Iteration in Programming

Help – Moving Line

Produced by:

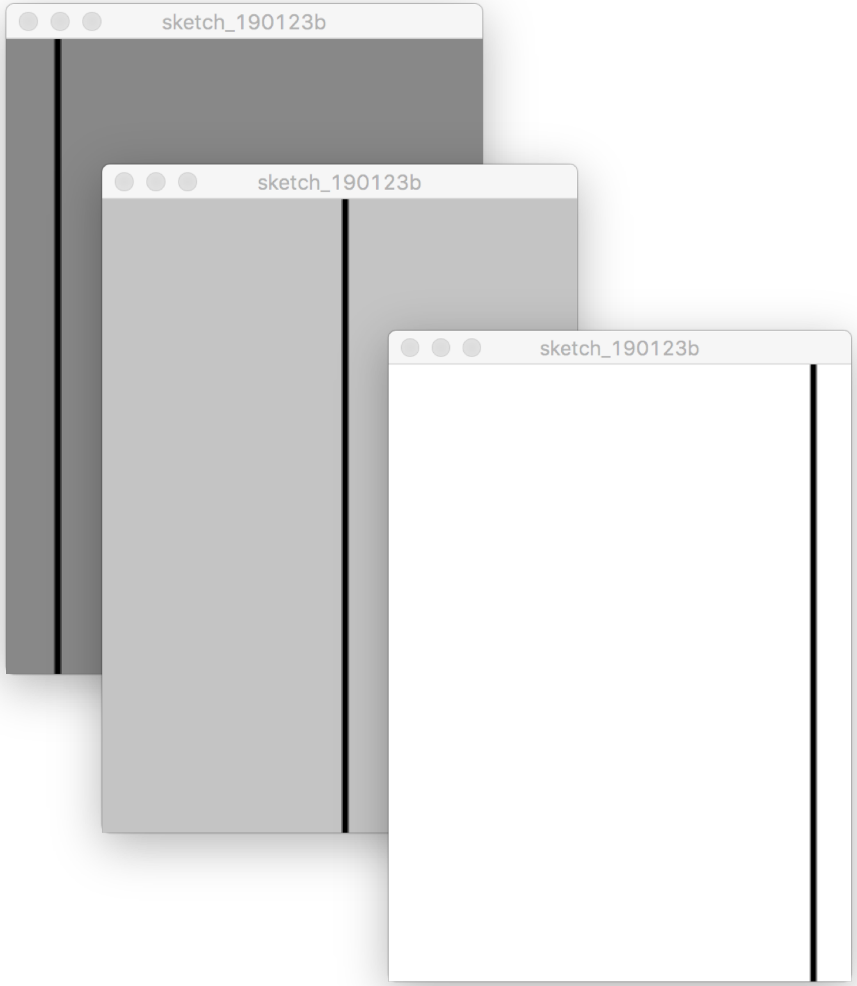
Dr. Siobhán Drohan
Mr. Colm Dunphy
Mr. Diarmuid O’Connor

Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

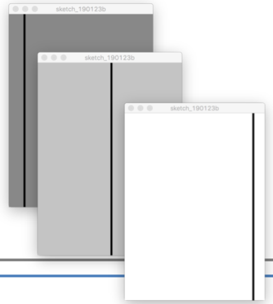
Department of Computing and Mathematics
http://www.wit.ie/

Topics list

- There are three types of loop in programming:
  - While loops:
    - Counter controlled (n times) - covered in previous talk
    - Sentinel based (covered later in the course)
    - Flag based (covered later in the course)
  - For loops (this slide deck)
  - Do While loops (covered later in the course)
- Comparative use of **while** and **for** loops
- Challenges
  - Lab02a - Challenge 1 - Bouncing Ball
  - Lab02a - Challenge 3 – Moving Line



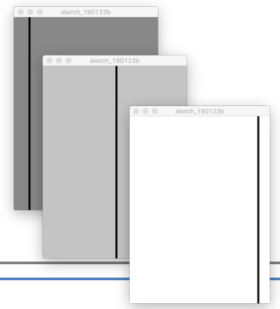
Lab02a - Challenge 3 – Moving Line



- In a new sketch,
  draw a **vertical line**
  that is the height of your display window.

- It starts in the left most position of your display window
  and **moves right, pixel by pixel**,
  until it reaches the right hand side of your display window.

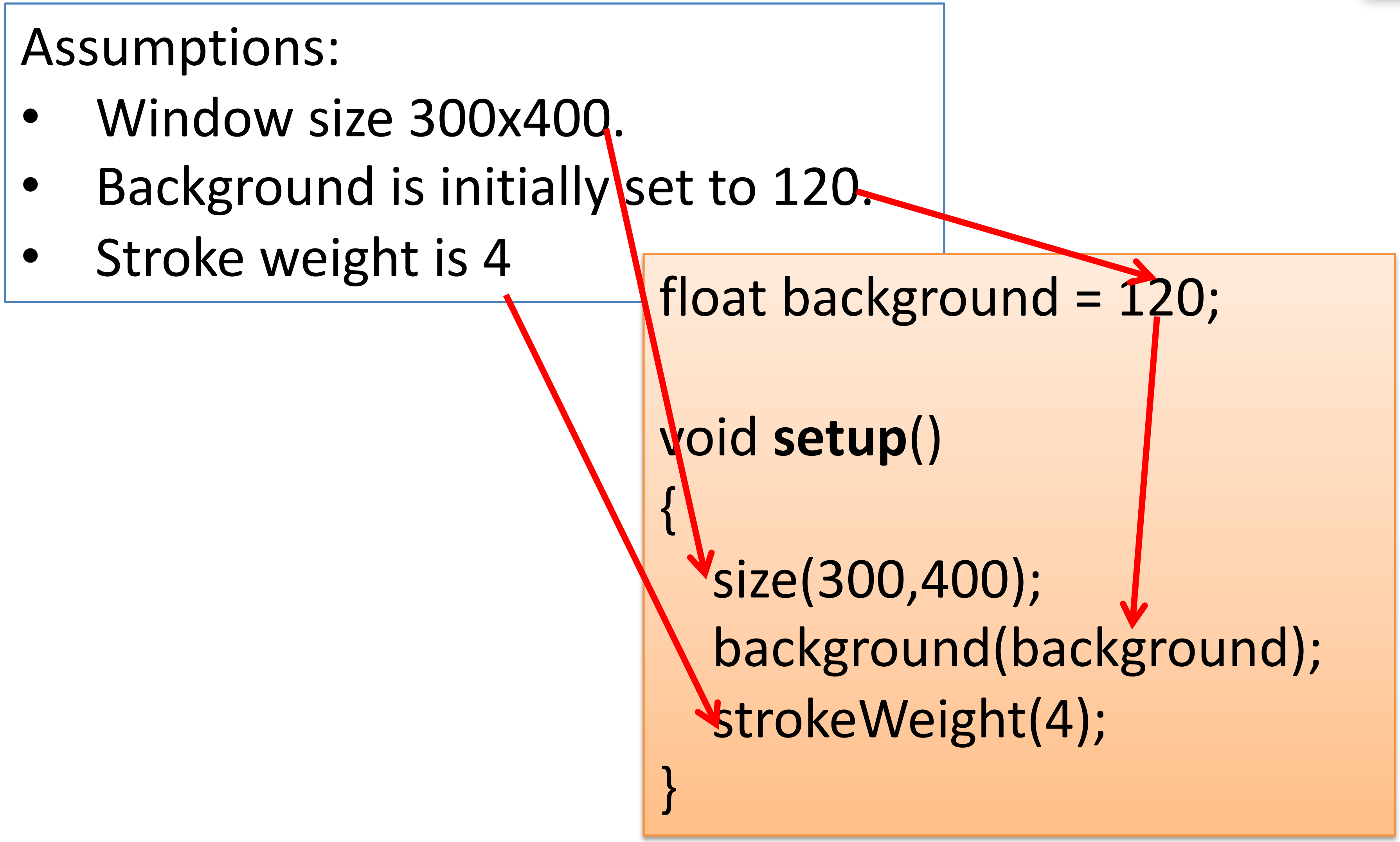
Lab02a - Challenge 3 – Moving Line



- Upon reaching the right hand side,
  the vertical line should **reverse direction**
  and return, pixel by pixel,
  to the left hand side of the display window.

- As your vertical line is continually traversing the display window,
  your **grayscale background should be varying** very slightly in colour.

Lab02a - Challenge 3 – Moving Line



Assumptions:

- Window size 300x400.
- Background is initially set to 120.
- Stroke weight is 4

```
float background = 120;

void setup()
{
  size(300,400);
  background(background);
  strokeWeight(4);
}
```

Lab02a - Challenge 3 – Moving Line

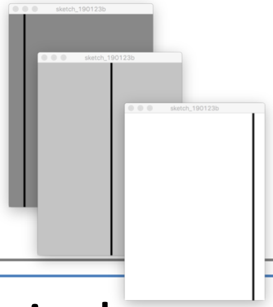
- Draw a **vertical line** that is the height of your display window.
- Call background to **clear the previously drawn line**.

```
float background = 120;
float xCoordinate = 0.0;

void setup(){
   size(300,400);
   background(background);
   strokeWeight(4);
}
```

```
void draw()
{
  background(background);
  line (xCoordinate, 0, xCoordinate, height);
}
```

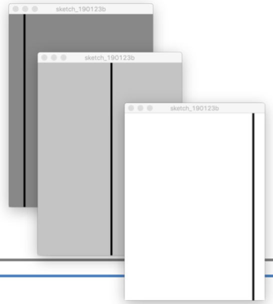
Lab02a - Challenge 3 – Moving Line



This vertical line should start in the left most position of your display window and **move right, pixel by pixel**, until it reaches the right hand side of your display window.

```
void draw(){
  xCoordinate = xCoordinate + 1;
  background(background);
  line (xCoordinate, 0, xCoordinate, height);
}
```

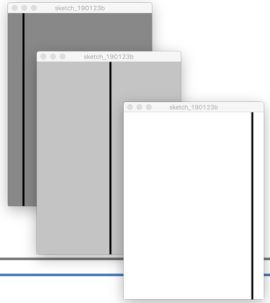
Lab02a - Challenge 3 – Moving Line



As your vertical line is continually traversing the display window,
your **grayscale background** should be **varying** very slightly **in colour.**

```
void draw(){
   xCoordinate = xCoordinate + 1;
   background = background + 0.5;
   background(background);
   line (xCoordinate, 0, xCoordinate, height);
}
```

Lab02a - Challenge 3 – Moving Line

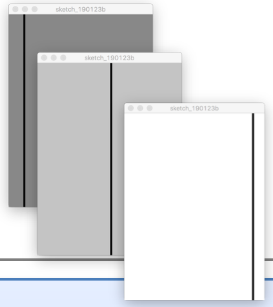


- Upon reaching the right hand side,
  the vertical line should **reverse direction**
  and return, pixel by pixel, to the left hand side of the display window.

- We need to keep track of the **direction** that the line should be moving
  i.e. is it going left-to-right, or has it reversed direction and gone from right-to-left?

- We will use a boolean variable to do this:
  - boolean **reverseDirection** will be initially set to false,
    indicating a left-to-right direction.
  - **false** indicates a **left-to-right direction**
  - **true** indicates a **right-to-left direction**.

Lab02a – Challenge 3



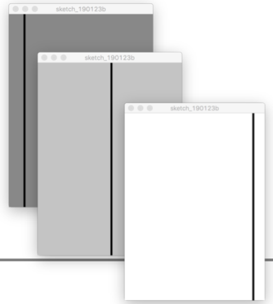
```
void draw()
{
 if (!reverseDirection){
     background = background + 0.5;
     xCoordinate = xCoordinate + 1;
  }
  else{
    background = background - 0.5;
    xCoordinate = xCoordinate - 1;
  }

  background(background);
  line (xCoordinate, 0, xCoordinate, height);
}
```

```
float background = 120;
float xCoordinate = 0.0;
boolean reverseDirection = false;

void setup(){
    size(300,400);
    background(background);
    strokeWeight(4);
}
```

Lab02a - Challenge 3 – Moving Line



- But, we have no code written that will set the flag to true e.g.

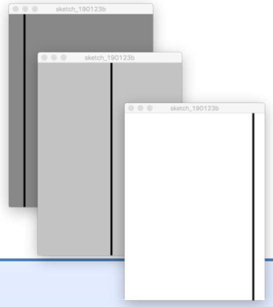
```
        boolean reverseDirection = true;
```

- <u>QUESTION</u>:
  Under what circumstances should the flag be set to true?
  And when should it be set back to false?

```
void draw(){
 if (xCoordinate == width)
    reverseDirection = true;
  if (xCoordinate == 0)
    reverseDirection = false;

if (!reverseDirection){
    background = background + 0.5;
    xCoordinate = xCoordinate + 1;
  }
  else{
   background = background - 0.5;
   xCoordinate = xCoordinate - 1;
  }

  background(background);
  line (xCoordinate, 0, xCoordinate, height);
}
```



```
float background = 120;
float xCoordinate = 0.0;
boolean reverseDirection = false;

void setup(){
   size(300,400);
   background(background);
   strokeWeight(4);
}
```