Review

- What is Computing?
- What can be Programmed?
- Creative Computing
- Processing
- Downloading Processing
- Dropbox

- Primitive Shapes
  - point
  - line
  - triangle
  - quad
  - rect
  - ellipse
- Processing Canvas
- Coordinate System
- Shape Formatting
  - Colors
  - Stroke
  - Fill

random\( (\text{high}) \); random\( (\text{low, high}) \);
Generate a random number in the range \text{low} (or 0) to \text{high}

mouseX
mouseY
Built-in predefined variables that hold the current mouse X and Y locations

print( \text{something} );
println( \text{something} );
Print something to the Processing console.

void setup()
{
  // Called once when program starts
}

void draw()
{
  /* Called repeatedly
   * while program runs */
}

randomEllipse

void setup()
{
  size(300, 300);
  smooth();
}

void draw()
{
  fill(random(255), random(255), random(255));
  ellipse(mouseX, mouseY, 30, 30);
}

Controlling the draw loop

frameRate(\text{fps});
Sets number of frames displayed per second. i.e. the number of times draw() is called per second. Default = 60.

noLoop();
Stops continuously calling draw().

loop();
Resumes calling draw().

More Graphics

arc(\ldots)
curve (\ldots)
bézier(\ldots)
shape(\ldots)
Arcs

arc( x, y, width, height, start, stop );

An arc is a section of an ellipse

x, y, width, height
location and size of the ellipse
start, stop
arc bounding angles (in radians)

Spline Curves

curve( x1, y1, x2, y2, x3, y3, x4, y4 );

Spline: A smooth line drawn through a series of points
A curve is a Catmull-Rom (cubic Hermite) spline defined by four points
x2, y2 and x3, y3
beginning/end points of visual part of curve
x1, y1 and x4, y4
control points that define curve curvature

Bézier Curves

bezier( x1, y1, cx1, cy1, cx2, cy2, x2, y2 );

A smooth curve defined by two anchor points and two control points
x1, y1 and x2, y2
anchor points of bézier curve
cx1, cy1 and cx2, cy2
control points that define curvature
Custom Shapes

- Composed of a series of vertexes (points)
- Vertexes may or may not be connected with lines
- Lines may join at vertexes in a variety of manners
- Lines may be straight, curves, or bézier splines
- Shape may be closed or open

beginShape( [option] );
vertex( x, y );
curveVertex( x, y );
bezierVertex( cx1, cy1, cx2, cy2, x, y );
endShape( [CLOSE] );

strokeJoin()

noFill();
smooth();
strokeWeight(10.0);
strokeJoin(MITER);
beginShape();
vertex(35, 20);
vertex(65, 50);
vertex(35, 80);
endShape();

noFill();
smooth();
strokeWeight(10.0);
strokeJoin(BEVEL);
beginShape();
vertex(35, 20);
vertex(65, 50);
vertex(65, 50);
vertex(35, 80);
endShape();

noFill();
smooth();
strokeWeight(10.0);
strokeJoin(ROUND);
beginShape();
vertex(35, 20);
vertex(65, 50);
vertex(65, 50);
vertex(35, 80);
endShape();

More Color

colorMode(RGB or HSB);

RGB: (red, green, blue)

HSB:
  hue
    • "pure color"
  saturation
    • "intensity"
  brightness
    • "lightness"
void mousePressed() {
    // Called when the mouse is pressed
}

void mouseReleased() {
    // Called when the mouse is released
}

void mouseClicked() {
    // Called when the mouse is pressed and released
    // at the same mouse position
}

void mouseMoved() {
    // Called while the mouse is being moved
    // with the mouse button released
}

void mouseDragged() {
    // Called while the mouse is being moved
    // with the mouse button pressed
}

void keyPressed() {
    // Called each time a key is pressed
}

void keyReleased() {
    // Called each time a key is released
}

void keyTyped() {
    // Called when a key is pressed
    // Called repeatedly if the key is held down
}

keyCode vs. key
key
  – A built-in variable that holds the character that was just
typed at the keyboard
keyCode
  – A built-in variable that holds the code for the keyboard key
that was touched

All built-in keyboard interaction functions ...
  • Set keyCode to the integer that codes for the keyboard key
  • Set key to the character typed
  • All keyboard keys have a keyCode value
  • Not all have a key value

Example Sketches...
  – LadyBug1
  – Monster1
  – Ndebele
  – Penguin1
  – SouthParkCharacter1
  – Sushi
  – GiorgioMorandi

OpenProcessing
http://www.openprocessing.org/
  – Bryn Mawr and SMU student sketches