Laboratory 14

Simulating Life & Heat

Objective

- Experiment with two simulation applets: "Game of Life" and "Heat transfer."

References

Software needed:

1) A web browser (Internet Explorer or Netscape)
2) Applets from the CD-ROM:
    a) Game of Life
    b) Heat transfer

Textbook reference: Chapter 14, pp. 451–455, 458–465

Background

The concepts of simulation and modeling are presented in Chapter 14, "Simulation and Other Applications."

Activity

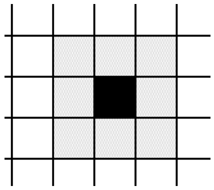
Part 1

Life was invented around 1970 by John Conway. It is an example of a *cellular automaton*, a simple "machine" whose parts are very rudimentary but whose overall behavior is surprisingly complex. The "parts" of the Life machine are the squares, and a few simple rules govern what these parts do.

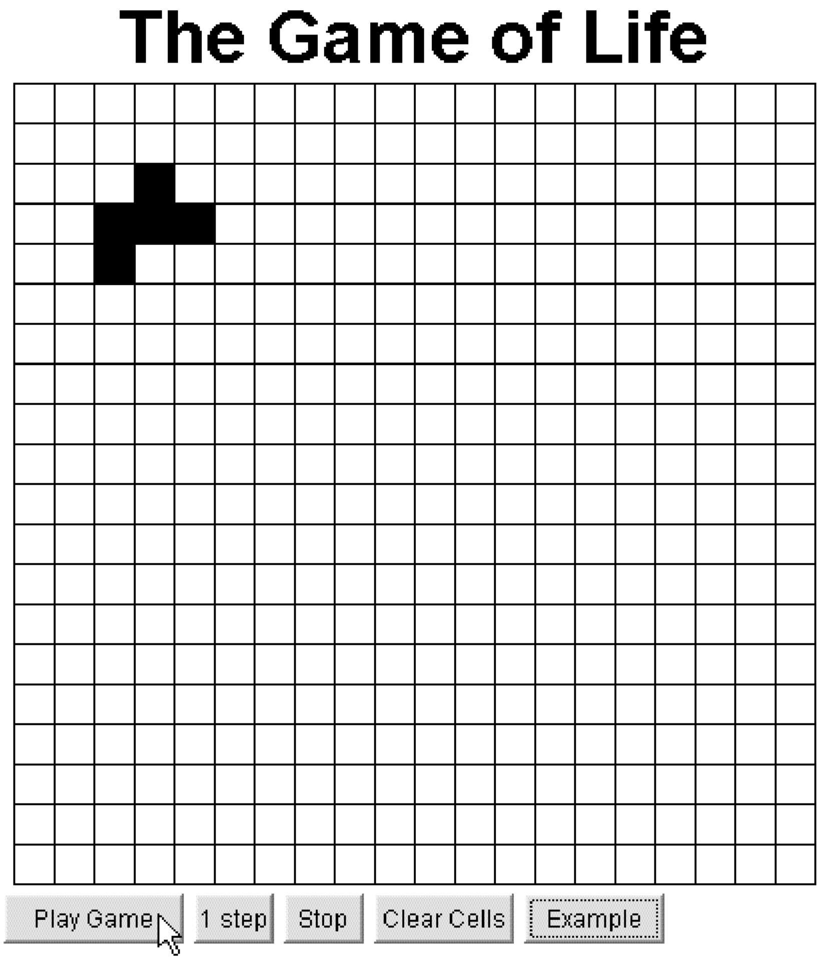
In Life, each square that is black is said to be "alive," and each white square is "dead." The board of squares is shown at a certain time t. The machine decides whether each square will be alive or dead at time $t+1$ using the following rules:

1) If a square has exactly three neighbors that are alive, it will be alive in the next time period.
2) If a square is alive and has one or no alive neighbors, it dies of loneliness.
3) If a square is alive and has four or more neighbors, it dies of overcrowding.
4) If a square has two neighbors, it is unchanged.
5) The squares at the edge do not change.

A little bit of terminology is necessary. A *neighbor* of a square is one whose row number or column number differs from the square's by just 1. This means the squares directly above, below, to the left, and to the right of a square are that square's neighbors, as are the squares on the diagonals. In the following picture, the gray cells are neighbors of the black square:



Start the "Game of Life" applet. Click on the *Example* button, which inserts a pattern of black squares on the grid, then click on *Play Game* to watch it go. Eventually this game stabilizes: the black square "life forms" will either cease moving and come to a stop, or reach a point where they continue moving in a consistent and unchanging pattern, or disappear completely. Click on the *Stop* button to end the game.



Now it's your turn to become master of this tiny universe. Clear the cells and click on several squares. If the square is white, it becomes black. If the square is black, clicking on it turns it white. Make a pattern and play the game.

You can either let the game run continuously by clicking once on *Play Game*, or you can advance the simulation one step at a time by clicking on *1 step*. See if you can follow the rules for a few cells by advancing only one step after trying to predict the results.

Various computer scientists have been fascinated by the Game of Life throughout the years. Steven Levy, in his 1984 book *Hackers,* tells about Bill Gosper and others at the MIT Artificial Intelligence lab who programmed Life on PDP-6 computers and then became obsessed with trying to predict what patterns would emerge. You can have some fun, too, (even become obsessed if you'd like) by experimenting with different patterns and categorizing their behaviors.

The pattern that the *Example* button inserts generates some interesting sub-patterns. Some of these patterns are very stable, such as four black squares that form a 2x2 square. It just sits there and does nothing (unless some neighbors come creeping up on it). Another stable pattern is a line of three squares. It seems to rotate forever. One pattern that also shows up is the glider, which moves across the screen. Gosper and his friends tried to create all kinds of gliders and other patterns that perpetuated themselves.

The Game of Life fits into several subfields of computer science, including simulation and artificial intelligence. It can be considered an example of simulation because Life models a simple universe and shows how to *discretize* (i.e., break into chunks) time and space. The link with AI is historical; many early AI researchers were fascinated by it and wondered what it meant. On a deeper level, Life is an example of how a few simple rules generate *emergent* behavior that is complex and unpredictable. Some AI researchers believe this is how plants, animals, humans, and societies derive their complexity: from simpler subsystems that interact in surprisingly complicated ways.

Tip

Game of Life can be used as a standalone Java application. If you use it as an application (not as an applet), you can load and save your pictures. To run the Java application, navigate to the folder containing the Game of Life class files and double-click on the **run_application.bat** file.

Part 2

The conduction of heat through a solid material is illustrated by the "Heat transfer" applet. It is similar to the Game of Life in that time and space are discretized. Simple rules are applied to the space chunks, called cells, in order to generate the values of the characteristic variables of these cells in the next time frame.

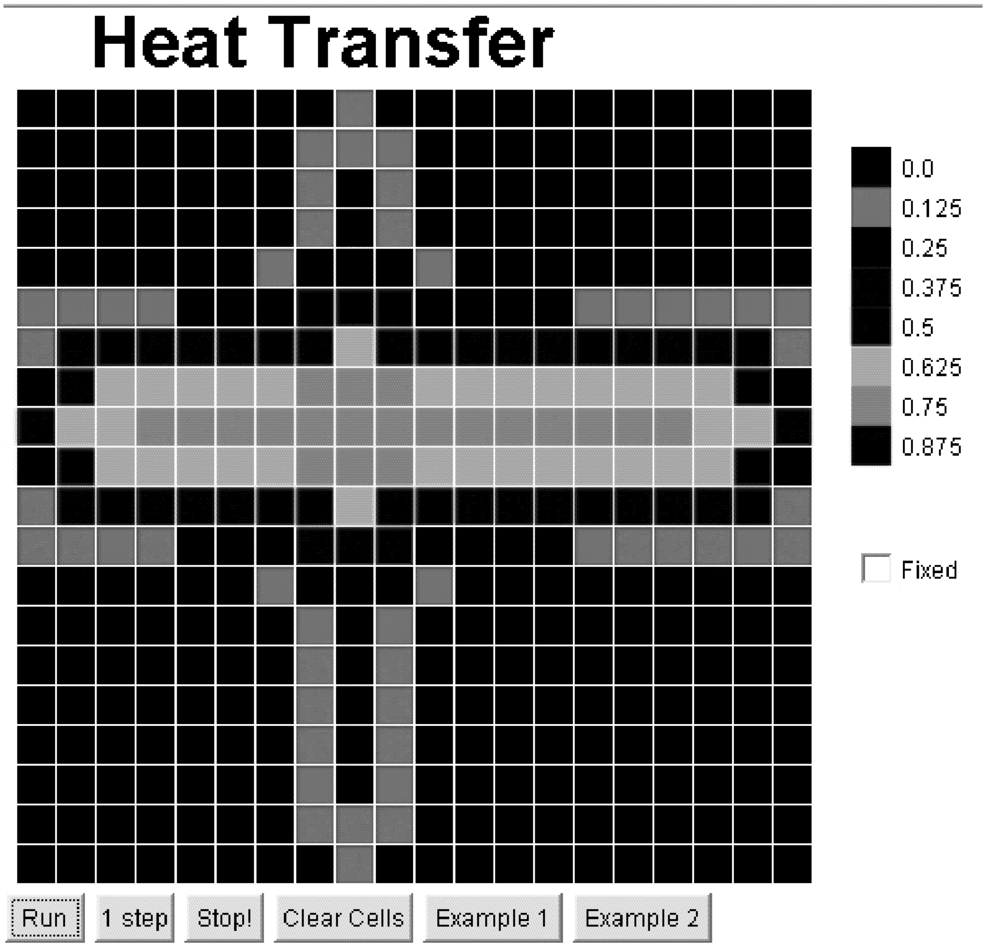
In the "Heat transfer" applet, the only characteristic value we are interested in is the temperature of the cell. The rule is quite simple: the temperature of a square in the next time step will be the weighted average of the temperature of that square in the current time step and the four direct neighbors' temperatures. This is called a five-point update stencil. (The Game of Life used an eight-point update stencil because it excluded the square in the middle but included the four diagonal neighbors.)

If you really want to impress your friends, show them the following equation, which governs the "Heat transfer" applet:

$$T_{i,j}^{(t+1)} = \frac{T_{i-1,j}^{(t)} + T_{i,j-1}^{(t)} + T_{i+1,j}^{(t)} + T_{i,j+1}^{(t)} + 4*T_{i,j}^{(t)}}{8}$$

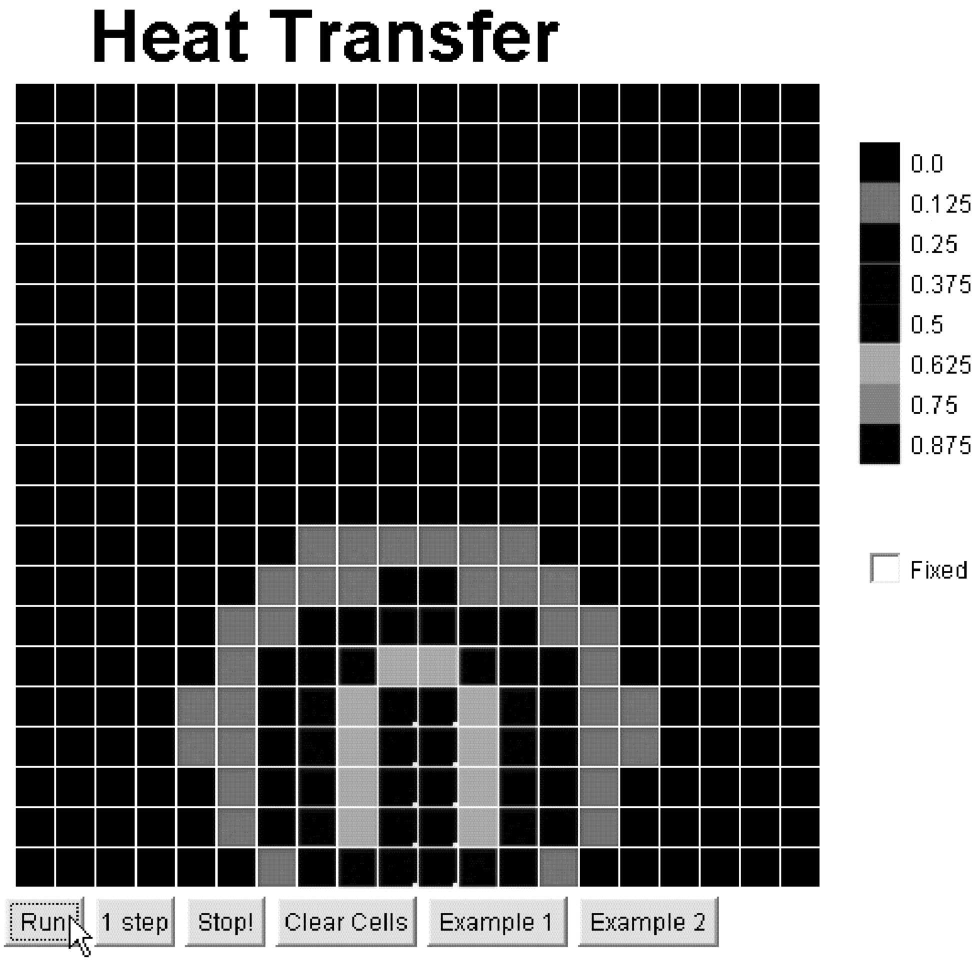
The subscripts are the row and column numbers. The superscripts in parentheses are the times. This merely says that the temperature of cell *i,j* at the next time step is the average of itself and its neighbors at the present time step. However, the cell itself is multiplied by 4, because its temperature counts for more than its neighbors. This is one parameter of the simulation and has the effect of *dampening*, or slowing down, the effects of a cell's neighbors on its new values.

Start the "Heat transfer" applet and click on *Example 1*. A pattern of colored squares appears, with a legend to the right indicating the temperature range (from 0 to 1.0). Red indicates the hottest temperature and black the coldest. This entire array of squares is assumed to be surrounded by an infinite number of always-cold squares. Click on *Run* to watch the simulation:



In this example, the entire grid of cells cools off to 0 eventually, making it completely black. (Do you agree with many leading physicists that this is the fate of our universe, although it probably won't happen for another 10^{100} years?)

Now stop the simulation and click on *Example 2*. This creates a small rectangle of red-hot cells near the bottom of the grid, and also fixes their values. You can tell when the values are fixed because a tiny white square appears in the lower right corner. Run the simulation for a while and watch the flame pattern emerge. Think of the red rectangle as a heating element or a burning wick:



You can create your own pattern by clicking *Stop!*, then *Clear Cells*, and then clicking your mouse on the grid. To set the current temperature, click on one of the legend squares to the right, then click on a square in the grid. If the *Fixed* box is checked, the temperature that you set for a square in the grid will be fixed and the simulation will not change it. A square can be changed to not fixed by unchecking the *Fixed* box, then clicking on the cell again. The little white square in the lower right corner tells you which cells are fixed.

Exercise 1

Name ______________________________ Date ____________________

Section ______________________________

One of the amazing things about the Game of Life is how utterly unpredictable it is. Classical science wants to make predictions about objects based on the characteristics of the objects. But two very similar objects in the Game of Life universe can cause wildly different behavior. Chaos theory grew out of this situation.

An object in the Game of Life is a pattern of black cells on the grid. In this lab we will investigate a series of similar objects and try to categorize their behaviors.

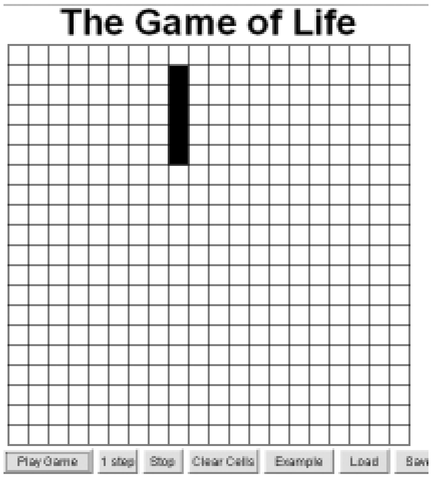
1) Start the “Game of Life” applet.

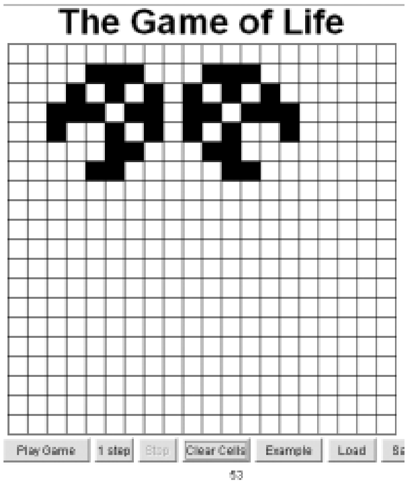
2) Draw a series of vertical lines in the middle of the screen of varying lengths. Start out with a single box. Click on “Play Game” to start the simulation. Let it run for a while and record the final configuration of the black cells in the table below and how many simulation steps it took to reach that configuration. Use the following codes for configurations:

D	disappear	all black cells vanish, the grid is all clear
O	oscillating	the grid changes between two different arrangements of cells forever
S	stable	the image is frozen into one arrangment
C	changing	the image keeps changing for a long time

As an example, a line of length 1 disappears very quickly. A line of length 3 oscillates between a horizontal and vertical line. A square of four black cells is stable.

For a second example, here’s a vertical line of 5 black cells and what the screen looks like after 53 simulation steps:





This pattern might be called C for changing. But let it run for longer and see if the screen disappears, or if the pattern freezes, or if it oscillates.

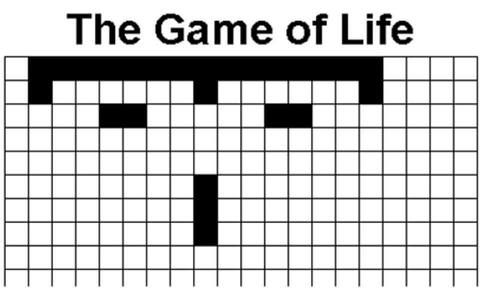
Make sure that the ends of your lines are not border cells, and that your line is in the middle of the screen. Since slightly different rules apply to border cells, the behavior of your line will change if it is up against an edge of the grid.

Here's the table to fill out:

Line length	ending config.	time to reach	notes
1	blank	1	disappears almost completely
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			

3) Are there any patterns in your data? Can you draw any conclusions?

4) Repeat the same experiment, only draw horizontal lines along the top of the grid. Once again, the variety of patterns and unpredictability is fascinating. Here is a screenshot of a 14-cell line, after a number of simulation steps. What the picture can't show is that part of the diagram oscillates.



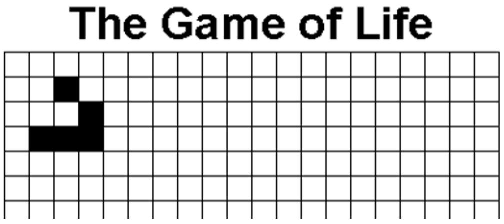
Line length	ending config.	time to reach	notes
2	________	________	
3	________	________	
4	________	________	
5	________	________	
6	________	________	
7	________	________	
8	________	________	
9	________	________	
10	________	________	
11	________	________	
12	________	________	
13	________	________	
14	________	________	

Exercise 2

Name ______________________________ Date ____________________

Section ______________________________

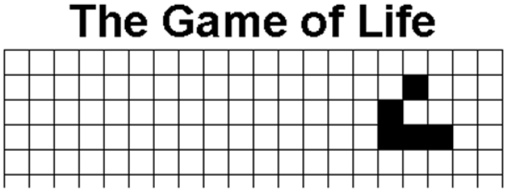
1) Start the "Game of Life" applet.
2) Create the following pattern on screen, placed in exactly the grid cells shown.



3) Is this a glider? Define "glider."

4) When does the glider stop and why?

5) Experiment with this pattern of five cells and see if every rotation is also a glider. For example, flip the pattern horizontally:



6) Flip the pattern vertically and see if that is also a glider.

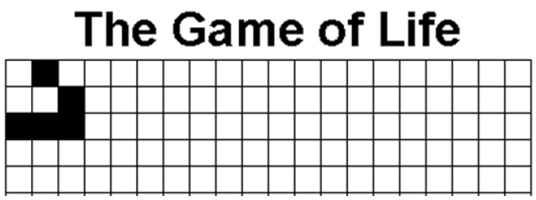
7) Flip the pattern vertically and horizontally and see if that is a glider, too.

Exercise 3

Name ______________________________ Date ____________________

Section ______________________________

1) Start the “Game of Life” applet.
2) Create the following pattern on screen, placed in exactly the grid cells shown. Notice that it is the same glider pattern, but now it is up against the edge.



3) Describe the behavior of this pattern. Is it a glider? Does it end? Does it oscillate?

4) Many patterns oscillate. A vertical line of three black cells oscillates. How many simulation steps occur before the original pattern repeats?

   vertical line of 3 cells __________

   pattern above __________

   horizontal line of 7 cells
   along the top __________

5) What is the difference between an oscillator and a glider?

Exercise 4

Name ______________________________ Date ____________________

Section ______________________________

1) Start the “Heat transfer” applet.
2) Check “Fixed” on the right edge of the black grid. Click on the red box in the legend. Then click on one cell right in the center of the black grid.
3) Start the simulation by clicking on the “Run” button. Run for about 30 seconds and take a screenshot.
4) Describe the pattern of colors that you see.

5) Does this pattern fit your intuition of what would happen if you put a red hot coal in the middle of a cold room?

6) List one way that the simulation doesn’t quite mimic what really happens to a red hot coal in a cold room.

7) Let the applet continue to run for a while. The number next to Running… should surpass 600. Is the simulation done? Is it stable or not? If it is changing, what can you say about the rate of change? Does the behavior of the simulation seem to fit reality?

8) While the applet is still running, click on the “Fixed” checkbox again to uncheck it. Click once on the red square in the legend. Then click on the red cell in the middle of your diagram. Since it is no longer fixed, its value can change. Remember that any cell that is fixed has a little white corner.

   Describe what happens to your simulation now. Does this make sense to you? That is, does it accord with your intuition about the physical world?

Deliverables

Turn in four screenshots from the “Game of Life” applet. Also, turn in one screenshot from the “Heat transfer” applet, along with your answers to the preceding questions.

If you are running Heat or Life as standalone Java applications, save your files. Your instructor may want you to hand them in. Consult the instructor for details on how to do this.

Deeper Investigation

The “Heat transfer” applet is an archetype of many simulation programs, from weather to stellar clusters to fluid dynamics. Your textbook shows the variables and equations that govern meteorological models on p. 461. Pretty scary, huh? Write down what variables you might need to include in a stellar cluster model, and which ones you should include in a fluid dynamics model.

The Game of Life is identical to Heat in principle, but unlike Heat, it seems to give more bizarre, unpredictable results. Speculate on why this might be the case, based on what you know about the underlying game’s rules and variables. (Hint: Think of what kinds of numbers are involved, namely reals or integers.)