

Event Handling

Mouse Events

Produced
by:

Department of Computing and Mathematics



Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

What is an event?

“An action such as a key being pressed,
the mouse moving, or
a new piece of data becoming available to read.

An event interrupts the normal
flow of a program to
run the code within an event block”

(Reas & Fry, 2014)

Mouse Events

Mouse Variables	Description
mousePressed	<p><i>true</i> if any mouse button is pressed, <i>false</i> otherwise.</p> <p>Note: this variable reverts to <i>false</i> as soon as the button is released.</p>
mouseButton	<p>Can have the value LEFT, RIGHT and CENTER, depending on the mouse button most recently pressed.</p> <p>Note: this variable retains its value until a <u>different</u> mouse button is pressed.</p>

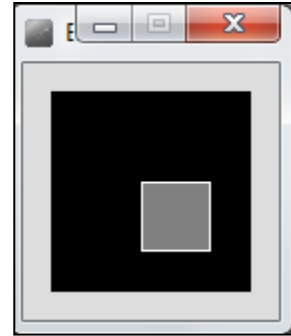
Mouse Events

- Mouse and keyboard events only work when a program has `draw()`.
- Without `draw()`, the code is only run once and then stops listening for events.

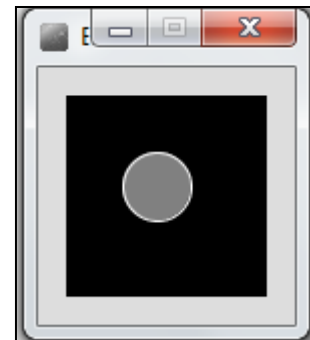
Processing Example 6.1

- Functionality:

- If the mouse is pressed, draw a gray square with a white outline.



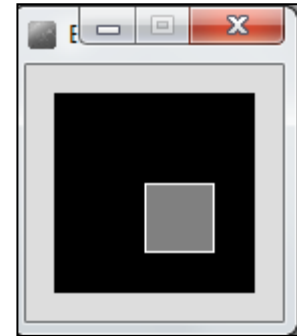
- Otherwise draw a gray circle with a white outline.



Processing Example 6.1 - Code

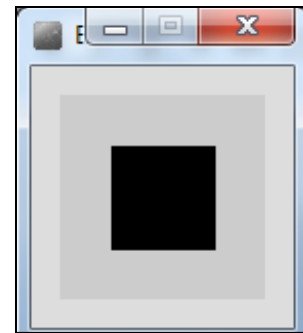
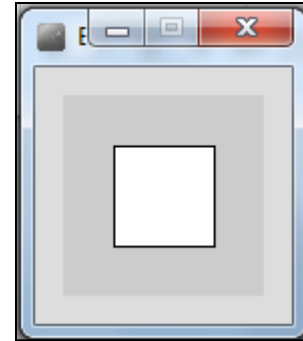
```
void setup() {  
  size(100,100);  
}
```

```
void draw() {  
  background(0);  
  stroke(255);  
  fill(128);  
  if (mousePressed){  
    rect(45,45,34,34);  
  }  
  else{  
    ellipse(45,45,34,34);  
  }  
}
```



Processing Example 6.2

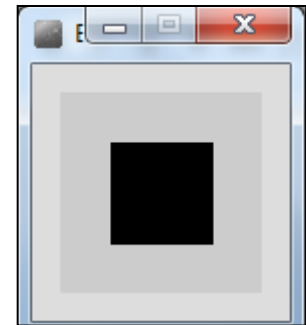
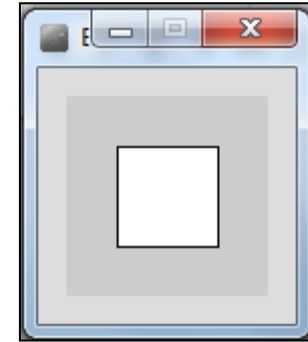
- Functionality:
 - If the mouse is pressed, set the fill to white and draw a square.
 - Otherwise set the fill to black and draw a square.



Processing Example 6.2

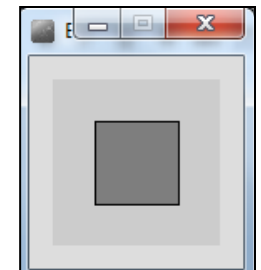
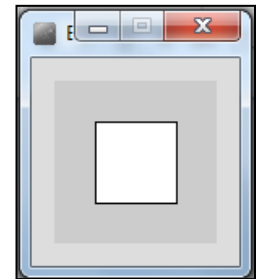
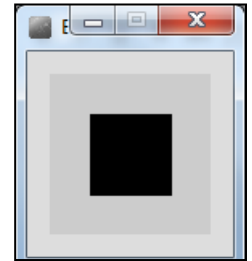
```
void setup() {  
  size(100,100);  
}
```

```
void draw() {  
  background(204);  
  if (mousePressed == true)  
  {  
    fill(255); // white  
  } else {  
    fill(0);   // black  
  }  
  rect(25, 25, 50, 50);  
}
```



Processing Example 6.3

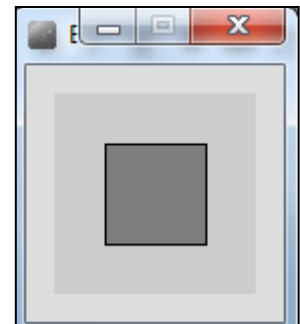
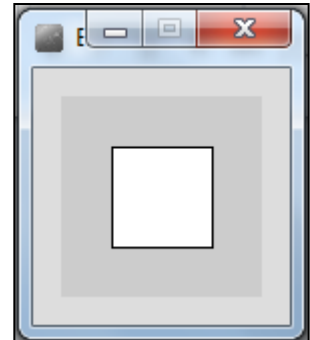
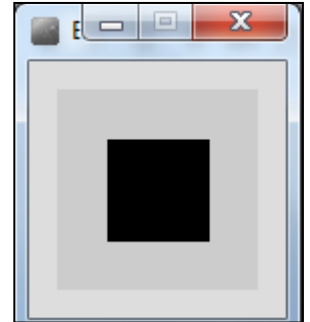
- Functionality:
 - If the LEFT button on the mouse is pressed, set the fill to black and draw a square. As soon as the LEFT button is released, gray fill the square.
 - If the RIGHT button on the mouse is pressed, set the fill to white and draw a square. As soon as the RIGHT button is released, gray fill the square.
 - If no mouse button is pressed, set the fill to gray and draw a square.



Processing Example 6.3

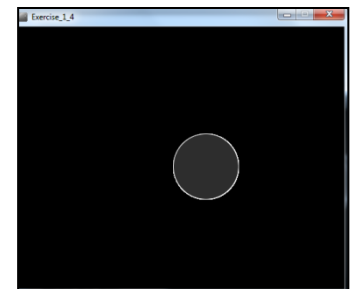
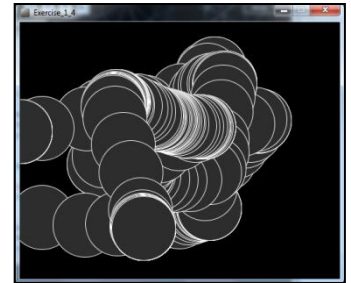
```
void setup() {  
  size(100,100);  
}
```

```
void draw() {  
  if (mousePressed){  
    if (mouseButton == LEFT)  
      fill(0);      // black  
    else if (mouseButton == RIGHT)  
      fill(255);    // white  
  }  
  else {  
    fill(126);      // gray  
  }  
  rect(25, 25, 50, 50);  
}
```



Processing Example 6.4

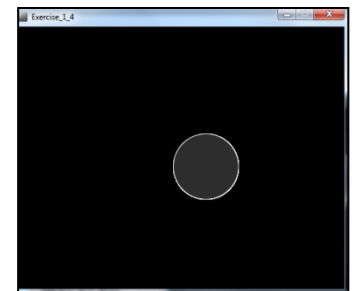
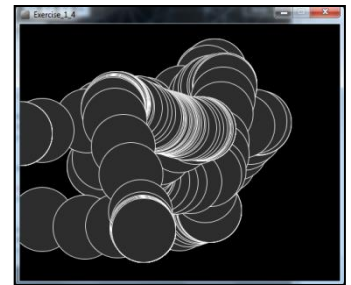
- Functionality:
 - Draw a circle on the mouse (x,y) coordinates.
 - Each time you move the mouse, draw a new circle.
 - All the circles remain in the sketch until you press a mouse button.
 - When you press a mouse button, the sketch is cleared and a single circle is drawn at the mouse (x,y) coordinates.



Processing Example 6.4

```
void setup() {  
  size(500,400);  
  background(0);  
}
```

```
void draw() {  
  
  if (mousePressed) {  
    background(0);  
  }  
  
  stroke(255);  
  fill(45,45,45);  
  ellipse(mouseX, mouseY, 100, 100);  
}
```



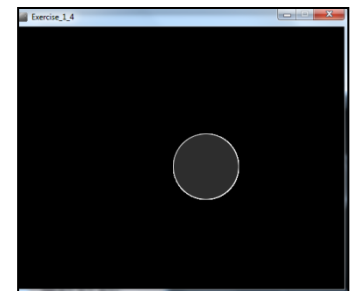
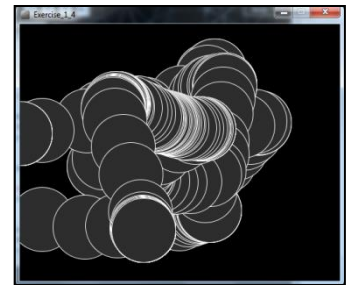
Processing Example 6.4

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

We moved the stroke and fill function calls to the setup() function.

Q: Does this change the functionality of our sketch?

```
void draw() {  
  
  if (mousePressed) {  
    background(0);  
  }  
  ellipse(mouseX, mouseY, 100, 100);  
}
```



Questions?



References

- Reas, C. & Fry, B. (2014) Processing – A Programming Handbook for Visual Designers and Artists, 2nd Edition, MIT Press, London.



Except where otherwise noted, this content is licensed under a Creative Commons Attribution-NonCommercial 3.0 License.

For more information, please see <http://creativecommons.org/licenses/by-nc/3.0/>



Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

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