CGRA 151 — Worksheet 2 — basic animation and mouse interaction

Objective: to give you confidence in what is needed to complete the core of Assignment 2.

Aim of this worksheet: to get an understanding of how to do simple animation by making a block move backwards and forwards left-right on the screen and then interact with the mouse in a simple way.

Drawing a block

Open processing.

Get a simple block on the screen.

```
float x=100, y=100 ;
void setup() {
  size(400, 300) ;
}
void draw() {
  rect(x, y, 40, 40);
}
```

We have parameterised the position (`x`, `y`) of the block so that we can move it.

Make it move

Add a velocity variable at the top of the sketch:

```
float vx = 1.0 ;
```

Make the rectangle move horizontally on every cycle by adding, to the `draw()` function:

```
x += vx ;  // remember this means x = x + vx ;
```

What happens when you run this?

Clear the screen on every cycle

Add a call to redraw the entire background, so clearing the screen on each cycle through the `draw()` function. Add this:

```
background(0);
```

Remember to put this at the start of the `draw()` function, before anything else, so that clearing the screen is the first thing that happens in the `draw()` function.

Stop that block from escaping off the right hand side of the screen

When the block hits the right hand edge of the screen, you want it to bounce off and start heading in the opposite direction. Bouncing is easy, just change the direction of the velocity:

```
vx = -vx ;
```

Add an `if` statement to the code that makes the sketch execute the above statement only if the block hits the right hand edge of the screen. Remember that the width of the screen is stored in the variable `width`.

For example:

```
if( ??? ) {
  vx = -vx ;
}
```

You need to work out what the "`???`" should be. Make sure that the block appears to bounce off the edge of the window, rather than disappear off to the right and reappear again.

Debugging hint 1: You should set the initial value of `vx` to be a higher number than 1.0 so that you can see the effect of your `if` statement more quickly after hitting the run button.

Debugging hint 2: If you are having problems, get the sketch to print out debugging messages. For example, in the body of the `if` statement, you might have:

```
println ( "bounce at x =", x ) ;
```

Stop that block from escaping off the left hand side of the screen

Now modify the `if` statement so that it also stops the block from escaping off the left edge of the screen also. Bouncing off the edge of the screen is identical for left and right sides of the screen, just use this code:

```
vx = -vx ;
```

Stuck on how to combine the conditions for bouncing off the left side and bouncing off the right side? Cannot remember how to combine Boolean expressions? Look at the Reference page for "|| (logical OR)" on the Processing Reference page (the very bottom entry in the left hand column on https://processing.org/reference/).

Simple mouse input

As an example of using mouse input, let's change the block's colour to red whenever the mouse's x-value is inside the block. Use the built-in parameter `mouseX`, which tells you that x-value.

Write an `if` statement that does this.

For example:

```
if( ??? ) {
  fill( 255, 0, 0 ) ;
} else {
  fill( 0, 128, 255 ) ;
}
```

You need to work out what the "`???`" should be. It will likely look similar to what you did for bouncing but you will likely need to use `&&` (logical AND) instead.

Debugging hint: you are going to need to slow that block down to be able to debug this effectively.

An unbelievably simple game

The game is to click on the block. If you click on it, you stop it from moving.

Stop a block from moving by setting its velocity to zero:

```
vx = 0 ;
```

Use the `mouseClicked()` callback function to make this work (see the Processing Reference manual for a simple example):

```
void mouseClicked() {
  if( ??? ) {
    vx = 0 ;
  }
}
```

Again, you need to work out what the "`???`" should be.

This obviously stops the block completely and that is the end of the 'game'. Think about ways to make this more interesting or more challenging.

Congratulations

Armed with these simple examples, you should now be able to start work on Assignment 2.