

Today's topics

**Java Applications
Simulation**

**Upcoming
Software Engineering (Chapter 7)**

**Reading
*Great Ideas, Chapters 6***

What does it mean to be human?

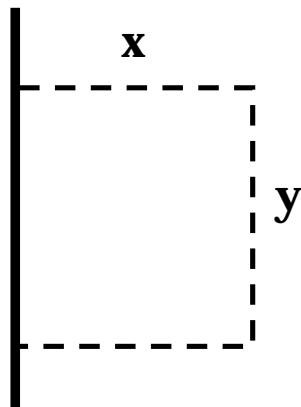
- **Tool User?**
 - **Some animals use tools**
- **Speech?**
 - **Some animals (whales?) seem to communicate by sound**
- **Do simulations?**
 - **???**
- **Many things we do could be called simulations**
 - **Drawing a diagram of something to build**
 - **Using a map to give directions**
 - **Moveable furniture cutouts on a floor plan**

Computer Simulation

- **As suggested before, can simulate without computer**
- **Computer greatly extends the limitations**
 - **Nowadays almost everything built is simulated first**
- **What are some of the things made possible by computer simulation?**
- **Early efforts:**
 - **Fancy camera lenses among first beneficiaries**
 - **Efficient paths for space ships**
 - **Population projections in relation to birth control policies**
- **Let's use the computer to find solution to simple problem**

Dog Lot Fence

- **Optimize:**
 - **I.e., give your dog the biggest lot in the face of constraints**
 - **Build lot against side of house**
 - **Fixed length roll of fencing (and posts)**
 - **Rectangular layout**



Length of fence is

$$2*x + y$$

- **Use program to try different values of x and y**
 - **Better than actually trying different layouts!!**

Fence

```
public class Fence extends java.applet.Applet implements
                                   ActionListener

{ TextField mInstruct;
  Label lLength;
  DoubleField gLength;
  Button bSimulate, bDisplay;
  TextArea mResults;
  int k;
  public void init()
  { lLength = new Label("Length");
    mInstruct = new TextField(70);
    mInstruct.setText(
      "Enter length of fence, the press Simulate or Display");
    gLength = new DoubleField(10);
    bSimulate = new Button("Simulate");
    bDisplay = new Button("Display");
    mResults = new TextArea(25,60);
```

Fence.2

```
bSimulate.addActionListener(this);
bDisplay.addActionListener(this);
add(mInstruct); add(lLength); add(gLength);
add(bSimulate); add(bDisplay); add(mResults);
}
public void actionPerformed(ActionEvent event)
{ Object cause = event.getSource();
  double fenceLength;
  if (cause == bSimulate)
  { fenceLength = gLength.getDouble();
    fenceTable(fenceLength);
  }
  if (cause == bDisplay)
  { fenceLength = gLength.getDouble();
    fencePlot(fenceLength);
  }
}
```

Fence.3

```
void fenceTable(double fenceLength)
{
    double area, x, y;
    x = 0.0;
    y = fenceLength - 2.0 * x;
    mResults.setText("Fence Optimization Table\n");
    while (y >= 0.0)
    {
        area = x * y;
        mResults.append("x = " + x + "    y = " + y +
                       "    area = " + area + "\n");

        x = x + 1.0;
        y = fenceLength - 2.0 * x;
    }
}
```

Fence.4

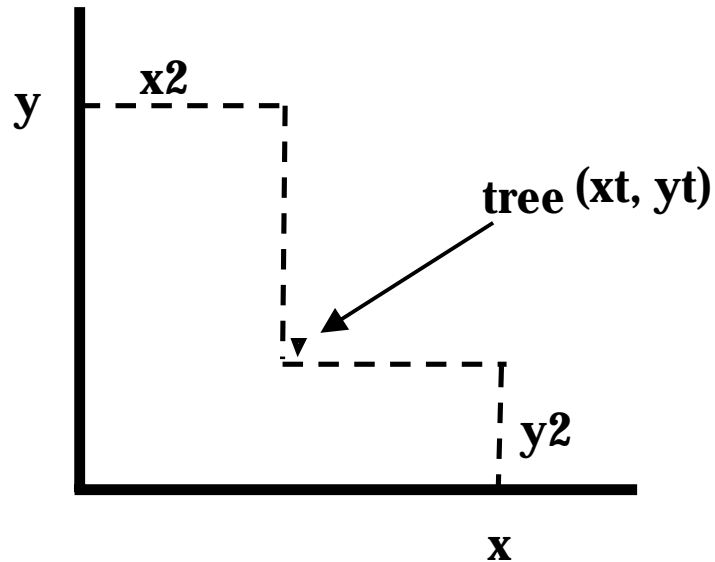
```
void fencePlot(double fenceLength)
{ double area, x, y;
  x = 0.0;
  y = fenceLength - 2.0 * x;
  mResults.setText("Fence Optimization Plot\n");
  while (y >= 0.0)
  { area = x * y;
    mResults.append(x+"\t"+plotString(area)+"\n");
    x = x + 1.0;
    y = fenceLength - 2.0 * x;
  }
}

String plotString(double area)
{ String s = "";
  while (area > 0) { s = s + "*"; area = area - 5.0;}
  return s;
}
}
```

Fence Optimization

- **Output makes it clear how fence should be arranged**
 - **Not necessarily intuitive (make simulation useful)**
 - **There are other tricks (non computer) to get answer**
- **Note we eyeballed the output to get answer**
 - **Could have had computer pick the maximum area**
 - **Could you sketch that program out?**
- **Let's use slightly different approach; answer not obvious**
 - **Fix area**
 - **Minimize amount of fencing used**
 - **Change scenario a bit**
 - **Build into corner**
 - **Put in a tree!**

Fence with Tree



- **Program a bit more complicated**
 - **Will not go over details**
 - **However, intuitive methods not likely to work**
 - **Must use program to get right answer**
 - **Program is on line**

Pitfalls in Automatic Methods

- **Optimization problems seem straightforward enough**
 - **Not always the case**
- **May involve many variables**
 - **Exhaustively checking all possible values may take too long**
 - **Need to *intelligently* look for optimal solution**
 - **However, can have local maxima or minima**
 - **Can lead to wrong answer**
- **Sometimes optimal Solution is computationally out of reach**
 - **Will come back to that theme at end of semester**

Simulation in Microelectronics

- **Modern microchips too complicated to be build without simulation**
 - **It take computers to build computers (recursion?)**
- **One chip takes tens of thousands of dollar to make**
 - **Additional ones are almost free**
 - **One error and it's useless**
- **Each much too complex to check by hand**
 - **Modern chips have millions of transistors**
- **Every aspect of the process is simulated**
 - **Logic**
 - **Layout**
 - **Circuit characteristics**
 - **Fabrication Process**
- **Sometimes optimal solution is computationally *out of reach***
 - **Will come back to that theme at end of semester**

Other Popular Simulation Targets

- **Games that are simulations**
 - **SimCity**
 - **Flight Simulator**
 - **Often serious tools**
- **Graphics**
 - **Many modern movies use computer graphics**
 - **Some entirely graphics**
 - **UNC Computer Science Walk-through**
- **Virtual Reality**
- **Drug Design**
 - **and the list goes one . . .**