Before this assignment I wasn’t very confident in Processing. Even during the tutorials I had a lot of trouble understanding and implementing what I had learnt into the software to develop a picture. I started off playing around with Processing. I drew lines, bezier curves and shapes. I think the main problem I had with processing was that some of the words used in Processing such as norm(); I would automatically think of “the normal” which is the perpendicular. Another thing which really confused me was where the point 0,0 was. In normal maths, the point 0,0 is in the bottom left hand corner, however the point 0,0 in Processing was in the top left hand corner. Hence when I was doing my coordinates, I would be confused. So this is why I started out just by drawing shapes and line so I could get used to the coordinates.

After playing around I thought of what I could draw. I had a frog sitting on my table next to my computer and I thought it would be a good idea if I could draw a frog which would jump. My initial idea was to just have the frog jump up and down and it would blink.

**Experiment one**

I started off drawing my frog with two main shapes. Ellipses and bezier curves.

```java
smooth();
size(600,600);
background(rand(255), rand(255), rand(255)); //background colour changes
strokeWeight(4);

//FROG'S EYES
fill(0); ellipse(250,150,80,80); //draws his left eye
ellipse(350,150,80,80); //draws his right eye

//FROG'S EYEBALLS
ellipse(270,150,20,20); //draws his left eyeball, which is a circle
ellipse(320,150,20,20); //draws his right eyeball which is a circle
better(320,200,200,200,220,220); //draws the mouth
strokeWeight(8); //make the stroke thicker
strokeCap(1); //sharpenes edges

//FROG'S BODY
fill(0);
stroke(0,0,0);
better(210, 150, 85, 150, 600, 350, 350, 270, 150); //Frog's face
better(0, 0, 0);
better(310, 200, 200, 200, 200, 220, 220, 220, 220); //Frog's body
better(260, 250, 300, 200, 200, 200, 200, 200, 200); //Frog's leg
```

The main trouble I had with this was joining the Bezier curves. I had to get the exact coordinates to make each Bezier curve join with another. The hardest Bezier curve was the frog’s face. My previous Bezier curves kept drawing a concave curve not a convex curve. I originally was trying to draw the frogs face, body and legs with curves, however that did not work, the curves did not look connected.
Experiment 2
After drawing my frog, I thought what I could do which every time I ran it would make a different picture. As I mentioned above that my initial idea was to make the frog blink, I used an “if” statement. The statement would say that the frog’s eyeballs would either be a circle or a line depending on what x value it is. I did not have a lot of trouble with this “if” statement because I basically used the lecture notes and modified them.

Experiment 3

After making the frog blink, my next step was to make the frog’s mouth change expressions. I could only think of three expression it could change between, the original smile [: )], a shocked frog [: o] and [: l]. As you can tell, there was a slight problem when I used the “if” statement.
This is what I initially put in my coding

```java
noFill();
float a=random(500);
if (a<100){
    bezier(250,220,200,280,380,280,350,220);//draws the mouth
}
if (a>300){
    ellipse(290,250,80,80);
} else{
    line(250,250,350,250);
    fill(0);
}
Then I tried this

```java
noFill();
float a=random(166.6);
if (a<33.3){
    bezier(83.3,73.3,66.6,93.3,126.6,93.3,116.6,73.3);//draws the mouth
}
if (a>33.3){
    fill(0);
    ellipse(96.6,83.3,20,20); //draws a circle which represents a shocked frog
}
float f=random(133.3);
if (f<33.3){
    line(83.3,83.3,116.6,83.3);// a mouth which is a straight line
}

```java
***PLEASE TAKE NOTE...my numbers became all decimal points because I originally drew my frog on a 600,600 canvas, I then had to convert it to a 200,200 canvas hence I divided every number by 3 and that is why they are all decimal numbers.***

I still could not figure out why the frog’s mouth was doing what it was doing. I decided to ask a third year (Andrew Morrison) why it was doing what it was doing. He said that I was trying to make one variable do two things at once. Hence when I ran my code it would run both the line/circle/Bezier curve. He showed me how to fix it so that only one variable would draw out either circle/line/Bezier curve.

This is the correct code:

```java
noFill();
float a=random(100);
if (a<20){  //20%
    bezier(83.3,73.3,66.6,93.3,126.6,93.3,116.6,73.3);//draws a smile
}
```
else if (a<75){ //55%
  fill(0);
  ellipse(96.6,83.3,20,20); //draws a circle which represents a shocked frog
}
else { //25%
  line(83.3,83.3,116.6,83.3); // a mouth which is a straight line
}

Hence, my frog would only have either a smile, shocked face or an unsure face.

Experiment 4

I got my frog to blink and change its mouth expressions. However whenever I ran the code, the frog would just be in the same position blinking or winking. I thought it would be good if I could make the frog jump around trying to eat flies (or at least look like it). To get the flying flies I used a loop statement. I actually had a lot of trouble understanding how the loop statement worked, but after looking at the example given to us on the reference website, I slowly understood by changing the numbers of the example loop statement.

This is the one that was given to us on the website

for (int i = 30; i < 80; i = i+5) {
  for (int j = 0; j < 80; j = j+5) {
    point(i, j);
  }
}

The picture given from this code was dots which were lined up next to each other. If I decreased the number from 5 to 2, the dots would be closer together; hence it would make a straight black line. I played around with the code and I managed to come up with a code which when I ran it, it would be little black dots that were in a different position after every run.

I changed and rearranged the code given to us from the website.

strokeWeight(1.5);
for (int i = 50; i < 600; i = i+70) {//the distance between each fly
  for (int j = 50; j < 600; j = j+70) {
    point((int)random(i),(int)random(j));
  }
}
After getting the flies to fly around, the last part was getting the frog to jump around and eat the flies. I had great difficulty with this because the frog was made from different parts put together. I thought that I needed to make everything “random” and I had no idea how to. I didn’t know how to make the entire frog jump around and stay in one piece. I then asked Andrew again and he introduced the pushMatrix idea to me. I know I haven’t learnt this concept but I had really no idea how to make the frog move so I had to accept his idea. The pushMatrix is taking a grid, and saving every coordinate on that grid. I also used the translate variable. The translate variable is a variable which allows you to move your object around the grid given a certain range. For example, in my code below, I had a variable which I called offsetX and the range was any random number between 0 and 400 on the x axis and any offsetY was any random number between 0 and 20. My code said to “translate” offsetX and offsetY which means to move my object anywhere around the grid between offsetX and offsetY ranges.

```java
int offsetX = (int)random(400); //any number between 0-400 in the x axis
int offsetY = (int)random(20); //any number between 0-20 in the y axis
pushMatrix();
translate(offsetX, offsetY); //this will translate or move my “frog” anywhere between the random numbers given above
```

However, I just deleted pushMatrix(); to see whether it would work or not and it did. I even changed my offsetY range to 50 and the frog jumps up and down even more. So instead of it simply jumping sideways like it did before, it now jumps up, down and sideways!

```java
int offsetX = (int)random(400); //any number between 0-400 in the x axis
int offsetY = (int)random(50); //any number between 0-50 in the y axis
translate(offsetX, offsetY); //this will translate or move my “frog” anywhere between the random numbers given above
```

Originally I was going to have the sky which was a gradient. It would start off blue and the very top and gradually become lighter so it looked like the sky. However, I could not work out how to do that so I just kept it a pale blue sky.

I hope you enjoy! 😊