

# Processing Paint Program Tutorial

## 1. Drawing lines

The most basic paint program we can make using processing simply draws lines on the canvas. This can be implemented with a relatively small amount of code:

**File: Paint1**

```
float oldX = 0;
float oldY = 0;

color black= color(0);

void setup() {

    size(500, 500);
    background(255);
}

void draw() {

    stroke(black);
    strokeWeight(1);

    line(mouseX, mouseY, oldX, oldY);

    oldX = mouseX;
    oldY = mouseY;
}
```

As we will be drawing using lines and lines use a start point and an end point, we will first need variables to hold the old mouse position, these are oldX and oldY, the line is always drawn to the current mouse position, which can be found by the variables mouseX and mouseY.

We also define a colour/color, black, using the color() function and setting the value to 0.

We then set up our processing sketch using the setup function. We set the canvas size to 500, and the background to 255, which is the colour white.

Then to actually draw anything we create our draw function, we set the stroke to the black colour and the stroke weight to 1.

We can then draw our line using the line function with the relevant X and Y coordinates.

Finally we set the old values of X and Y to the current values of X and Y.

There is a problem here though, the program is always drawing a line and we have no way of controlling when it starts or stops. In order to fix this we need to use a conditional.

## 2. Adding mouse click control

We can add mouse click control using a conditional statement, the 'if' statement like this:

**File: Paint2**

```
float oldX = 0;
float oldY = 0;

color black= color(0);

void setup() {

    size(500, 500);
    background(255);
}

void draw() {

    stroke(black);
    strokeWeight(1);

    if (mousePressed == true) {

        line(mouseX, mouseY, oldX, oldY);
    }

    oldX=mouseX;
    oldY=mouseY;
}
```

Now when we press the left mouse button, it will change the value of mousePressed to true. In a conditional statement, when the condition is true, the code inside the curly braces is executed, in this case the line() function.

## 3. Changing the stroke weight

Now we can draw on our canvas with a certain degree of control, but it would be nice to have a few more features, in most paint programs one can control the size/weight of the paint brush and change the colours. Lets do that!

We are first going to let users change the weight of the paint stroke using the '+' and '-' keys on the keyboard.

To do this we need to implement the keyPressed function, which, you guessed it, listens for key presses. Lets see what we need to do:

## File: Paint3

```
float oldX = 0;
float oldY = 0;

color black= color(0);

int size = 10;

void setup() {

    size(500, 500);
    background(255);
}

void draw() {

    stroke(black);
    strokeWeight(size);

    if (mousePressed == true) {

        line(mouseX, mouseY, oldX, oldY);
    }

    oldX=mouseX;
    oldY=mouseY;
}

void keyPressed() {

    if (key == '+') {

        size++;
    }
    if (key == '-') {

        size--;
    }
}
```

Here we have done a number of things, first we created a variable called 'size', which will hold the size of the paint brush.

You can see in the draw() function we have replaced the constant value 1, with the variable 'size' in the strokeWeight function, this means that when we change the value of size, the stroke weight will change as well.

We receive key presses inside the keyPressed function. In here we have two conditionals, if the key is an '+' key, we add one to the size variable using the short hand size++ ( this is equivalent to size = size + 1), if the key is the '-' key, we take away one using the size-- expression.

There is a problem here though, there's nothing stopping us from making the stroke size ridiculously large, or a negative number, which doesn't even make sense. So we need to stop that. Here is an improved version of the keyPressed function.

```
void keyPressed() {  
    if (key == '+' && size < 30) {  
        size++;  
    }  
    if (key == '-' && size > 0) {  
        size--;  
    }  
}
```

This is much better, using the && operator inside the conditional, which basically means 'and', we are now saying if the key pressed is the '+' symbol AND the size is less than 30, you can make the size bigger, we are also saying if the key pressed is '-' AND the size is greater than zero we can make it smaller. Problem solved.

Unfortunately though we have no visual indicator of the stroke size other than when we use the paint brush, it would be nice to be able to see the stroke size before we use it, let's implement that.

#### File: Paint4

```
float oldX = 0;  
float oldY = 0;  
  
color black = color(0);  
  
int size = 10;  
  
void setup() {  
    size(500, 500);  
    background(255);  
}  
  
void draw() {  
    stroke(black);  
    strokeWeight(size);  
  
    fill(black);  
    ellipse(width - 50, 40, size, size);  
  
    strokeWeight(size);  
  
    if (mousePressed == true) {
```

```

    line(mouseX, mouseY, oldX, oldY);
}

oldX=mouseX;
oldY=mouseY;
}

void keyPressed() {

    if (key == '+' && size < 30) {

        size++;
    }
    if (key == '-' && size > 0) {

        size--;
    }
}

```

If you look at the draw function you can see that after we set the stroke colour and the strokeWeight, we have a new function to set the fill colour to black, and then we draw an ellipse 40 pixels from the right of the screen, 40 pixels down and with a size of the stroke weight.

There is a problem here though, we can see the stroke increasing in weight but when we make the stroke smaller we can't see it on the canvas, why?

It's because there is nothing clearing the canvas after each time the draw function is called, this is good, because then we wouldn't be able to draw anything, but for things like controls, it makes it so we are unable to see some changes in them like in this case. Lets fix this by clearing just the top of the screen where the stroke weight indicator is.

### File: Paint5

```

float oldX = 0;
float oldY = 0;

color black= color(0);
color white= color(255);
int size = 10;

void setup() {

    size(500, 500);
    background(255);
}

void draw() {

    strokeWeight(1);
    fill(white);
    rect(0, 0, width, 70 );
}

```

```

stroke(black);
strokeWeight(size);

fill(black);
ellipse(width - 50, 40, size, size);

strokeWeight(size);

if (mousePressed == true) {
    line(mouseX, mouseY, oldX, oldY);
}

oldX=mouseX;
oldY=mouseY;
}

void keyPressed() {
    if (key == '+' && size < 30) {
        size++;
    }
    if (key == '-' && size > 0) {
        size--;
    }
}

```

We've done 2 main things here, first at the top of the file we have created another colour 'white'. This will be used to fill the top of the page. Next, in the draw function in the first few lines we clear the area. First we set the stroke weight to 1, because we want a small border on the cleared area, next we set the fill colour to white and then draw the rectangle.

If you run the program now you can see the size of the stroke weight decreasing.

## 4. Changing colours

It would be great if we were able to draw with more than one colour, so lets implement that!

We will first create a place for our palette so we may select the colours. Lets make a palette with red and black.

### File: Paint6

```

float oldX = 0;
float oldY = 0;

color black= color(0);
color white= color(255);

```

```

color red = color(255, 0, 0);

int size = 10;

void setup() {

    size(500, 500);
    background(255);
}

void draw() {

    strokeWeight(1);
    fill(white);
    rect(0, 0, width, 70 );

    fill(red);
    rect(10, 10, 25, 25 );

    fill(black);
    rect(10, 35, 25, 25);

    stroke(black);
    strokeWeight(size);

    fill(black);
    ellipse(width - 50, 40, size, size);

    strokeWeight(size);

    if (mousePressed == true) {

        line(mouseX, mouseY, oldX, oldY);
    }

    oldX=mouseX;
    oldY=mouseY;
}

void keyPressed() {

    if (key == '+' && size < 30) {

        size++;
    }
    if (key == '-' && size > 0) {

        size--;
    }
}

```

Here at the top of the file we have created another colour, red. In the draw function you can see that we are now also drawing 2 squares that are red and black after we clear the top of the screen using the fill and rect functions.

This is great, but also useless, we need to be able to change the colours, so lets go ahead and make it useful.

## File: Paint7

```
float oldX = 0;
float oldY = 0;

color black= color(0);
color white= color(255);
color red = color(255, 0, 0);

color selectedColor = black;

int size = 10;

void setup() {

    size(500, 500);
    background(255);
}

void draw() {

    stroke(black);

    strokeWeight(1);
    fill(white);
    rect(0, 0, width, 70 );

    fill(red);
    rect(10, 10, 25, 25 );

    fill(black);
    rect(10, 35, 25, 25);

    fill(selectedColor);
    ellipse(width - 50, 40, size, size);

    strokeWeight(size);

    if (mousePressed) {

        if (mouseX > 10 && mouseX < 35) {

            if (mouseY > 10 && mouseY < 35) {

                selectedColor = red;
            }

            if (mouseY > 35 && mouseY < 60) {

                selectedColor = black;
            }
        }
    }
}
```



```

}

if (mousePressed == true) {

    stroke(selectedColor);
    line(mouseX, mouseY, oldX, oldY);
}

oldX=mouseX;
oldY=mouseY;
}

void keyPressed() {

    if (key == '+' && size < 30) {

        size++;
    }
    if (key == '-' && size > 0) {

        size--;
    }
}

```

So first we have created a variable called `selectedColor`, this variable holds the colour value that we will paint with. Next if we look at the `draw` function we first set the stroke colour to black so the borders of our palette squares and cleared area at the top will be black. Next we draw our palette and stroke weight indicator as normal.

The biggest difference here is the new `if` statement, this statement looks to see if the mouse has clicked over either the red palette square or the black palette square, depending on which one it sets our `selectedColor` variable to the necessary colour which is then used in the final conditional statement of the `draw` function which actually draws our line.

## Challenges

If you managed to complete the program so far can you think of anything else you could add, perhaps more colours, or even changing the shape of the paint brush, make a spray painting effect, try some things out and experiment to see what you can do.....

Try adding additional colours to the palette. You could try doing this by making a function to draw each of your palette tiles and to detect if the mouse has selected the colour. An example can be found in the `PaintPalette` file.

Try adding a palette of different brushes, so instead of just drawing with a circle, try drawing with a square, perhaps you will need to change the preview of the brush size depending on what brush is selected. Along with the various shapes, you could do a spray paint effect, hint. you would use the `random()` function. A proof of concept can be found in the `PaintSpray` sketch.

Try adding more keyboard controls, it would be nice to be able to clear the canvas so we can start a new

drawing, perhaps add a keyboard shortcut for this?