0) = 0 and ((0,1) = 1 and (-1,1) = 1 and (0,2) = 1 and (1,1) = 1)}

rule : 2 80 { (0,0) = 0 and (1,0) = 2 and (1,1) != 1 and (2,0) != 1 and (1,-1) != 1}

rule : 2 80 { (0,0) = 0 and ((0,-1) = 2 and (-1,-1) = 2)}

rule : 2 80 { (0,0) = 0 and ((-1,0) = 2 and (-2,0) = 2 and (-1,1) = 2) }

rule : 2 80 { (0,0) = 0 and ((0,1) = 2 and (-1,1) = 2 and (0,2) = 2 and (1,1) = 2)}

*δ*int = NA

*δ*ext, = NA

*λ* = NA

D = {1 , .1s ; 2 , .08s} // fish transitions would take place at .1 s and shark transitions take place at .8 s

// Shark delay slower so that they can catch the fish

**Simulation and Testing :-**

A complex set of rules was written to simulate common movement for the fish first. There were some discrepancies found when there a simulation was run and of the 150 or so fish simulated around 10 fish were lost due to collisions. After carefully revaluating rules a solution to reduce the collisions further was not found and I proceeded to introduce the sharks. An example of the rules for the fish is as follows :-

// Following the precedence of North , East , West and South

rule : 1 100 { (0,0) = 0 and (1,0) = 1}

rule : 1 100 { (0,0) = 0 and ((0,-1) = 1 and (-1,-1) = 1)}

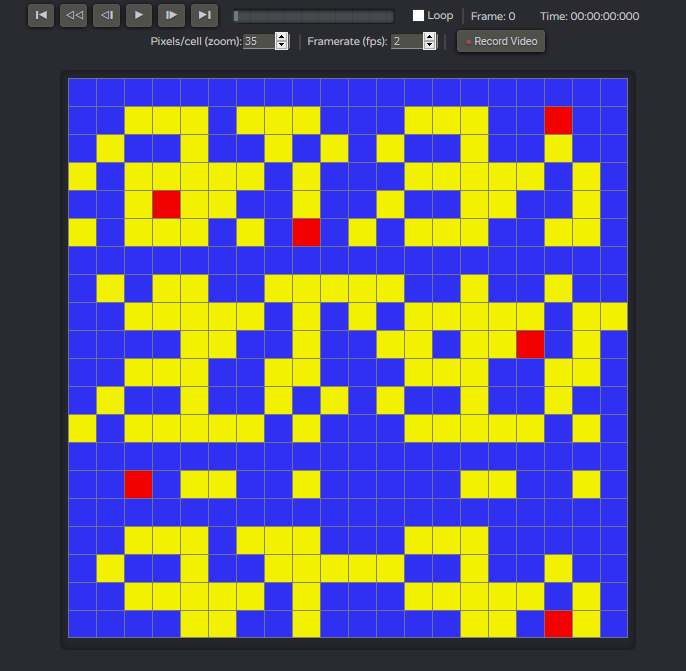
rule : 1 100 { (0,0) = 0 and ((-1,0) = 1 and (-2,0) = 1 and (-1,1) = 1) }

rule : 1 100 { (0,0) = 0 and ((0,1) = 1 and (-1,1) = 1 and (0,2) = 1 and (1,1) = 1)}

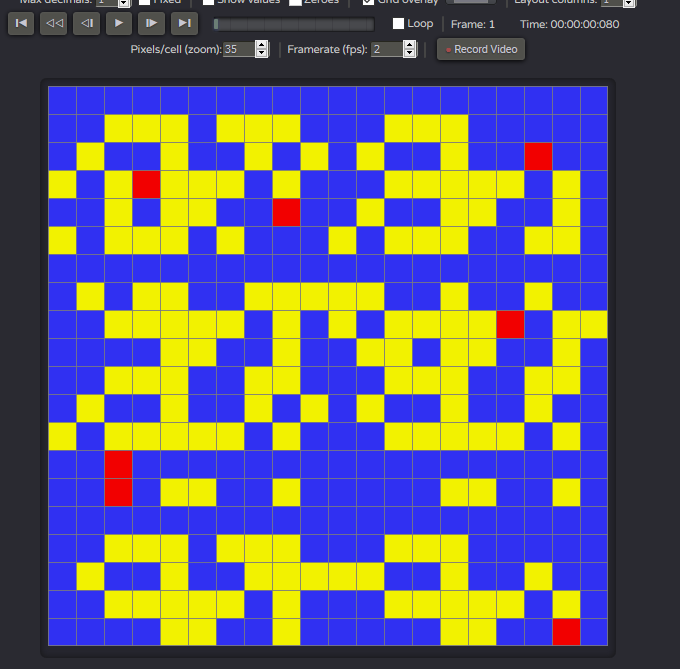
Once the sharks were added the simulation was run. The shark rules were also not without bugs as eventually there were cases where the sharks would disappear due to collisions. Revaluating the rules again did not provide a solution. The simulation works as expected for the most part as can be seen from the attached video and attached screenshots of the simulation.

Screenshots of simulation and run :-

Red is sharks , Yellow is fish and blue is water.



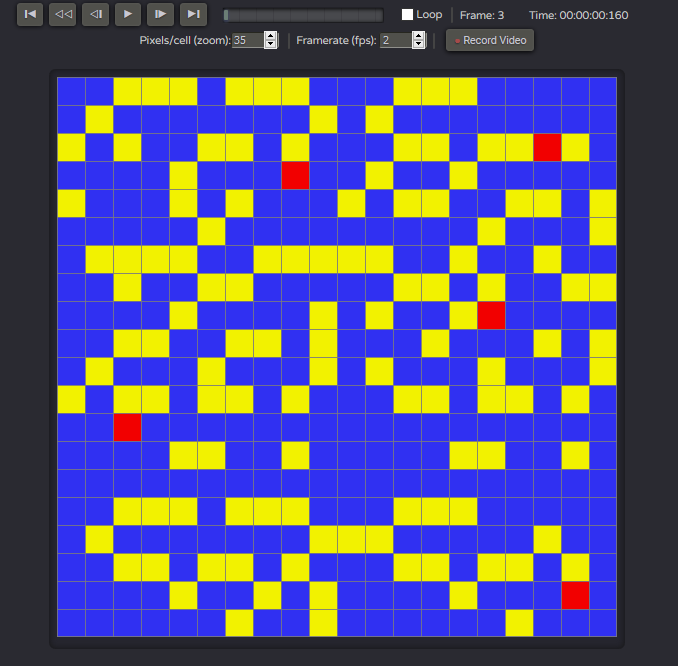
Intial conditions.



After .08s just the sharks have moved to the place where there is fish according to the precedence set.



After 0.1s the fish move according to the precedence set.



After .06s (.08s delay for sharks) each of the sharks again eat the respective fish according to precedence. However the top left most shark has gone due to a bug (4,4) probably due a collision that I could not find the reason to.

**Reference**

**[1]** "Computer Recreations: Sharks and fish wage an ecological war on the toroidal planet Wa-Tor" by *Alexander Keewatin Dewdney* [**http://home.cc.gatech.edu/biocs1/uploads/2/wator\_dewdney.pdf**](http://home.cc.gatech.edu/biocs1/uploads/2/wator_dewdney.pdf)

**[2] Wa-Tor** [**https://en.wikipedia.org/wiki/Wa-Tor**](https://en.wikipedia.org/wiki/Wa-Tor)