while Loops

Objectives

On completion of this lab you should understand variable scope and be able to code static drawings using for and while loops.

Variable scope (local and global)

- In this step, we will implement the code examples from your lectures.

Understanding variable scope

- Create a new Processing sketch in your workspace and call it **practical07_variable_scope**.
- Enter the following code into your sketchbook (don't copy and paste...write the code out):

```
void setup()
{
 size(500,400);
 background(0);
 stroke(255);
 fill(45,45,45);
}

void draw() {
 int diameter = 100;
 if (mousePressed)
 {
  diameter = diameter – 10;
  background(0);
 }
 ellipse(mouseX, mouseY, diameter, diameter);
}
```

- Run your code. Is your circle reducing in size? Do you see the problem?
    - The diameter variable is declared in the draw() function i.e. it is a local variable.
    - It is only “alive” while the draw() function is running.
    - Each time the draw() function:
        - finishes running, the diameter variable is destroyed.
        - is called, the diameter variable is re-created.
- To fix this, change your code so that it looks like this:

```
int diameter = 100;
```

```
void setup()
{
 size(500,400);
 background(0);
 stroke(255);
 fill(45,45,45);
}

void draw()
{
 if (mousePressed)
 {
  diameter = diameter - 10;
  background(0);
 }
 ellipse(mouseX, mouseY, diameter, diameter);
}
```

- Run your code. Does it work as you would expect?
- There is a problem with the code. In the ellipse method, the width and height are absolute values (the negative sign is dropped).
- To handle this logic bug, we need to stop reducing the diameter by 10 when we reach a certain value, say 20.
- Implement this code and test your code again:

```
int diameter = 100;

void setup()
{
 size(500,400);
 background(0);
 stroke(255);
 fill(45,45,45);
}

void draw()
{
 if ((mousePressed) && (diameter > 20))
 {
  diameter = diameter - 10;
  background(0);
 }
 ellipse(mouseX, mouseY, diameter, diameter);
```

```
}
```

- Did you notice that it seems the reduction is larger than 10 when we press the mouse?
- Why? The default frame rate is 60 refreshes of the screen per second i.e. draw() is called 60 times per second.
- You can change the frame rate by calling the frameRate() function.
- Now try this solution:

```
int diameter = 100;

void setup()
 {
  size(500,400);
  background(0);
  stroke(255);
  fill(45,45,45);
  frameRate(20);
}

void draw()
{
  if ((mousePressed) && (diameter > 20)){
    diameter = diameter - 10;
    background(0);
  }
  ellipse(mouseX, mouseY, diameter, diameter);
}
```

Exercises

- For each exercise listed below, open a new sketchbook.
- You may need to visit the Processing website for additional information.
- When you are finished all your exercises, zip all your exercises into one file and send them as **Practical 7** to your lecturer.

Console Exercise 1

- Create a new sketch and use a **while** loop to print hello 25 times to the console.

Console Exercise 2

- Create a new sketch and use a **while** loop to print the numbers 1 to 10 on different lines to the console.

Console Exercise 3

- Create a new sketch and use a **while** loop to print the numbers 10 to 1 on different lines to the console.

Console Exercise 4

- Create a new sketch and use a **while** loop and the println method to print
  10, 9, 8, 7, 6, 5, 4, 3, 2, 1, blast off
  on the same line to the console.

Console Exercise 5

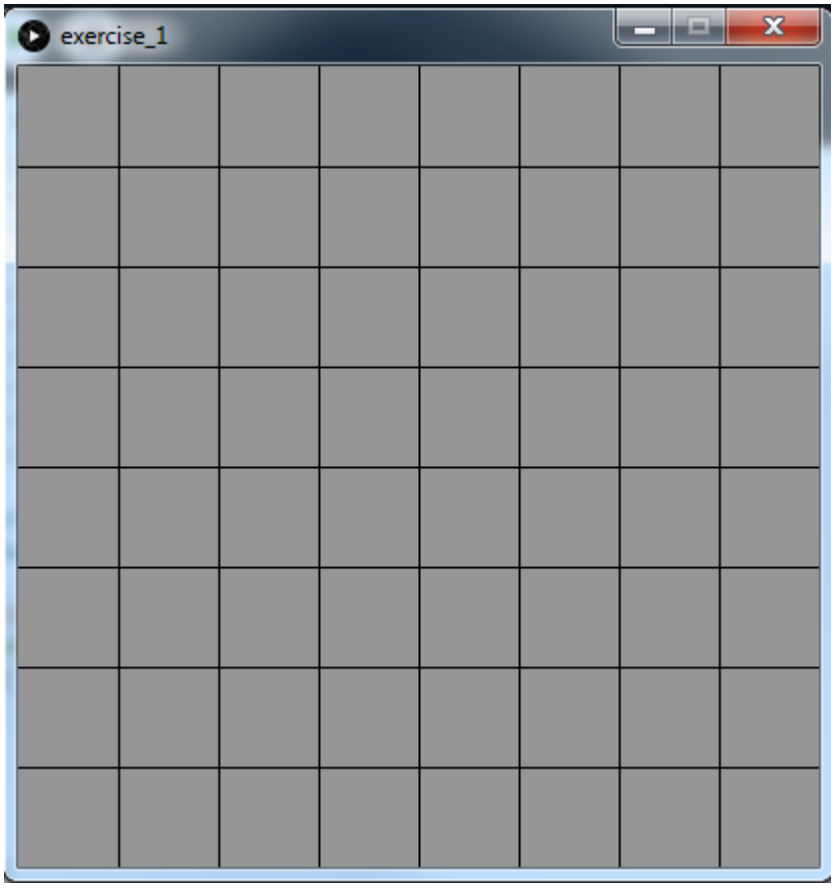
- Create a new sketch and use a **while** loop to print all the even numbers between 1 and 10 to the console.

Console Exercise 6

- Create a new sketch and use a **while** loop and the println method to print all the odd numbers between 1 and 100 to the console.

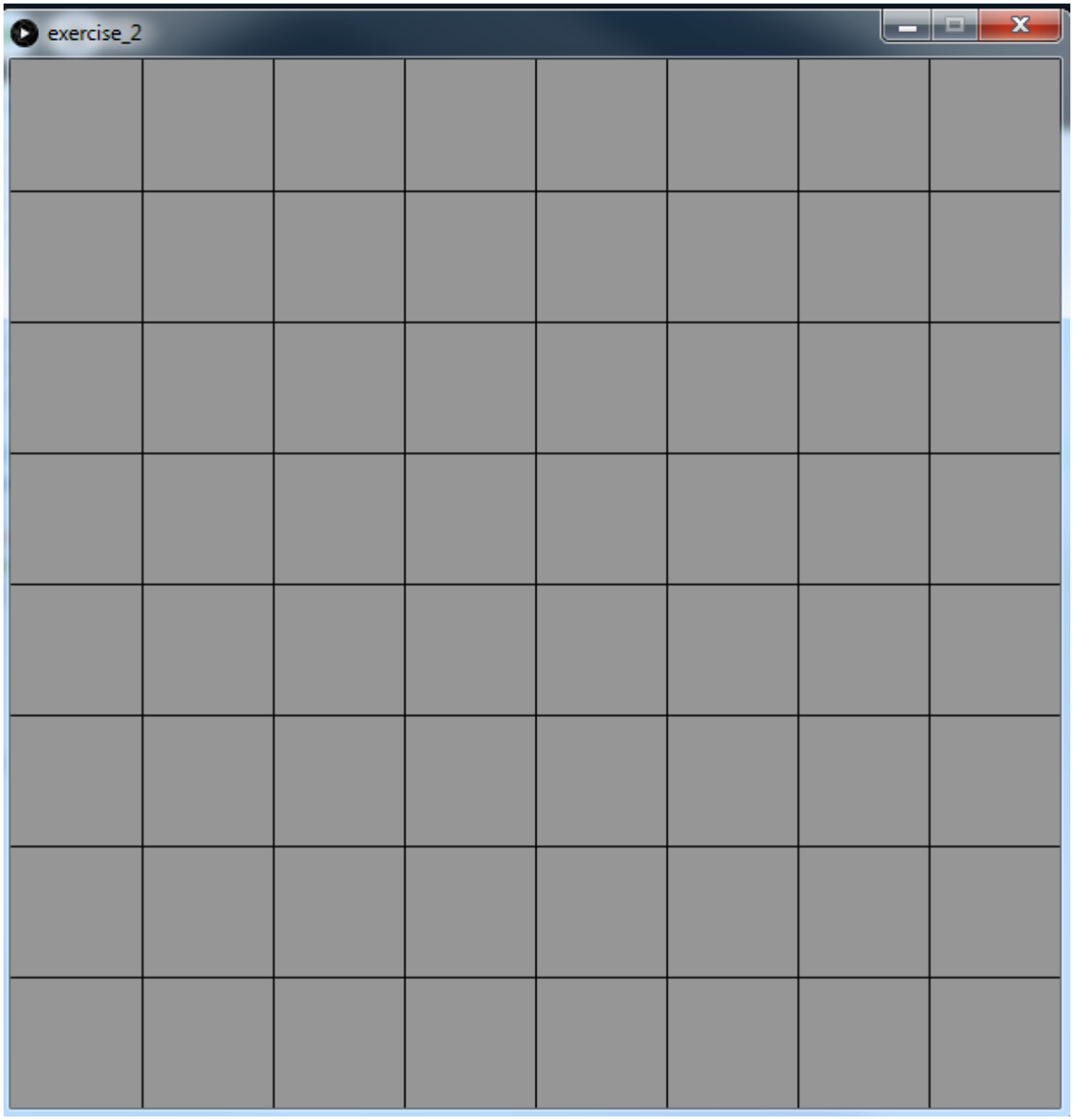
Exercise 1 (static drawing)

- Create a display window of 400x400 with a grey background.
- In the setup() method, use a **while** loop to draw a chessboard (for this exercise, use the line() method).
- Hint: you only need two lines of code in your while loop (not including your LCV update).
- A chess board is an 8x8 grid and should look like the screen shot below:



Exercise 2 (static drawing)

- Create a new sketch and re-write the **Exercise 1** code so that the chess board is drawn correctly regardless of the width and height of the display window.



Exercise 3 (static drawing)

- Create a new sketch to draw 10 points (look up point method) with 10 pixels (spaces) between them.

Exercise 4 (static drawing)

- Create a new sketch to draw 10 points with 10 pixels (spaces) between them going horizontally.