Computation as an Expressive Medium

Lab 5: More Cubbies, This and That

Joshua Cuneo
Agenda?

- Announcements
- ArrayLists
- Java API
- super and this
- Project 2 Questions
ArrayLists

- A different kind of array

Clark  Bruce  Diana  Peter
Simple Example - ArrayList

class Point
{
    int x, y;

    Point(int x, int y)
    {
        this.x = x;
        this.y = y;
    }
}
Simple Example - ArrayList

ArrayList pointList = new ArrayList();
Simple Example - ArrayList

Point p1 = new Point(45, 50);
pointList.add(p1);
Simple Example - ArrayList

Point p2 = new Point(79, 23);
pointList.add(p2);
Simple Example - ArrayList

Point p1 = new Point(45, 50);
pointList.add(p1);

Point p2 = new Point(79, 23);
pointList.add(p2);

OR

pointList.add(new Point(45, 50));
pointList.add(new Point(79, 23));
Simple Example - ArrayList

3. Store the Point in Point variable p2

Point p2 = (Point)pointList.get(1);

1. Get whatever’s in index 1
2. That whatever is a Point

new Point(45, 50);  new Point(79, 23);
Simple Example - ArrayList

```java
ellipse(p2.x, p2.y, 10, 10);
```

79 23
void mouseReleased()
{
    pointList.add(new Point(mouseX, mouseY));
}

Simple Example - ArrayList
void draw()
{
    background(255);
    if(pointList.size() > 1)
    {
        beginShape(POLYGON);
        for(int i = 0; i < pointList.size(); i++)
        {
            Point p1 = (Point)pointList.get(i);
            vertex(p1.x, p1.y);
        }
        endShape();
    }
}

Every time we click, the polygon will re-draw, using the new point we've added.
ArrayLists

- **Pros:**
  - Changes size

- **Cons:**
  - Only stores and returns objects, not primitives (int, float, etc.)
What a pain!
ArrayLists

ArrayLists are NOT in the Processing API

http://java.sun.com/j2se/1.4.2/docs/api/
class Cookie {
    int numChips;

    Cookie(int chips) {
        numChips = chips;
    }

    void drawCookie() {
        ellipse(numChips, numChips, 10, 10);
    }
}
class ChocolateCookie extends Cookie
{
    int amtChocolate;

    ChocolateCookie(int amt)
    {
        super(amt);
        amtChocolate = amt;
    }

    void drawCookie()
    {
        fill(188, 143, 143);
        super.drawCookie();
    }
}
class ChocolateCookie extends Cookie
{
    int amtChocolate;

    ChocolateCookie(int amt)
    {
        super(amt);
        amtChocolate = amt;
    }

    void drawCookie()
    {
        fill(188, 143, 143);
        super.drawCookie();
    }
}

class Cookie
{
    int numChips;

    Cookie(int chips)
    {
        numChips = chips;
    }

    void drawCookie()
    {
        ellipse(numChips, numChips, 10, 10);
    }
}
class ChocolateCookie extends Cookie
{
    int amtChocolate;

    ChocolateCookie(int amt)
    {
        super(amt);
        amtChocolate = amt;
        //this.drawCookie();
    }

    void drawCookie()
    {
        fill(188, 143, 143);
        super.drawCookie();
    }
}
Project 2

The contemporary computer scene is dominated by the graphical user interface (GUI). For almost every task, from manipulating text, imagery, sound or video to configuring a computer's operating system (e.g. control panels), from searching for and organizing information (e.g. the web) to the process of programming (e.g. integrated development environments), there are special purpose GUI tools supporting the task through analogies to embodied interaction with physical objects. But no tool is neutral; every tool bears the marks of the historical process of its creation, literally encoding the biases, dreams, and political realities of its creators, offering affordances for some interactions while making other interactions difficult or impossible to perform or even conceive. While the ability to program does not bring absolute freedom (you can never step outside of culture, and of course programming languages are themselves tools embedded in culture), it does open up a region of free play, allowing the artist to climb up and down the dizzying tower of abstraction and encode her own biases, dreams and political realities. What graphical tools would you create? Create your own drawing tool, emphasizing algorithmic generation/modification/manipulation. Explore the balance of control between the tool and the person using the tool. The tool should do something different when moving vs. dragging (moving with the mouse button down). The code for your tool should use at least one class.
Project 2: Translation

- Create an awesome drawing tool
- Give the user some control and the computer some control
- Make it do something different with moving vs. dragging