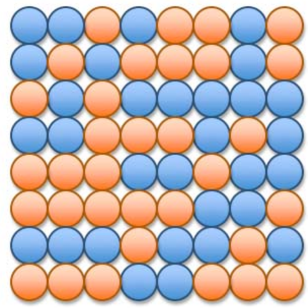


UNIT 12A
Simulation: Basics, Example

1

Announcement

- Exam 3 has been moved to November 28.

Simulation

- The imitative representation of the functioning of one system or process by means of the functioning of another [e. g., a computer program]. (Merriam Webster)
- Used in many contexts for
  - Performance optimization, safety engineering, testing of new technologies
  - Providing lifelike experiences in training, education, games
  - Gaining a better understanding of natural and human systems

Large Scale Simulations

- Computing power of today enables large scale simulations. For example,
  - Department of Defense: Battle simulations
  - National Center for Atmospheric Research : 1,000 year of climactic changes
  - Blue Brain Project at EPFL

Modeling

- The act of simulating something requires that a model be developed first.
- The model represents the system itself, whereas the simulation represents the operation of the system over time.

Computational Models

- Physical models: small-replicas
  - May not exist, may be unsafe to work with, expensive to build and change.
  - Some change too slowly over time.
- Computational models deal with these issues better.
- Computational sciences use computational models as the basis of obtaining scientific knowledge.

Abstraction

- In building models a major issue is to achieve a certain level of accuracy while keeping the complexity manageable
  - Identify factors that are the most relevant to the functioning of the system.

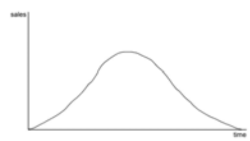
Stochastic Components

- Parts of the system may be stochastic (may exhibit random behavior).
  - Use statistical approximations

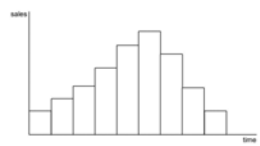
8

Types of Simulation

- How does the simulation reflect the passage of time?
  - steady-state
  - dynamic
    - continuous
    - discrete
      - time-stepped
      - event-driven



-continuous changes occur continuously (time is a real number)



-discrete changes at discrete points in time (time is an integer)

Types of Simulation

- How does the simulation reflect the passage of time?
  - steady-state
  - dynamic
    - continuous
    - discrete
      - time-stepped
      - event-driven

- time-stepped
Time intervals are regular. The simulation is organized with loop, such that each iteration represents the passing of a fixed amount of time.

- event-driven
Time intervals are irregular. Updates are associated with events, which are scheduled in advanced. Usually implemented with a priority queue.

Data Organization

- Simulations of physical phenomena tend to be either *grid-based or meshfree.*
  - grid-based (a.k.a. stencil codes)
    Data is associated with discrete cells at particular locations in a grid. Updates occur to each cell based on its previous state and those of its neighbors.
  - meshfree
    Data is associated with individual particles. Updates look at each pair of particles. More expensive than grid-based.

Climate Modeling

NASA/Goddard Space Flight Center Scientific Visualization Studio — GEOS-5 Modeled Clouds — http://svs.gsfc.nasa.gov/goto?3723

Categorization:

- Time-stepped or event-driven?
  - 30 minutes time steps (mostly)
- Grid-based or meshfree?
  - 5-km per grid cell

Example from Public Health Domain

- Use of modeling and simulation for disease spread

https://www.youtube.com/watch?v=nZxXqWM8nP4

Example from Public Health Domain

- Texas Pandemic Flu Toolkit

https://www.youtube.com/watch?v=0Q7zBk-PpRc

Example: Flu Virus Simulation

- Goal: Develop a simple graphical simulation that shows how disease spreads through a population.

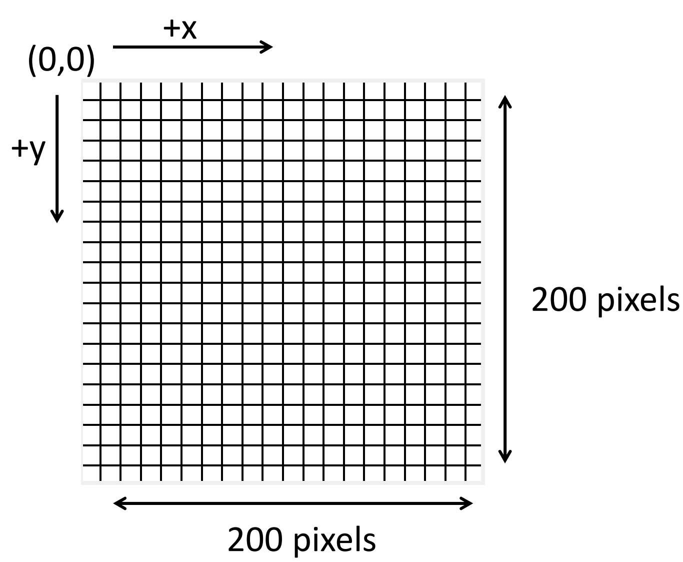
Model Assumptions

- A person starts off as healthy.
- Each day, a healthy person comes in contact with 4 random people. If any of those random people is contagious, then the healthy person becomes infected.
- It takes one day for the infected person to become contagious.
- After a person has been contagious for 4 days, then the person is non-contagious and cannot spread the virus nor can the person get the virus again due to immunity.

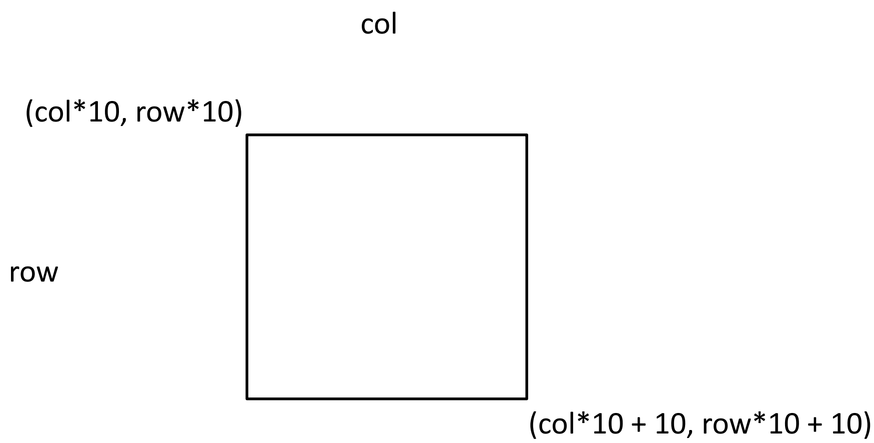
Data Abstractions

- Population
- Person
- Health state of a person

Displaying the Population



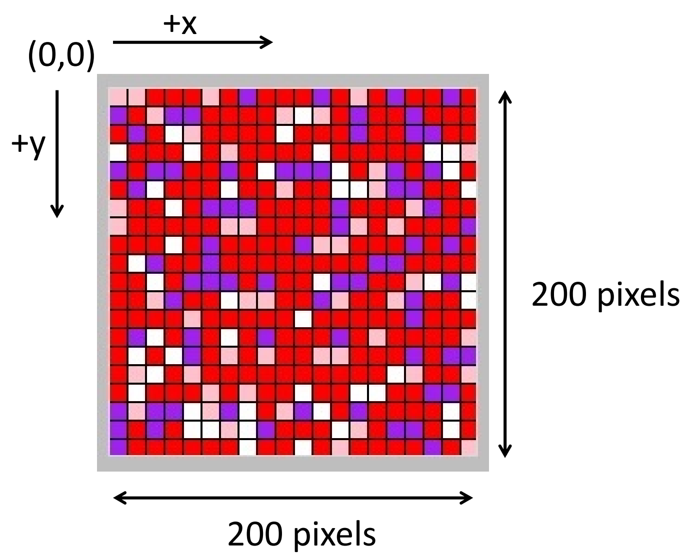
Displaying one person



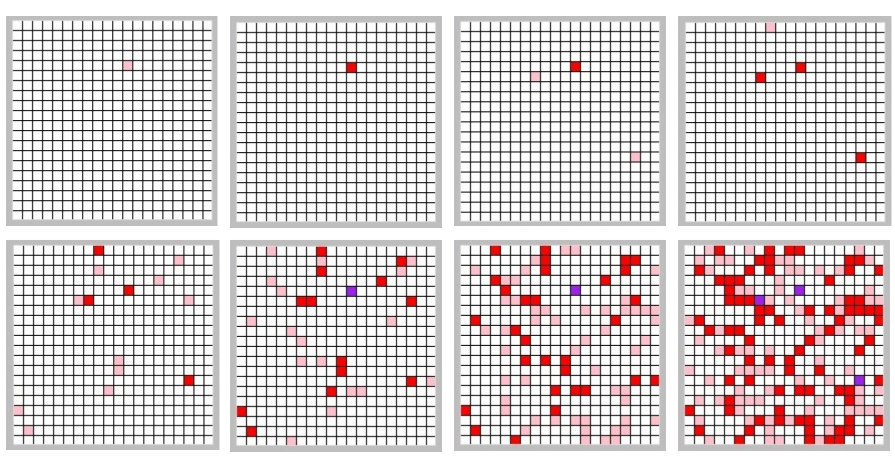
Health States

0	white	healthy
1	pink	infected
2	red	contagious (day 1)
3	red	contagious (day 2)
4	red	contagious (day 3)
5	red	contagious (day 4)
6	purple	immune (non-contagious)

Display of 400 people



Graphical Simulation



Displaying the matrix

```
def display(matrix)
  for row in 0..matrix.length-1 do
    for col in 0..matrix[row].length-1 do
      person = matrix[row][col]
      if person == 0     #healthy
        color = "white"
      elsif person == 1      #infected
         color = "pink"
      elsif person >= 2 and person <= 5   #contagious
         color = "red"
      else      #non-contagious or wrong input
        color = "purple"
      end
      Canvas::Rectangle.new(col*10, row*10, col*10+10, row*10+10,
:fill => color, :outline => "black")
    end
  end
end
```

Testing `display`

```
def test_display()
  # create a canvas of size 200 X 200
  Canvas.init(200, 200, "Testing_Display")
  # initialize matrix a randomly
  a = Array.new(20)
  for i in 0..19 do
    a[i] = Array.new(20)
    for j in 0..19 do
      a[i][j] = rand(7)
    end
  end
  # display the matrix using your display function
  display(a)
end
```

Checking Health State

```
def immune?(matrix, i, j)
  if matrix[i][j] == 6 then
    return true
  else
    return false
  end
end
```

```
def contagious?(matrix, i, j)
  if matrix[i][j] >= 2 and
matrix[i][j] <= 5 then
    return true
  else
    return false
  end
end
```

```
def infected?(matrix, i, j)
  if matrix[i][j] == 1 then
    return true
  else
    return false
  end
end
```

```
def healthy?(matrix, i, j)
  if matrix[i][j] == 0 then
    return true
  else
    return false
  end
end
```

Updating the matrix

```
def update(matrix)
  #create new matrix, initialized to all zeroes
  newmatrix = Array.new(20)
  for i in 0..19 do
    newmatrix[i] = Array.new(20)
    for j in 0..19 do
      newmatrix[i][j] = 0
    end
  end
```

```
#create next day
  for i in 0..19 do
    for j in 0..19 do
      if immune?(matrix, i, j)
        newmatrix[i][j] = 6
      elsif infected?(matrix, i, j) or contagious?(matrix, i, j)
        newmatrix[i][j] = matrix[i][j] + 1
      elsif healthy?(matrix, i, j)
        for k in 1..4 do     # repeat 4 times
          if contagious?(matrix, rand(20), rand(20)) then
            newmatrix[i][j] = 1
          end
        end
      end
    end
  end
  return newmatrix
end
```

```
def test_update()
  # create a canvas of size 200 X 200
  Canvas.init(200, 200, "Testing_Update")
  # initialize matrix a to all healthy individuals
  a = Array.new(20)
  for i in 0..19 do
    a[i] = Array.new(20)
    for j in 0..19 do
      a[i][j] = 0
    end
  end
  # infect one random person
  a[rand(20)][rand(20)] = 1
  display(a)
  sleep(2)
  # run the simulation for 10 "days"
  for day in 1..10 do
    a = update(a)
    display(a)
    sleep(2)
  end
end
```

Events by chance

If a healthy person contacts a contagious person, she gets sick 40% of the time.

```
if contagious?(matrix, rand(20), rand(20))
        and rand(100) < 40 then
     newmatrix[i][j] = 1
end
```

Neighbors

```
cell = matrix[i][j]

north = matrix[i-1][j]    NO!

if i == 0 then            YES!
 north = nil
else
 north = matrix[i-1][j]
```