Computation as an Expressive Medium

Lab 2: Kindergarten Cubbies, Back to the Future and Lego Mania

Joshua Cuneo
This Week

- Assignment 1
- Arrays
- Time
- Casting
- Reading code
- Project 1
Assignment 1

A1-04: Control the position of two lines with one variable.
A1-05: Control the position and size of two lines with two variables.
A1-06: Control the properties of two shapes with two variables.
A1-07: Create a simple, regular pattern with six lines.
A1-08: Program your pattern from Assignment 1-07 using while().
A1-10: Redo Assignment 1-05 using mouseX and mouseY as the variables.
A1-11: Draw two visual elements that each move in relation to the mouse in a different way.
A1-12: Draw three visual elements that each move in relation to the mouse in a different way.
A1-13: Move a visual element across the screen. When it disappears off the edge, move it back into the frame.
A1-14: Draw a visual element that moves in relation to the mouse but with a different relation when the mouse is pressed.
A1-15: Using if and else, make the mouse perform different actions when in different parts of the window.
A1-16: Develop a kinetic image which responds to the mouse.
A Few Reminders

- We know you are not programmers
- Don’t get too caught up in syntax
- The API is your friend
- Have fun with your assignments
Brian
- Class format
- Class grades
- Reading/presentations
- Due dates
- I love my TA

Joshua
- Lab format
- Assignment grades
- Programming assignments
- I love my professor
Arrays
Arrays

Clark  Bruce  Diana  Peter

cubbies
Arrays

cubbies[Bruce]
Arrays

```java
int[] numbers = new int[3];

numbers[0] = 90;
nnumbers[1] = 150;
nnumbers[2] = 30;
```

OR

```java
int[] numbers = {90, 150, 30};
```
Arrays

Clark  Bruce  Diana  Peter

Clark  Bruce  Diana  Peter
Arrays

Clark  Bruce  Diana  Peter

*Some exceptions apply
Why Do We Care?

- Concise
- Efficient
- Loop-able
void bubblesort(int[] input)
{
    int temp;
    for(int i = 0; i < input.length; i++)
    {
        for(int j = 0; j < input.length - 1; j++)
        {
            if(input[j] > input[j+1])
            {
                temp = input[j+1];
                input[j+1] = input[j];
                input[j] = temp;
            //end if
        } //end for(j)
    } //end for(i)
} //end for(i)
Processing API

- append()
- arrayCopy()
- concat()
- expand()
- reverse()

- shorten()
- sort()
- splice()
- subset()
//Retrieving array values
int a = numbers[0] + numbers[1]; // Sets variable a to 240
int b = numbers[1] + numbers[2]; // Sets variable b to 180

//Changing array size at runtime
String[] sal = { "OH ", "NY ", "CA "};
String[] sa2 = append(sal, "MA ");
println(sa2); // Prints OH, NY, CA, MA
Time

- `int hour()`
- `int minute()`
- `int second()`
- `int day()`
- `int month()`
- `int year()`
- `long millis()`
Time

EXAMPLE
Casting

```c
int x = 3;
int y = x/2;  // y = 1?????

int x = 3;
float y = (float)x/2;  // y = 1.5
```
void setup()
{
    size(400, 400);
}

int ellipseSize = 1;

void draw()
{
    smooth();
    background(0);
    stroke(234, 98, 71);
    strokeWeight(10); fill(0); ellipse(200, 200, ellipseSize, ellipseSize);

    if ((mouseX > 0) && (mouseX < 400) && (mouseY > 0) && (mouseY < 200))
    {
        ellipseSize = ellipseSize + 1;
    }
    else
    {
        ellipseSize = ellipseSize - 1;
    }

    if (ellipseSize < 1)
    {
        ellipseSize = 1;
    }

    if (ellipseSize > 400)
    {
        ellipseSize = 400;
    }
}
void setup()
{
    size(400, 400);
}

int ellipseSize = 1;

void draw()
{
    smooth();
    background(0);
    stroke(234, 98, 71);
    strokeWeight(10); fill(0); ellipse(200, 200, ellipseSize, ellipseSize);

    if ((mouseX > 0) && (mouseX < 400) && (mouseY > 0) && (mouseY < 200))
    {
        ellipseSize = ellipseSize + 1;
    }
    else
    {
        ellipseSize = ellipseSize - 1;
    }

    if (ellipseSize < 1)
    {
        ellipseSize = 1;
    }

    if (ellipseSize > 400)
    {
        ellipseSize = 400;
    }
}
void setup()
{
    size(400, 400);
}

int ellipseSize = 1;

void draw()
{
    smooth();
    background(0);
    stroke(234, 98, 71);
    strokeWeight(10); fill(0); ellipse(200, 200, ellipseSize, ellipseSize);

    if ((mouseX > 0) && (mouseX < 400) && (mouseY > 0) && (mouseY < 200))
    {
        ellipseSize = ellipseSize + 1;
    }
    else
    {
        ellipseSize = ellipseSize - 1;
    }

    if (ellipseSize < 1)
    {
        ellipseSize = 1;
    }

    if (ellipseSize > 400)
    {
        ellipseSize = 400;
    }
}
void setup()
{
  size(400, 400);
}

int ellipseSize = 1;

void draw()
{
  smooth();
  background(0);
  stroke(234, 98, 71);
  strokeWeight(10); fill(0); ellipse(200, 200, ellipseSize, ellipseSize);

  if ((mouseX > 0) && (mouseX < 400) && (mouseY > 0) && (mouseY < 200))
  {
    ellipseSize = ellipseSize + 1;
  }
  else
  {
    ellipseSize = ellipseSize - 1;
  }

  if (ellipseSize < 1)
  {
    ellipseSize = 1;
  }

  if (ellipseSize > 400)
  {
    ellipseSize = 400;
  }
}
void setup()
{
    size(400, 400);
}

int ellipseSize = 1;

void draw()
{
    smooth();
    background(0);
    stroke(234, 98, 71);
    strokeWeight(10); fill(0); ellipse(200, 200, ellipseSize, ellipseSize);

    if ((mouseX > 0) && (mouseX < 400) && (mouseY > 0) && (mouseY < 200))
    {
        ellipseSize = ellipseSize + 1;
    }
    else
    {
        ellipseSize = ellipseSize - 1;
    }

    if (ellipseSize < 1)
    {
        ellipseSize = 1;
    }

    if (ellipseSize > 400)
    {
        ellipseSize = 400;
    }
}
Reading Code

- One line at a time
- Read comments
- Draw pictures
- Run pieces of code
- Go screaming to your TA
Project 1

From the central heartbeat of the central processor, to the obsessive timestamping of files and blog entries, to ever present clock displays, time is a fundamental feature of computation. Display the progress of time in a non-traditional way. It is OK to consider large temporal scales (e.g. seasons), but smaller temporal scales should also be displayed (or be available to be displayed, perhaps as a function of user input). You may make use of mouse input if you wish.
From Idea to Computation
From Idea to Computation
From Idea to Computation
1. Create your design
2. Break it down into functional components
3. Code each component
4. Fit the components together as you go
5. Modify your idea if it becomes too difficult
6. Consult your friendly neighborhood TA
More Useful Tools

- Pimage
- beginShape()
- endShape()
- vertex()
- stroke()
- fill()