Exercise 4.1 Garfield Cartoon

In this task I had to complete the complete the Garfield comic strip and apply the text as well.

Exercise 4.1 Intro Code

// Ex4-1: Garfield Cartoon

var canvas;

canvas = openGraphics();

canvas.paint();

The way in which I did this was starting with the variables as you would normally. I then added some code in the programme so that all three images could be displayed. The exact code that I used was

var imageName;

var xpoisition = 20;

var ypoisition = 20;

imageName = "garfield1.gif";

canvas.drawImage( imageName, xpoisition, ypoisition, 199, 177);

imageName = "garfield2.gif";

canvas.drawImage( imageName, xpoisition+199, ypoisition, 203, 177);

imageName = "garfield3.gif";

canvas.drawImage( imageName, xpoisition+199+203, ypoisition, 199, 177);

This code displayed 3 separate images of the comic strip

As you can see from this screenshot. The code that I just imput has worked correctly and you can see three different images. This means that the code has worked correctly.

I then had to add the text for the second part of this exercise. The way in which I did this was with a simple canvas.drawString code. The exact code that I used was

canvas.setFont("comic sans ms", "10px", Font.BOLD);

canvas.drawStringRect ("There's A Good Scary Movie on TV Tonight", 32, 25, 150, " center");

canvas.drawStringRect ("Yeah Sure", 140, 67, 50, " center");

canvas.drawStringRect ("You Say That Every Night", 335, 50, 50, " center");

canvas.drawStringRect ("Invasion Of The 50 Foot Adolescents", 445, 23, 110, " center");

canvas.drawStringRect ("Tonight Thought You Would Be Correct", 548, 52, 60, " center");

The third value determines the centre location ^

This code will display text and with the application of the X and Y variables which I implemented earlier I was able to change the text so that it could fit into the thought bubbles exactly.

As a result of this code I was then presented with what I hope was a cartoon strip with text in their allocated thought bubbles.

As we can see from this screenshot there are 3 separate images and there is text inside each of the thought bubbles.

Exercise 4.1 Final Code

// Ex4-1: Garfield Cartoon

var canvas;

2

canvas = openGraphics();

var imageName;

var xpoisition = 20;

var ypoisition = 20;

imageName = "garfield1.gif";

canvas.drawImage( imageName, xpoisition, ypoisition, 199, 177);

imageName = "garfield2.gif";

canvas.drawImage( imageName, xpoisition+199, ypoisition, 203, 177);

imageName = "garfield3.gif";

canvas.drawImage( imageName, xpoisition+199+203, ypoisition, 199, 177);

canvas.setFont("comic sans ms", "10px", Font.BOLD);

canvas.drawStringRect ("There's A Good Scary Movie on TV Tonight", 32, 25, 150, " center");

canvas.drawStringRect ("Yeah Sure", 140, 67, 50, " center");

canvas.drawStringRect ("You Say That Every Night", 335, 50, 50, " center");

canvas.drawStringRect ("Invasion Of The 50 Foot Adolescents", 445, 23, 110, " center");

canvas.drawStringRect ("Tonight Thought You Would Be Correct", 548, 52, 60, " center");

canvas.paint();

Exercise 4.1 My thoughts

Didn’t think this exercise was all that bad. It only required two different commads which were display the image and write the text and I knew how to do this. The only problem which I encountered was moving the code around the canvas to match up with the location of the thought bubbles.

Exercise 4.2 Guess the number

In this task I had to create a numbers guessing game. This was done by making a function in which you had to guess the exact number, and the game would say if the number was higher or lower than your guess.

Initial Code

/ Ex4-2: Guess the Number

var canvas;

canvas = openGraphics();

var max;

max = 100;

var numberToGuess;

numberToGuess = 50; // use a known value for testing

// (1) canvas.drawString( ... suitable title ......

var guess;

// (2) get a guess from the user and make sure it is a numerical value

var message;

// (3) if too high .....

// (4) if too low .....

// (5) "just right" said Goldilocks .....

// (6) canvas.drawString( message ......

canvas.paint();

Exercise 4.2 Task 1

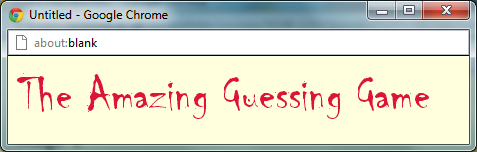
My first task was to load up the initial code and implement code to output a suitable title. This was doing using the simple drawString/setFont/setColor commands. The exact code that I used was

canvas.setFont ( "chiller", "50px", "set.FONT");

canvas.setColor("crimson");

canvas.drawString("The Amazing Guessing Game", 10, 10);

As a result I was presented with this output box.

As we can see from this screenshot the cod e that I have implemented has resulted in an appropriate name, font and colour for my guessing game.

I also needed to imput the code again below the drawstring so that the text after this title were not displayed in the same, colour, font and size.

canvas.setFont ( "cambria", "15px", "set.FONT");

canvas.setColor("black");

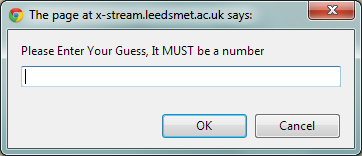
Exercise 4.2 Task 2

My next task was to add the code needed to get a numerical input from the user.

This was done using a simple prompt code.

var guess = prompt ( "Please Enter Your Guess, It MUST be a number");

The code above provided the user with an input box in which to place a guess. I also stated in the code that the guess must be a numerical value or it would not work.

 As we can see from this screenshot the code I input above has worked correctly and looking from this screenshot the prompt has stated that the input guess must be a number.

Exercise 4.2 Task 3

My third task was to add three if() statements that will set the value of message to appropriately as a result of comparing the values of guess and numberToGuess.

The way in which I did this was I made three if statements which their own conditions and if the user input matched the conditions of one of the if statements than they would be presented with the corresponding answer. The answer being the Guess was too high, the guess was too low or the guess was correct.

The exact code which I used was

if(guess > numberToGuess)

{

canvas.drawString("Your Guess Was Too High", xposition, yposition);

}

canvas.paint();

if(guess < numberToGuess)

{

canvas.drawString("Your Guess Was Too Low", xposition, yposition);

}

canvas.paint();

if(guess == numberToGuess)

{

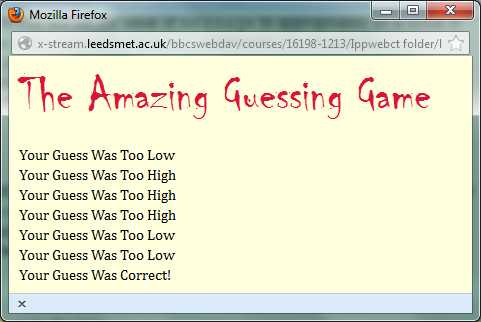
canvas.drawString("Your Guess Was Correct!", xposition, yposition);

guessed = true;

}

canvas.paint();

As a result of this code I will be presented with either “too high”, “too low”, or “correct” as my responses.

As we can see from this screenshot there are numerous results which have been presented. As a result I can see that the code has worked correctly and all three if statements work correctly.

Exercise 4.2 Task 4

My fourth task was to create an output line

canvas.drawString("Your Guess Was Correct!", xposition, yposition);

This was the code that I input to create my output.

This can be seen from the previous screenshot

Exercise 4.2 Task 5

My fifth task in this section was to implement a loop into the equation. The exact code that I used to create a loop was

These two lines of code were placed at the beginning of the if Statements.

loop = 10; 🡨

while (loop >= 0)

This line of code were placed at the end of the if Statements. This code was to stop the loop continuing for an infinite amount of time

loop= loop - 1; 🡨

As a result I was presented with a prompt after prompt asking for a value until the value input was correct.

Exercise 4.2 Section 2

Exercise 4.2 Task 1

In this section I had to declare another identifier, called guessed and assign it the value false. This declaration should be near the declaration for guess.

The exact code that I used was

var guessed;

I put this code beside all my other variables as the top of the code.

Exercise 4.2 Task 2

I then had to Put ALL the code from the section that gets the user's input to the very end of the program into the body of a while loop where the condition is:

while( !guessed )

Include canvas.paint(); inside the loop.

This code here would result in the loop continuing until the number input matched the number you had to guess.

This is what the code looked like when I added the two pieces of code above

while( !guessed ){

var guess = prompt ( "Please Enter Your Guess, It MUST be a number");

yposition = yposition + 20;

if(guess > numberToGuess)

{

canvas.drawString("Your Guess Was Too High", xposition, yposition);

}

canvas.paint();

if(guess < numberToGuess)

{

canvas.drawString("Your Guess Was Too Low", xposition, yposition);

}

canvas.paint();

if(guess == numberToGuess)

{

canvas.drawString("Your Guess Was Correct!", xposition, yposition);

guessed = true;

}

canvas.paint();

}

canvas.paint();

The program will continue to loop until the correct value is found.

Exercise 4.2 Task 3

I then had to add code so that the programm could detect if the answer was correct.

The exact code which I input was

if(guess == numberToGuess)

{

canvas.drawString("Your Guess Was Correct!", xposition, yposition);

guessed = true;

}

canvas.paint();

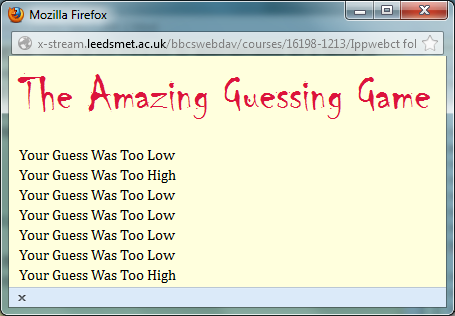
This code would then stop the loop from repeating as the code above tells stops the program as the correct value has been found.

Exercise 4.2 Task 4

I has to change the location of each of the output responses or else they would just remain in one area. The way in which I stopped this from happening was by putting in some new code which meant that the location of the output from the continuous loop was different.

The exact code I used was

yposition = yposition + 20;

This code results in the current location of the y position text which was 70 having an additional 20 on the scale and as a result the output moves down the output box.

As we can see from this screenshot the output responses move continuously down the y position of the output box.

Exercise 4.2 Task 5

In this task I had to input code so that the program would pick a number randomly and so the game could not be easily beaten by the user who set the number to guess.

The code which I input was

numberToGuess = Math.floor( Math.random() \* max ) + 1;

The code above replaced the standard

numberToGuess = 50;

As a result the program chooses the number and the game is now completely and fully operational.

Exercise 4.2 Extensions

I then decided to change the colour of the output response. For example if the guess was too high the colour would be displayed in a red colour. If the guess was to low then the colour would be displayed in a blue colour and if the prediction was correct then the colour of the text would be green.

This was a simple setColour code which was input in each of the if statements for example if(guess < numberToGuess)

{

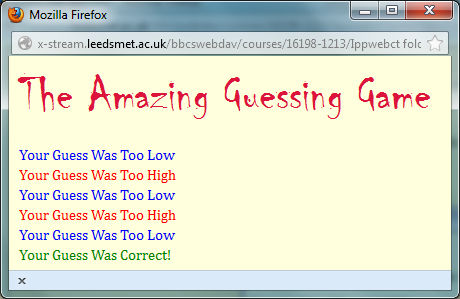
canvas.setColor("blue");

canvas.drawString("Your Guess Was Too Low", xposition, yposition);

}

canvas.paint();

This code would mean that the output information would be displayed in a blue.



As we can see from this screenshot the code which I have input in each of the if statements has worked correct and all 3 if statements has worked correctly as the low guesses are blue, high guesses are red and correct guesses are green.

Exercise 4.2 Final Code

// Ex4-2: Guess the Number

var canvas;

canvas = openGraphics();

var max;

max = prompt("what do what want to maximum number to be?");

var xposition = 10;

var yposition = 70;

var loop;

var guessed;

var numberToGuess;

numberToGuess = Math.floor( Math.random() \* max ) + 1;

canvas.setFont ( "chiller", "50px", "set.FONT");

canvas.setColor("crimson");

canvas.drawString("The Amazing Guessing Game", 10, 10);

var message;

canvas.setFont ( "cambria", "15px", "set.FONT");

canvas.setColor("black");

loop = 10;

while( !guessed ){

var guess = prompt ( "Please Enter Your Guess, It MUST be a number");

yposition = yposition + 20;

if(guess > numberToGuess)

{

canvas.setColor("red");

canvas.drawString("Your Guess Was Too High", xposition, yposition);

}

canvas.paint();

if(guess < numberToGuess)

{

canvas.setColor("blue");

canvas.drawString("Your Guess Was Too Low", xposition, yposition);

}

canvas.paint();

if(guess == numberToGuess)

{

canvas.setColor("green");

canvas.drawString("Your Guess Was Correct!", xposition, yposition);

guessed = true;

}

canvas.paint();

}

canvas.paint();

Exercise 4.2 my thoughts

I thought this exercise was very simple the only problem which I encountered in this exercise was my location of specific code. I forgot to input guessed = true; code inside the if statement which determined the correct response for my game what going round on a continuous loop not ending the game and not knowing if the answers we gave were even correct in the first place.

Exercise 4.3 – BMI Calculator

In this exercise I had to produce a BMI calculator. I did this in a long series of instructions.

Exercise 4.3 Initial Code

// Ex4-4: BMI Calculator

Exercise 4.1 Task 1 – The simplest BMI calculator

In this task I had to declare three identifiers. These identifiers were height, weight and BMI. The exact code which Input was.

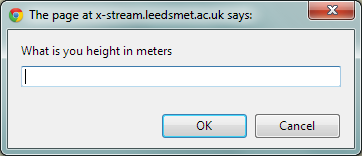
var height;

var weight;

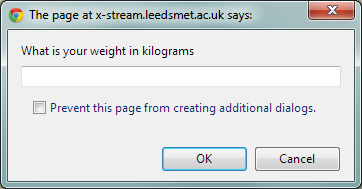
var BMI;

Secondly I had to Assign values to the height and weight; pick values from the test data table in Step 3 of the Introduction. This was easily done using prompt commands. The exact code which I used was.

Height = prompt( "What is you height in meters");

Weight = prompt( "What is your weight in kilograms");

As you can see from these two screenshots the code has worked correctly and the prompts have been displayed as I programmed them to do.

The value in the two prompts boxes which them be calculated by the formula which I input and ultimately a BMI will be calculated.

I then had to create an expression to calculate the BMI and assign it to the BMI identifier. I did this by input a formula into my program. The exact piece of code that I used to calculate the BMI was.

BMI = weight / (height \* height);

This formula determines the BMI from the values which I input in my height and weight prompts.

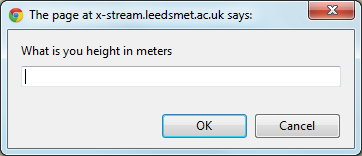
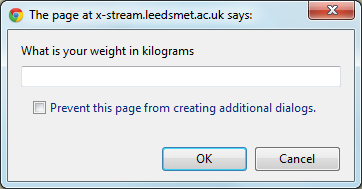
Exercise 4.1 Task 2 – Add user imput

I then had to change the code for the values so that the user can have an input. This was done using simple prompt code. The exact code that I used was

height = prompt( "What is you height in meters");

weight = prompt( "What is your weight in kilograms");

This code would display two output boxes asking for the weight and the height of the user.



As you can see from these screenshot above, the code has worked and I have been presented with two prompts. The values that the user put in these boxes will determine the value which is presented in the BMI calculator.

I then had to add further code so that the users input can be calculated has a whole number. This can be done using the code parseFloat.

height = parseFloat(height);

weight = parseFloat(weight);

Exercise 4.1 Task 3 – Classify the BMI

I then had to write more code into my program, so that the values determine by the BMI formula I input are classified. These way in which I did this was through standard if formulas. The exact code that I used was.

if(BMI<18.5) 🡨I changed this value many times to include all values in the BMI

{

canvas.drawString( "is Underweight", 130, 70); 🡨 This drawstring code changed as well.

canvas.paint();

}

Exercise 4.1 Final Code

// Ex4-4: BMI Calculator

var canvas;

canvas = openGraphics();

var height;

var weight;

var BMI;

height = prompt( "What is you height in meters");

weight = prompt( "What is your weight in kilograms");

height = parseFloat(height);

weight = parseFloat(weight);

BMI = weight / (height \* height);

canvas.drawString("Your Height is :" + height, 10, 10);

canvas.drawString("Your Weight is :" + weight, 10, 30);

canvas.drawString("Your BMI is: " + BMI , 10, 50);

canvas.drawString("Your Classification: " , 10, 70);

if(BMI<18.5)

{

canvas.drawString( "is Underweight", 130, 70);

canvas.paint();

}

else if(BMI>18.5 && BMI<= 24.9)

{

canvas.drawString( "is Ideal", 130, 70);

canvas.paint();

}

else if(BMI>=25 && BMI<= 29.9)

{

canvas.drawString( "is Overweight", 130, 70);

canvas.paint();

}

else if(BMI >=30)

{

canvas.drawString( "is Obese", 130, 70);

canvas.paint();

}

else if(BMI > 40)

{

canvas.drawString( "is Very Obese", 130, 70);

canvas.paint();

}

canvas.paint();

Exercise 4.1 My thoughts

I felt that this program was fairly easy; I thought that due to the fact that the code was similar to the “Grading Work” I was able to complete the work quite quickly, the only time consuming task was the research I had to do to find out about the requirements for each classification. The if statements were very easy to complete as I had already done them before. The only problem which I did encounter was the parseFloat code. I didn’t know how to write them so as a result I research them and found the correct way.

Exercise 5.1 - Weather Data Graph

In this task I had to create a program that will accept some of this data, input by the user, and to produce a graph that is suitably formatted.

Exercise 5.1 - Initial Code

var canvas;

canvas = openGraphics();

//You may have to decide where this is best placed

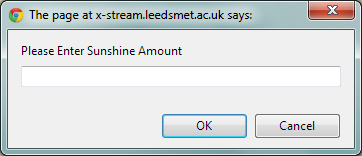
canvas.paint();

Exercise 5.1 Task 1 – Input from the user

In order to record the information on my graph I first need to gain the information to calculate the size of the bars in my graph. The values are from the user and inputting a standard prompt code I can easily get this information. The exact code that I used was

var value = prompt("Please Enter Sunshine Amount");

value = parseInt(value,10);

As we can see from this screenshot the code has worked and as a result I have been presented with this prompt box.

Exercise 5.1 Task 2 - Drawing my Graph

I first had to declare my variables so that I can then write more code to draw my axis in which my graph will rest on. The exact code that I used to declare my variables was.

var canvas;

canvas = openGraphics();

var xpos = 30;

var ypos = 40;

var height = 225;

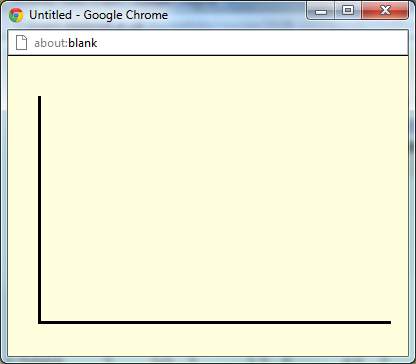
var width = 350;

canvas.setStroke(3);

canvas.drawLine(xpos, ypos + height, xpos, ypos);

canvas.drawLine(xpos, ypos + height, xpos+ width, ypos + height);

canvas.paint();

As a result, after I declared my variables and wrote the code which displayed the axis for my graph I was presented with an X position line and a Y position line. As we can see from this screenshot the code that I have input has worked correctly and I have been presented with this axis. This means that the code has worked correctly. The code now provides me with space to add bars and record the values that I input into my program.

Exercise 5.1 Task 3 - Naming my axis

I then had to write some code so that the x axis and the y axis had values to that any input I put in has meaning. The way in which I did this was inputting code after my drawLine code. The exact code that I used was.

// the code below is the values on the y axis going up in 45 until it reaches 225

var ylabel = 0;

while(ylabel <= 225)

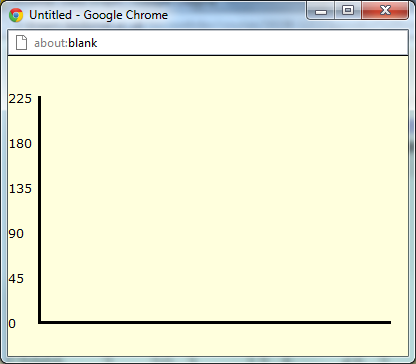
{

canvas.drawString(ylabel, 0, ypos + height - ylabel - 5);

ylabel += 45;

}

This code meant that the Y axis had values on it.

As you can see from this screenshot, the Y axis has values and they are increasing in 45s until it reaches 225.

I then had to name my X axis; this consisted of the months of the year. The way in which I did this was again through code. The code which I used to name the x axis was this code.

//the code below is the values on the x axis going from January to December.

var months = "JFMAMJJASOND";

var xlabel = xpos + 10;

var barwidth = width/12;

for (var loop = 0; loop < 12; loop++)

{

var nextLetter= months.charAt(loop);

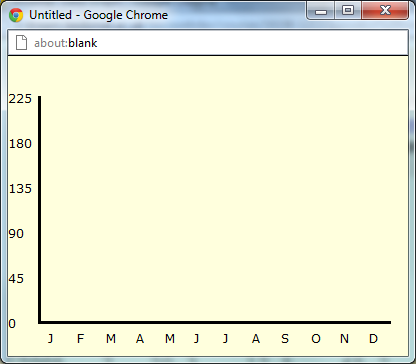
canvas.drawString(nextLetter, xlabel, ypos + height + 10);

xlabel = xlabel + barwidth;

}

xpos = xpos + 3;

This code above names the x axis. As you can see from this screenshot below we can see from this screenshot the x axis has been named so that means the code has worked correctly.



We can now see from the screenshot above that both the x axis and the y axis are names and have values on them.

Exercise 5.1 Task 4 – Creating the bars

I then had to create a loop so that the bar could represent the 12 months of the year. The way in which I did this was through a standard loop. The exact code that I used was

// the code below creates the bars and gives them specific shapes, outlines and colours

for (var month = 0; month < 12; month++)

{

var value = prompt("Please Enter Sunshine Amount");

value = parseInt(value,10);

canvas.setStroke(Stroke.Dotted);

canvas.setColor("pink");

canvas.fillRect( xpos, ypos + height - value, barwidth, value);

canvas.setColor("black");

canvas.drawRect( xpos, ypos + height - value, barwidth, value);

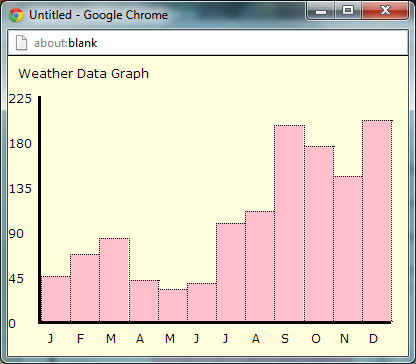
xpos = xpos + barwidth;

canvas.paint();

}

The code above determines the size, colour and shape of the bars which display the users inputted values.

The screenshot below shows the effects of that code.

As we can see from the screenshot each of the bars is divided perfectly across the x boarder. Due to the code

var barwidth = width/12

This code means that the bar is divided across 12 months

Exercise 5.1 – Final Code.

//exercsie 5.1

var canvas = openGraphics();

var xpos = 30;

var ypos = 40;

//the code below creates the axis for my graph to rest on.

var height = 225;

var width = 350;

canvas.setStroke(3);

canvas.drawLine(xpos, ypos + height, xpos, ypos);

canvas.drawLine(xpos, ypos + height, xpos+ width, ypos + height);

// the code below names the y axis. going up in 45s until 225

var ylabel = 0;

while(ylabel <= 225)

{

canvas.drawString(ylabel, 0, ypos + height - ylabel - 5);

ylabel += 45;

}

canvas.drawString("Weather Data Graph", 10 , 10);

//The code below names the x axis. a letter for each letter of every month

var months = "JFMAMJJASOND";

var xlabel = xpos + 10;

var barwidth = width/12;

for (var loop = 0; loop < 12; loop++)

{

var nextLetter= months.charAt(loop);

canvas.drawString(nextLetter, xlabel, ypos + height + 10);

xlabel = xlabel + barwidth;

}

xpos = xpos + 3;

// the code below creates the bars and gives them specific shapes, outlines and colors

for (var month = 0; month < 12; month++)

{

var value = prompt("Please Enter Sunshine Amount");

value = parseInt(value,10);

canvas.setStroke(Stroke.Dotted);

canvas.setColor("pink");

canvas.fillRect( xpos, ypos + height - value, barwidth, value);

canvas.setColor("black");

canvas.drawRect( xpos, ypos + height - value, barwidth, value);

xpos = xpos + barwidth;

canvas.paint();

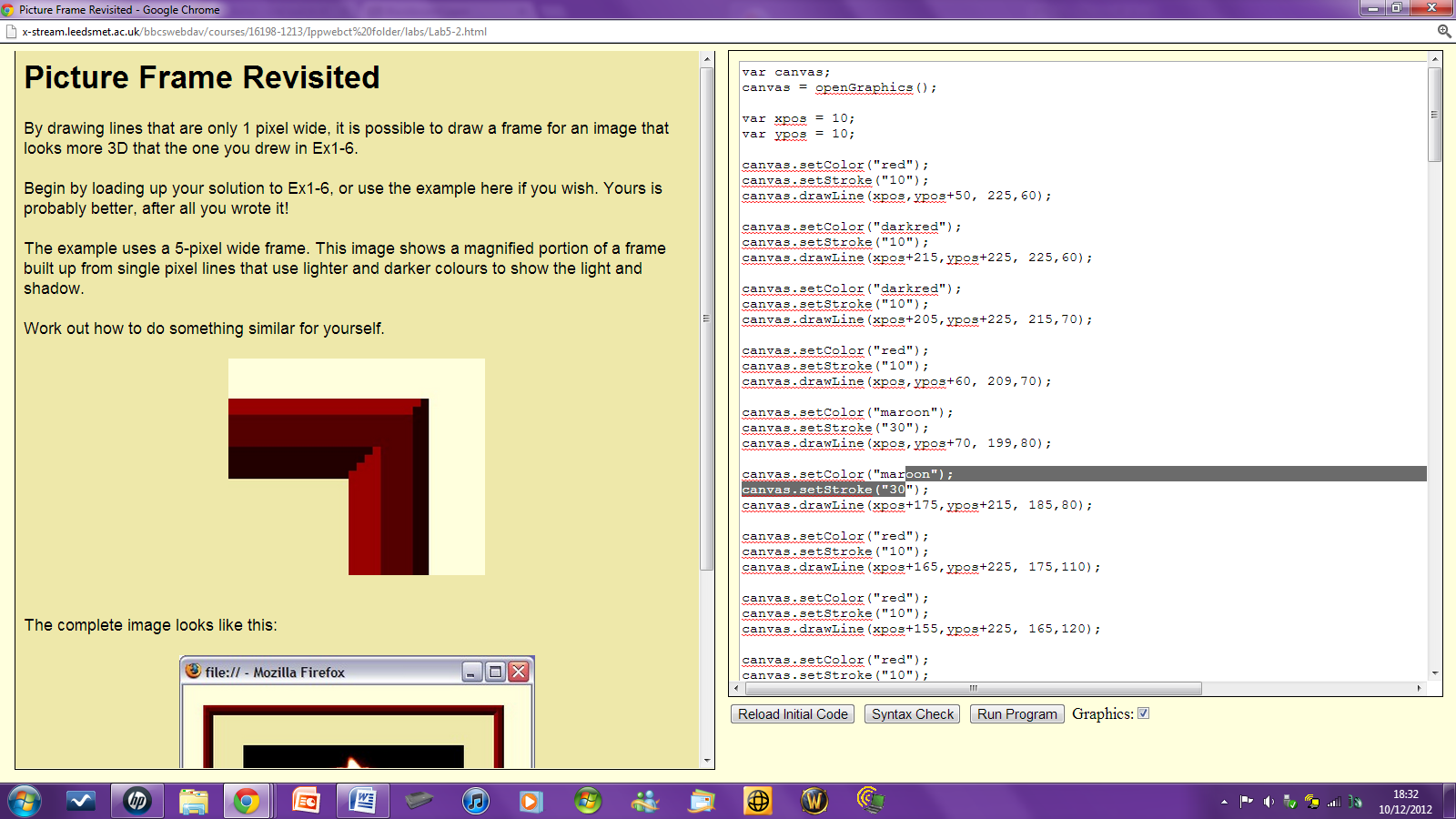
}

Exercise 5.1 – My thoughts.

I felt that this exercise was a very hard exercise. Due to the placing and the creation of the bars, the code which you had to write so that the names of the axis were positioned correctly and the axis itself. The majority of the code was quite complex and at some times challenge, the task was time consuming when it came to understanding what everything did.

Exercise 5.2 – Picture Frame Revisited.

In this exercise I had to use code to create this image below.

The way in which I went about writing the code and creating this image was through the code known as canvas.drawLine.

I started this exercise with the simple canvas.drawLine code and declaring all my variables which I required.

The exact code that I used was this.

var canvas;

canvas = openGraphics();

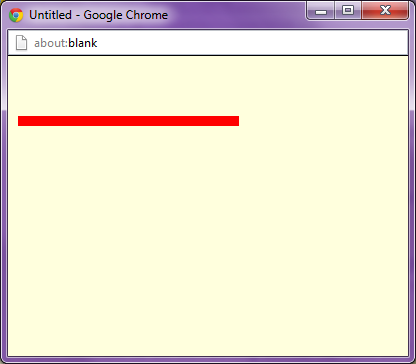
var xpos = 10;

var ypos = 10;

canvas.setColor("red");

canvas.setStroke("10");

canvas.drawLine(xpos,ypos+50, 225,60);

The result of this code was that I was then presented with this image.

As we can see from this screenshot the code has worked and I have been provided with a red line which is exactly what I wanted and provides the foundation for the next part of the exercise and will undoubtedly help me finishing it.

Exercise 5.2 – Complete Code

I replicated the first lines of code over and over and changed the values so that the location and size of the lines would also change as it appears in the example picture. The finished code which provided me with exact image was.

var canvas;

canvas = openGraphics();

var xpos = 10;

var ypos = 10;

canvas.setColor("red");

canvas.setStroke("10");

canvas.drawLine(xpos,ypos+50, 225,60);

canvas.setColor("darkred");

canvas.setStroke("10");

canvas.drawLine(xpos+215,ypos+225, 225,60);

canvas.setColor("darkred");

canvas.setStroke("10");

canvas.drawLine(xpos+205,ypos+225, 215,70);

canvas.setColor("red");

canvas.setStroke("10");

canvas.drawLine(xpos,ypos+60, 209,70);

canvas.setColor("maroon");

canvas.setStroke("30");

canvas.drawLine(xpos,ypos+70, 199,80);

canvas.setColor("maroon");

canvas.setStroke("30");

canvas.drawLine(xpos+175,ypos+215, 185,80);

canvas.setColor("red");

canvas.setStroke("10");

canvas.drawLine(xpos+165,ypos+225, 175,110);

canvas.setColor("red");

canvas.setStroke("10");

canvas.drawLine(xpos+155,ypos+225, 165,120);

canvas.setColor("red");

canvas.setStroke("10");

canvas.drawLine(xpos+145,ypos+225, 155,130);

canvas.setColor("red");

canvas.setStroke("10");

canvas.drawLine(xpos+135,ypos+225, 145,140);

canvas.setColor("darkred");

canvas.setStroke("10");

canvas.drawLine(xpos+160,ypos+100, 10,110);

canvas.setColor("darkred");

canvas.setStroke("10");

canvas.drawLine(xpos+150,ypos+110, 10,120);

canvas.setColor("darkred");

canvas.setStroke("10");

canvas.drawLine(xpos+140,ypos+120, 10,130);

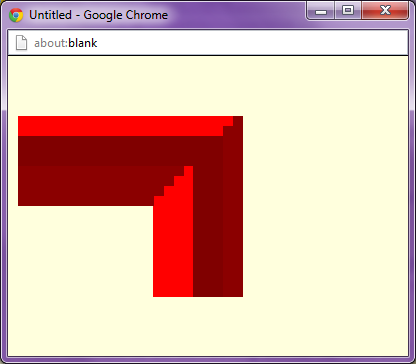
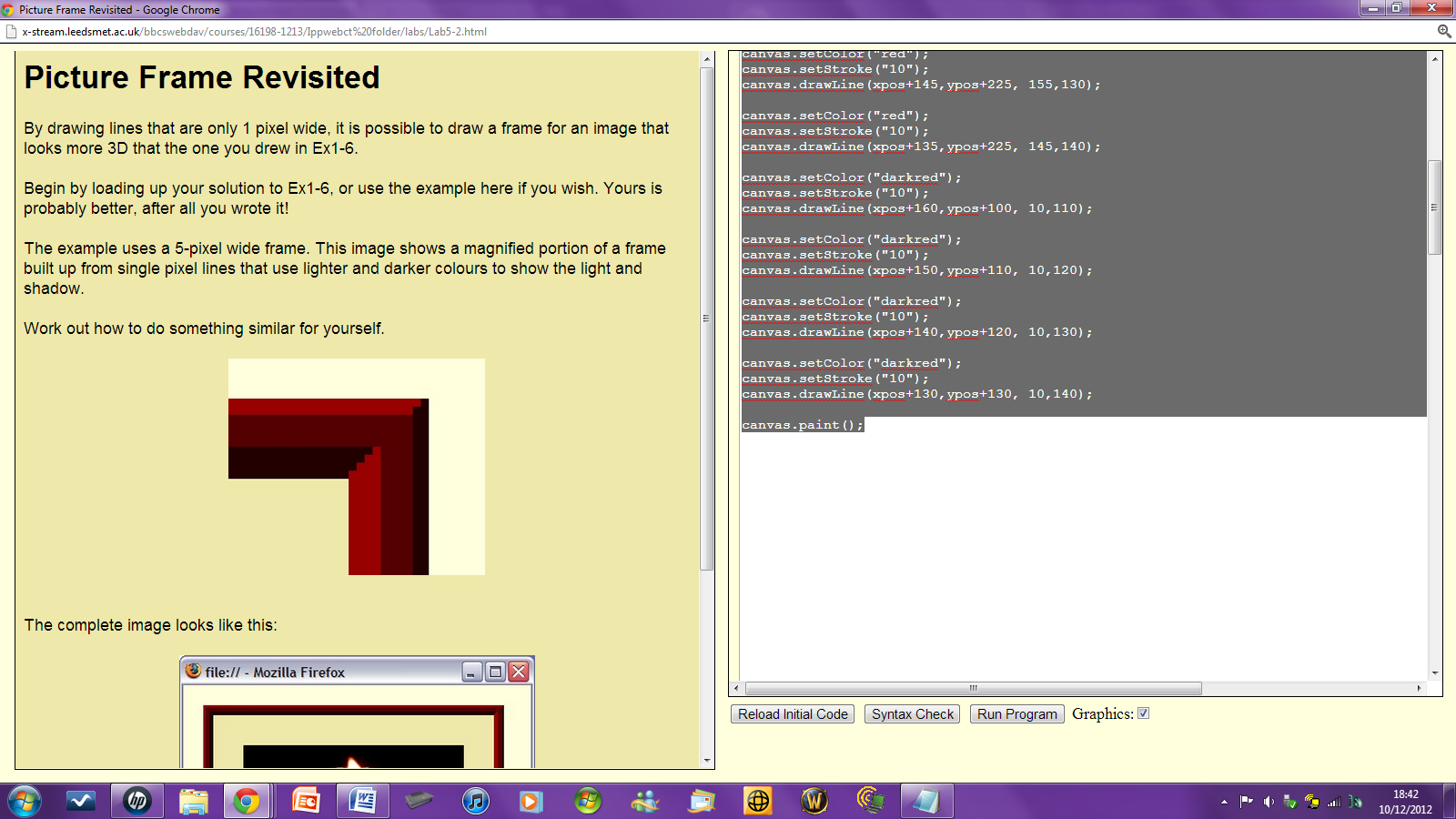
canvas.setColor("darkred");

canvas.setStroke("10");

canvas.drawLine(xpos+130,ypos+130, 10,140);

canvas.paint();

This complete code provided me with this image.



As we can see the image on the left which is my image is exactly the same as the example which was presented to me befor the exercise. This means that the code that I used is correct and functioning correctly.

Exercise 5.2 - My thoughts

I felt that this exercise was very simple the only comands which were required in this exercise were the drawLine commands. The only problem which I encountered was that the process of moving the location of the line and their location became a very tedious and time consuming task.