Mouse Events

Note on import:

The following programs use a different form of import.
from pygame import *

What this means is that every time you use a pygame command, you do not have to type out pygame.

For example:

pygame.draw.rect(screen, RED, (250, 250, 50, 75))

is replaced by

draw.rect(screen, RED, (250, 250, 50, 75))

So watch the code carefully!!!

From the previous document we know that three different events occur in pygame that deal with the mouse.

MOUSEMOTION, MOUSEBUTTONUP, MOUSEBUTTONDOWN

Example 1

The following code will test the buttons on your mouse.

```
# the following code will always put the screen in the top corner
import os
os.environ['SDL_VIDEO_WINDOW_POS'] = "%d, %d" %(20, 20)

from pygame import *
init()
size = width, height = 800, 600
screen = display.set_mode(size)
button = 0
BLACK = (0, 0, 0)
WHITE= (255, 255, 255)
RED = (255, 0, 0)
myFont = font.SysFont("Times New Roman",30)

def drawScene(screen, button):
    if button > 0:
        draw.rect(screen, BLACK, (0, 0, width, height))
    # Creating font...we will get to that later
    string = "The last button pressed is " + str(button) + "."
    text = myFont.render(string, 1, RED)
    screen.blit(text, Rect(100, 100, 500, 500))

running = True
myClock = time.Clock()

# Game Loop
while running:
    for e in event.get():            # checks all events that happen
        if e.type == QUIT:
            running = False
        if e.type == MOUSEBUTTONDOWN:
            mx, my = e.pos
            button = e.button

    drawScene(screen, button)
    display.flip()
    myClock.tick(60)                 # waits long enough to have 60 fps

quit()
```

Take note of the buttons and their numbers on your mouse for later.

Example 2

The following code will move the output to wherever your mouse is.

```
from pygame import *
init()
size = width, height = 800, 600
screen = display.set_mode(size)
button = 0
BLACK = (0, 0, 0)
RED = (255, 255, 255)
myFont = font.SysFont("Times New Roman",30)

def drawScene(screen, mx, my, button):
    draw.rect(screen, BLACK, (0, 0, width, height))
    # Draw circle if the left mouse button is down.
    string = "The last button pressed is " + str(button) + "."
    text = myFont.render(string, 1, RED)
    size = myFont.size(string)
    screen.blit(text, (mx, my, size[0], size[1]))

running = True
myClock = time.Clock()

mx = my = 0
# Game Loop
while running:
    for e in event.get():            # checks all events that happen
        if e.type == QUIT:
            running = False
        if e.type == MOUSEBUTTONDOWN:
            mx, my = e.pos
            button = e.button
        if e.type == MOUSEMOTION:
            mx, my = e.pos
    drawScene(screen, mx, my, button)
    display.flip()
    myClock.tick(60)                 # waits long enough to have 60 fps

quit()
```

Example 3

Write a program that displays a circle on the screen every time we click the first mouse button.

```
# the following code will always put the screen in the top corner
import os
os.environ['SDL_VIDEO_WINDOW_POS'] = "%d, %d" %(20, 20)

from pygame import *
init()
size = width, height = 800, 600
screen = display.set_mode(size)
button = 0

BLACK = (0, 0, 0)
GREEN = (0, 255, 0)

def drawScene(screen, button):
    # Draw circle if the left mouse button is down.
    if button==1:
        draw.circle(screen,GREEN,(mx,my), 10)

running = True
myClock = time.Clock()

# Game Loop
while running:
    for e in event.get():           # checks all events that happen
        if e.type == QUIT:
            running = False
        if e.type == MOUSEBUTTONDOWN:
            mx, my = e.pos
            button = e.button


    drawScene(screen, button)
    display.flip()
    myClock.tick(60)                # waits long enough to have 60 fps

quit()
```

You may or may not have noticed, but there is a slight problem. When moving the mouse, I cannot click and make a circle. The mouse must be stopped.

This is due to if I don't move the mouse while I click, I get a MOUSEBUTTONDOWN. However, if I move it, I get MOUSEMOTION. Our code only used MOUSEBUTTONDOWN.

Example 4:

Write a program that fixes the previous problem.

```
# the following code will always put the screen in the top corner
import os
os.environ['SDL_VIDEO_WINDOW_POS'] = "%d, %d" %(20, 20)

from pygame import *
init()
size = width, height = 800, 600
screen = display.set_mode(size)
button = 0

BLACK = (0, 0, 0)
GREEN = (0, 255, 0)

def getVal(tup):
    """ getVal returns the (position+1) of the first 1 within a tuple.
        This is used because MOUSEBUTTONDOWN and MOUSEMOTION deal with
        mouse events differently
    """
    for i in range(3):
        if tup[i]==1:
            return i+1
    return 0

def drawScene(screen, button):
    # Draw circle if the left mouse button is down.
    if button==1:
        draw.circle(screen,GREEN,(mx,my), 10)

running = True
myClock = time.Clock()

# Game Loop
while running:
```

```
    for e in event.get():            # checks all events that happen
        if e.type == QUIT:
            running = False
        if e.type == MOUSEBUTTONDOWN:
            mx, my = e.pos
            button = e.button
        if e.type == MOUSEMOTION:
             mx, my = e.pos
             button = getVal(e.buttons)  #finds the first pushed down value

    drawScene(screen, button)
    display.flip()
    myClock.tick(60)                    # waits long enough to have 60 fps

quit()
```

This should work much better and we can now draw a circle as the mouse moves and clicks.

Other Mouse Functions:

You can also use other mouse functions that are not event driven. Some of them are as follows:

Command	Explanation
mouse.get_pressed()	- gets the states of the buttons at any time but is not tied to the events so there is a possibility we could miss a click
mouse.get_pos()	- returns the x and y of the mouse in a list
mouse.set_pos()	- used to set the x and y position of the mouse - mouse.set_pos(100, 100)
mouse.set_visible()	- used to set the visibility of the mouse - False is not visible, True is visible - mouse.set_visible(False)

Example 5:

Write a program that moves a circle with the mouse. A left mouse click hides the mouse while a right click shows it again.

```
# the following code will always put the screen in the top corner
import os
os.environ['SDL_VIDEO_WINDOW_POS'] = "%d, %d" %(20, 20)

from pygame import *
init()
size = width, height = 800, 600
screen = display.set_mode(size)
button = 0

BLACK = (0, 0, 0)
GREEN = (0, 255, 0)

def drawScene(screen, button):
    draw.rect(screen, BLACK, (0,0, width, height))
    mx, my = mouse.get_pos()
    if button == 1:
        mouse.set_visible(False)
    if button == 3:
        mouse.set_visible(True)

    draw.circle(screen,GREEN,(mx,my), 10)

running = True
myClock = time.Clock()

# Game Loop
while running:
    for e in event.get():           # checks all events that happen
        if e.type == QUIT:
            running = False
        if e.type == MOUSEBUTTONDOWN:
            button = e.button

    drawScene(screen, button)
    display.flip()
    myClock.tick(60)                # waits long enough to have 60 fps
quit()
```

Exercises

1) Write a program that toggles the colour of the circle in Example 5 from red to green with each click of the mouse.

2) Write a program that stamps a circle to the black background with each left click of the mouse. A right click clears the background.

3) Write a simple drawing program that allows the users to select a colour from a palette before drawing circles with a radius of 1 pixel on a white canvas. Modify the program to change the radius of the circle as well.