

Chapter 1 : Color

Exercise P.1.1.2

Step 1:

This exercise will focus on a color spectrum in a circle. In the setup function we create an canvas of dimensions 800 x 800 with no stroke.

Step 2:

In the draw function we set the color mode to HSB and give the parametres as 360 for the hue, this is a full spectrum of color in a circle. The saturation is set to width and the brightness is set to height. This is adjusted in such a way that the mouse coordinates can be taken as their values.

```
function draw() {  
  colorMode(HSB, 360, width, height);  
  background(360, 0, height);  
}
```

We create two variables called segmentCount and radius. These two variables are both set to 360. We then create a variable called angleStep. This angle is equalled to 360 and divided by the segment count. This is done as the angle increment depends on how many segments are to be drawn. In this case we want a full circle so we use 360.

```
var angleStep = 360 / segmentCount;
```

Step 3:

Here we are still in the draw function. After we begin the triangle shape we give a vertex the parameters of width/2 and height/2. The first vertex point is in the middle of the display. We use vertices because to create the circle we use a series of fan shapes to make up the full circle. It is important that the first vertex is in the centre of the circle.

```
var angleStep = 360 / segmentCount;
```

Different to the first vertex the rest of them the angle must be converted from degrees to radians. The functions cos and sin require this conversion. The fill color for the next segment is defined by HSB as angle for hue, mouseX for saturation and mouseY for brightness.

```
for (var angle = 0; angle <= 360; angle += angleStep) {  
  var vx = width / 2 + cos(radians(angle)) * radius;  
  var vy = height / 2 + sin(radians(angle)) * radius;  
  vertex(vx, vy);  
  fill(angle, mouseX, mouseY);  
}
```

Step 4:

The final step is to create some keyboard functions to switch keys. The switch command checks the last key pressed and enables easy switching between keys. The segmentCount is varied between each key pressed.

```
function keyPressed() {  
  if (key == 's' || key == 'S') saveCanvas(gd.timestamp(), 'png');  
  
  switch (key) {  
    case '1':  
      segmentCount = 360;  
      break;  
    case '2':  
      segmentCount = 45;  
      break;  
    case '3':  
      segmentCount = 24;  
      break;  
    case '4':  
      segmentCount = 12;  
      break;  
    case '5':  
      segmentCount = 6;  
      break;  
  }  
}
```