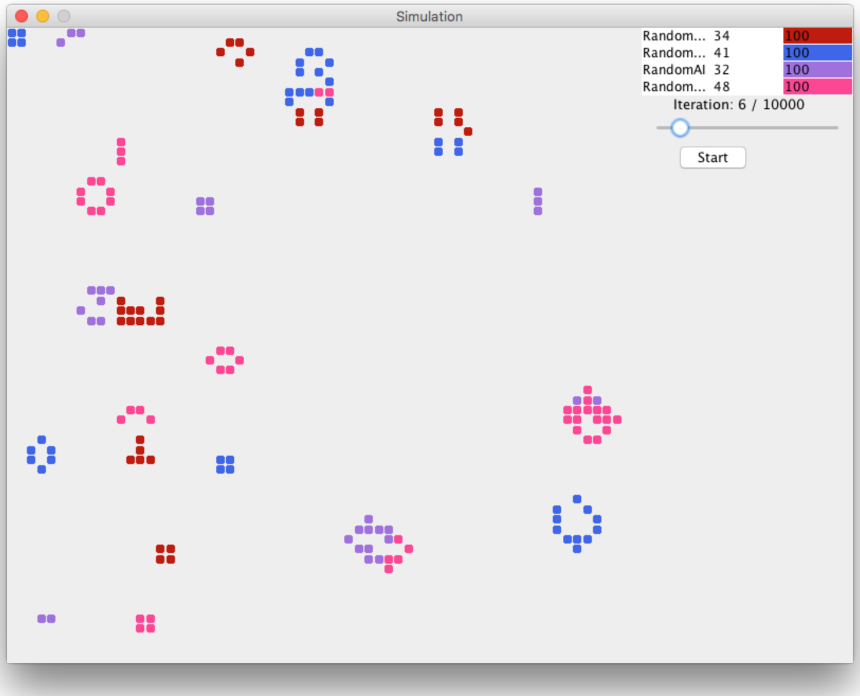
Data Structures
Tiveron
Cell Wars
Adapted with permission
from Mr. Fahrenbacher NWHS

Part One: A New Game (of Life)

1. Play around with this interactive Game of Life simulation.
2. Learn more about the Game of Life here.

Part Two: Revenge of the Cell

3. Download the CellWars.zip
4. Run the Simulation.java file to see the random AIs play against each other
   - The winners and losers will appear in the console
   - Run the simulation a number of times to get comfortable with the interface and the output
5. What do you think the 100 stands for? Try slowing the simulation all of the way down to help
   What do you think the numbers that are counting down stand for?
   Can you see the principals for Conway's Game of Life at play?



Random...	34	100
Random...	41	100
RandomAI	32	100
Random...	48	100

Iteration: 6 / 10000

Part Three: The Phantom Directions

6. Here's how the game works:

Each player will get to design their own AI which must be a subclass of CellAI

The progression of the game is follows:

a) The game is played on a square grid made of int values. Numbers that are 0 or larger in the grid represent player cells, while the value -1 indicates a dead cell. Each CellAI is given a unique id number. In a 4 player game, there will be 5 different values in the grid: -1, 0, 1, 2, and 3

b) At the beginning, each player has their id number placed in random spots in the grid

Did you figure out what the 100 means?

The number of cells in the grid depends on the number of players - you can find information about this in the populateGrid method of the Simulation class.

c) The first player loaded into the game selects a location in the grid to either

i) kill a cell that is alive (A player can kill their own cell or an opponent's cell)

ii) spawn an offspring if the selected cell is dead

After this choice is made, one iteration of the game of life is run.

d) After the first player's turn is over, then it is the second player's turn. This process continues until there is only one player left (or 0 if both players die out at the same time).

e) Here's how the rules of the Game of Life are modified for multiplayer

i) If a cell is dead and it has three living neighbors, then it will come to life. The type of spawned cell will depend on which neighbor is most common. If there is a three-way tie for most common, then the cell type is chosen at random from the most common neighbors.

ii) if a cell is alive and has more than 3 or fewer than 2 neighbors, then it dies

iii) if a cell is dead and doesn't have three living neighbors, then it stays dead

iv) if a cell is alive and has 2 or 3 living neighbors, then it will change to the type of the most common neighbor (or be choosen at random in the case of a tie).

Part IV: The Class Strikes Back

7. Before starting to create your own AI read through all of the steps in this section carefully.

8. To get started, make a subclass of the CellAI class and give it a code name.
In the comments at the top of your class make sure to include your name.

9. Override the getAIName() method so it returns **the code name for your AI**.

10. Override the select() method. This method gets called on your turn. The parameter passed to this method is the current state of the grid. If you call getID() it will tell you the id of your cells. Any number 0 or larger in the grid is a living cell, -1 is a dead cell. **You are to return a Location object that presents your choice of spot to either to bring a cell to life or to kill a living cell**. Remember that the first row is 0 and the first column is 0. I wonder what the last row and column would be? You might be wondering - can't I just modify the grid? Sorry - the grid parameter is an immutable object.

11. Spend time examining the provided AI classes. What strategy are they employing? Also, spend time looking through the Grid and GridFunction classes. Each provides some very helpful methods for building your AI. Spend some time thinking about what kind of strategy you want to employ. How can you leverage Conway's Game of Life rules to your advantage?

12. Get started building your AI. Your implementation must be fundamentally different than one of the samples provided. Have fun with it and experiment with different strategies. **Include a description of the strategy your AI is implementing in the header of the class file.**

Part V: Return of the Simulation (Class)

13. The top of the Simulation class holds some variables you can change in your testing. Specifically, there is an aisPerRound variable that keeps track of how many ais should fight at one time.

Part VI: The Grade Awakens

At the end of the project we will have a tournament where everyone's Ais will compete against each other. The tournament champion will earn the right to face off against the Cell Grand Master in a head to head competition. The winner will become the new Cell Grand Master!

Rubric

Create an AI that is different from the RandomAI and follows all the directions -Points awarded for completeness, creativity, originality, and use of available methods	30pts
Create an AI that defeats the RandomAI in head to head competition -Points awarded based on a 15-round match	15pts
Performance in the Cell Wars Tournament -Points based on position in the tournament	15pts
Total:	**/60 pts**
Schoology Grade:	**/30 pts**