Midterm
If you are in here, you had better step up your game.

Announcements
- No sections next Monday. Go to sections today and tomorrow if you can.
- Creativity in Processing
  - Due next Tuesday. Should have started by now.
  - Get a programming partner! Go Crazy!
- Today: Functions, For Loops, Some creativity stuff
- Next week

Walking

Why Pair Programming?
- Having a partner gives you a built-in helper
- Many people find computers more fun if they get to work with someone else
- Learning to work with computers is like learning a foreign language, lots of new words and codes
- => Easier to learn a language if you have some to ‘converse with’ in the language
- Working with a partner helps you develop good teamwork skills
- Most programming projects in real life done in teams
- Employers look for people who are good at teamwork

HW9: Creativity in Processing
Assign. Write your program to do whatever you want (but don’t copy the examples above), and try to make them clever or interesting or cute or have some property that would interest a viewer. You should try to use those you have learned in your former Processing homework, because those are the basics and one goal of this assignment is to practice the basics. But, if you need some other feature of Processing that you find in the reference page go ahead and use it. The goal is creativity ... but don’t spend forever on it either.

A creativity program from last year
What can we do with Processing we haven’t done yet?

If we generate it with processing, is it art?

Mondrian, Pollock, Albers are stars ... 

Art In A Click

- Computer art, that is, art generated by computers, not art created by people using computers, leads to some fun Web sites.
- Google “piet mondrian”. He was a cubist and created pictures that look like this http://www.google.com/search?client=safarir&rls=en&q=“Piet +Mondrian”&ie=UTF-8&oe=UTF-8
- Could we generate this in processing?

How about something a little easier first?

Let’s Try it!

What would we do in Processing? What does Random do?
Random Numbers

- Random numbers should be called random number sequences, because the definition requires that no matter how many numbers you already know in the sequence, it’s not possible to predict the next one. A non-random sequence is 2, 4, 6, 8, 10, …
- Computers cannot produce random numbers (because computers are completely predictable), but they can produce a sequence of numbers that passes all of the tests for randomness. These are called pseudo-random numbers, but everyone drops the “pseudo” part.
- To generate a random number in Processing we write:
  
  `random(<smallest possible number>, <largest possible number>).`

  We get back a number – we can’t predict which – between the two limits, including the end points.

  To generate a random number between 0 and 255, write `random(0, 255).`

  To generate a number between 0 and 1, write `random(0, 1).`
System Variables (Processing)

- mouseX, mouseY
- pmouseX, pmouseY
- width, height
- frameCount

void setup()
{
  size(400, 400); // position A
}

void draw()
{
  fill(255-abs(mouseX-pmouseX));
  rect(pmouseX, pmouseY, mouseX, mouseY); // position B
}

Where should the line "background(255);" be placed so that the sketch shows just a single moving rectangle?
Choose option E if it would work with either C or D.

Placement_click_3

void setup()
{
  size(400, 400);
  //background(255);
}

void draw()
{
  background(255,8,8);
  fill(255-abs(mouseX-pmouseX));
  rect(pmouseX, pmouseY, mouseX, mouseY);
  //background(255);
}

Arty version of Placement. arty_placement

Plant r abi
color newColor;

void setup()
{
  size(400, 400);
  background(320);
  frameRate(20);
}

void draw()
{
  r = random(8,255);
  g = random(8,255);
  b = random(8,255);
  newColor = color(r,g,b);
  fill(newColor-orrround(mouseX));
  rect(pmouseX, pmouseY, mouseX, mouseY);
}

Pmouse

pmouseX

// Move the mouse quickly to see the difference in position
// between the current and previous position

Placement_click_3

Another arty version. arty_placement_v2
Processing_review_ball_click5

```cpp
int ballX, ballDia = 50;
void setup() {
  size(400, 400);
  ballX = -ballDia/2;
}
void draw() {
  if (ballX > width+ballDia/2) ballX = -ballDia/2;
  ellipse(ballX, height/2, ballDia, ballDia);
  ballX = ballX + 1;
}
```

arty_ball_click5

A. Ball moves across jumping back to left edge and changing color as soon as it touches the right edge.
B. Ball moves across as one color until it moves off the right edge then changes color and moves back in from the left edge.
C. Ball moves across until halfway off the right edge (showing just a half circle) then changes color.
D. Ball moves across then disappears and doesn’t come back.

arty_ball_click5

```cpp
void draw() {
  if (ballX > width+ballDia/2) {
    ballX = -ballDia/2;
    fill(newColor);
    ellipse(ballX, height/2, ballDia, ballDia);
    ballX = ballX + 1;
  }
}
```

POINTILISM

- [http://www.processing.org/learning/pixels/](http://www.processing.org/learning/pixels/)

Transparency, Tints and More!

- [http://www.processing.org/learning/pixels/](http://www.processing.org/learning/pixels/)

How about Jackson Pollock?

- Google “jackson pollock”.
- He was an abstract expressionist and created paintings that look like this!

Circa 1950
Computer generated Jackson Pollock?

- I made this on this website.
- http://www.jacksonpollock.org

Can we do that in processing? jackson_pollack

```java
void setup() {
  size(500, 500);
  intro();

  // I made this on this website.
  http://www.jacksonpollock.org
}
```

What does this display?

A) many nested circles (like a target) changing color and growing in size
B) a white circle growing in size
C) a black circle growing in size
D) pulsing images of circles growing and changing color each time
E) nothing (a circle with diameter 0)

Arty_ellipse_click1

```java
void setup() {
  size(500, 500);
  intro();

  // I made this on this website.
  http://www.jacksonpollock.org
}
```

What does this display?

A) many nested circles (like a target) changing color and growing in size
B) a white circle growing in size
C) a black circle growing in size
D) pulsing images of circles growing and changing color each time
E) nothing (a circle with diameter 0)

Arty_ellipse_click1

```java
void setup() {
  size(500, 500);
  intro();

  // I made this on this website.
  http://www.jacksonpollock.org
}
```

What does this display?

A) many nested circles (like a target) changing color and growing in size
B) a white circle growing in size
C) a black circle growing in size
D) pulsing images of circles growing and changing color each time
E) nothing (a circle with diameter 0)

http://processing.org/learning/basics/redraw.html
What does this display?

A) many nested circles (like a target) changing color and growing in size
B) a white circle growing in size
C) circle changes color and grows as well at each mouse click
D) pulsing images of circles changing color and growing at the frame rate

---

Two ways

- For Loops
- Functions

---

Homework 10: Functions

- void cell(int x, int y, int s, color tint)
- void triple(int x, int y, int s, color tint)
- Also functions for
  - block()
  - row()
  - cellarray()

---

For loops (Repetition)

- Repeating commands is powerful:
  - Lightbot 2.0 used recursion, a function calling itself
  - Symbolic Lightbot prefixed a number, a:Step
- Processing uses a for loop:
  
  ```
  void setup() {
    size(500,200);
    background(0);
    smooth();
    fill(255);
    for (int i = 0; i < 16; i++) {
      ellipse(100+25*i, 100, 15, 15);
    }
  }
  ```

---

Don’t forget about Open Processing
# Repetition, the Picture

- A for loop has several parts, all required ...

```
for (int i = 0; i < 16; ++i) {
    ellipse(100+25*i, 100, 15, 15);
}
```

The result of this statement is 16 copies of the stuff to be repeated. 16 Pacman pills

# For loops: what about with redraw?

```
def midi():
    background(0)
    fill(0,0,255)
    for i = 0; i < 5; i = i + 1 {
        fill(0, i*255, 0)
        ellipse(100,100, 100-(20*i),100-(20*i));
    }
```

What does this display?
A) Lots of little pills going off the left and right side of the screen
B) Rows of little pills repeating with every mouse click
C) A row of little pills that change color with every mouse click
D) Pulsing images of circles changing color and growing at the frame rate

# For loops. What about with redraw?

```
for (int i = 0; i < 16; ++i) {
    ellipse(100+25*i, 100, 15, 15);
}
```

# Repetition, Another Picture

- Or how about a bullseye?

```
int i;
size(200,200);
background(0);
fill(0,0,255);
//fill(0,255,0);
//fill(255,0,0);
strokeWeight(4);
for (i = 0; i < 5; i = i + 1) {
    fill(0, i*255, 0)
    ellipse(100,100, 100-(20*i),100-(20*i));
}
```

Note the loop variable must be declared ... but could do it in loop itself like we did for pacman pills:
```
for (int i = 0; ...
What would functions let us do?

Functions abstract by packaging computation

Example: Abstraction in Everyday Life

Example of Functional Abstraction

Abstracting

Functions abstract by packaging Computation

Instructions Formed of Simpler Instructions

Example: Abstraction in Everyday Life

Abstracting

Functions abstract by packaging Computation
The Function Becomes A Concept

- Because F1 processes a riser, I think of the programming task as:
  - Process a riser F1
  - Move to next riser
  - Process a riser F1
  - Move to next riser
  - Process a riser F1

- With F1 as a concept, I simplify the programming to just 5 steps rather than 21.
- It also suggests another concept:
  - Move_to_next_riser()

Functions In Processing

- Form of function definition in Processing:
  ```
  <return type> <name> (<param list>) {  
  <body>  
  }
  ```

- As in:
  ```
  void draw_a_box(int x_pos, int y_pos) {  
  rect(x_pos, y_pos, 20, 20);  
  }
  ```

- Or:
  ```
  color pink() {  
  return color(255, 200, 200);  
  }
  ```

Functions In Processing: Result

- Functions that do something, but do not return a value, have void as their <return type>.
- Functions that return a value must say its type.

```
Functions In Processing: Params

- Parameters are the values used as input to the function; parameters are not required, but the parentheses are.
- The type of each parameter must be given.

```java
void draw_a_box(int x_pos, int y_pos) {
    rect(x_pos, y_pos, 20, 20);
    color pink();
    return color(255, 200, 200);
}
```

More On Parameters …

**Parameters:** Customize each function call to a specific situation — they are the input to the function.
- **Parameters** are the names of the input values used inside of the procedure body.
- **Arguments** are the values from outside to be used for each of the parameters.

Arguments Become Parameters

- Notice that if the DEFINITION has $n$ parameters, the CALL needs $n$ arguments.
- The parameters and arguments correspond.

Inside of the function, the parameter, e.g. $xloc$, is declared and initialized to the corresponding argument (width*25). Then, the definition uses it, e.g. `ellipse(xloc, yloc, ...)`.

Functions In Processing: Return

- A function returns its value with the `return` statement … the stuff following return is the result.
- The function is done when it reaches return.

```java
void draw_a_box(int x_pos, int y_pos) {
    rect(x_pos, y_pos, 20, 20);
    color pink();
    return color(255, 200, 200);
}
```

Recursive Functions. See examples on Open Processing and in processing.org tutorials.

Sierpinski: function call when mouse pressed
Layering: Building Functions out of Functions. Next week.

Functional Abstraction Reduces Complexity

Homework 10: Functions

- void cell(int x, int y, int s, color tinto)
- void triple(int x, int y, int s, color tinto)
- Also functions for
  - block()
  - row()
  - cellarray()