Obamicon

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

Drawing Primitives
- point\((x, y)\);
- line\((x_1, y_1, x_2, y_2)\);
- triangle\((x_1, y_1, x_2, y_2, x_3, y_3)\);
- rect\((x, y, \text{width}, \text{height})\);
- ellipse\((x, y, \text{width}, \text{height})\);

Modes
- rect\((x, y, \text{width}, \text{height})\);
- ellipse\((x, y, \text{width}, \text{height})\);
- rectMode\((\text{CENTER})\);
- ellipseMode\((\text{CORNER})\);

Programming Principle
- Syntax is important!
- Function name
- Parentheses
- Arguments
- Statement terminator

Odds and Ends
- Processing programs carry the extension \texttt{.pde}
- must be in a folder with the same name
- \texttt{myProgram.pde} must be inside a folder called \texttt{myProgram}
- Code block
  - The curly braces {}
- Comments
  - //
  - /* and */
- Naming convention
Basic Processing Program

```java
void setup() {
  // Called once when program starts
}
void draw() {  
  /* Called repeatedly
     while program runs */
}
```

The Event Loop

- Any code in `draw()` is executed 60 times per second
- Put code that you only want executed once in `setup()`
  – defaults
- `noLoop()`
- `loop()`

Mouse Interaction

- Built-in predefined variables that hold the mouse X and Y locations
  - current `mouseX` `mouseY`
  - previous (last) `pmouseX` `pmouseY`
  - 0 if mouse is not in window

More Graphics Primitives

- `arc(...)`
- `curve(...)`
- `bezier(...)`
- `shape(...)`

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

\begin{equation}
\text{curve}(x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4);
\end{equation}

\textit{spline}: A smooth curve drawn defined by four points

\begin{itemize}
  \item \(x_2, y_2 \text{ and } x_3, y_3\) \hspace{1cm} \text{beginning/end points of visual part of curve}
  \item \(x_1, y_1 \text{ and } x_4, y_4\) \hspace{1cm} \text{control points that define curve curvature}
\end{itemize}

Bézier Curves

\begin{equation}
\text{bezier}(x_1, y_1, cx_1, cy_1, cx_2, cy_2, x_2, y_2);
\end{equation}

A smooth curve defined by two \textit{anchor points} and \textit{two control points}

\begin{itemize}
  \item \(x_1, y_1 \text{ and } x_2, y_2\) \hspace{1cm} \text{anchor points of bézier curve}
  \item \(cx_1, cy_1 \text{ and } cx_2, cy_2\) \hspace{1cm} \text{control points that define curvature}
\end{itemize}

Custom Shapes

- Composed of a series of vertexes (points)
  - Vertices 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

\begin{verbatim}
beginShape([option]);
  vertex(x, y);
  curveVertex(x, y);
  bezierVertex(cx1, cy1, cx2, cy2, x, y);
endShape([CLOSE]);
\end{verbatim}
void keyTyped() {
    // Called when a key is pressed and immediately released.
}

void keyPressed() {
    // Called when a key is pressed and held down.
}

void keyReleased() {
    // Called when the key is released.
}

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 moved.
}

void mouseDragged() {
    // Called while the mouse is being moved.
}

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...
- LadyBug
- Monster
- Ndebele
- Penguin
- SouthParkCharacter
- Sushi
- GiorgioMorandi