Key Events

At this point we know that there are two main events that deal with key input. They are:

KEYDOWN and KEYUP

The main one we will be using is KEYDOWN.

Example #1

The following program displays a red circle controllable by the left and right arrow keys.

```
from pygame import *

init()

RED = (255, 0, 0)
BLACK = (0,0,0)

info = display.Info()
width = 500
height = 300

SIZE = (width, height)
screen = display.set_mode(SIZE)

# function that draws everything we need onto the screen
def drawScreen(location):
  draw.rect(screen, BLACK, (0, 0, width, height))
  draw.circle(screen, RED, (location, height - 100), 10)
  display.flip()

running = True
x = width//2

while running:
  for evnt in event.get():
    if evnt.type == QUIT:
      running = False
    if evnt.type == KEYDOWN:
      keys = key.get_pressed()
      if keys[K_RIGHT]:
        x = x + 5
      if keys[K_LEFT]:
        x = x - 5

  drawScreen(x)
  display.flip()

quit()
```

The problem with the previous code is that is only moves the circle when we press the key. But what if we wanted to keep it moving when the key was pressed?

Example #2

The following program controls a red circle but now you can hold down the keys.

```
from pygame import *

init()

RED = (255, 0, 0)
BLACK = (0,0,0)

info = display.Info()
width = 500
height = 300

SIZE = (width, height)
screen = display.set_mode(SIZE)

#some game states
KEY_RIGHT = False
KEY_LEFT = False
```

```
def drawScreen(location):
  draw.rect(screen, BLACK, (0, 0, width, height))
  draw.circle(screen, RED, (location, height - 100), 10)
  display.flip()

running = True
x = width//2

while running:
  for evnt in event.get():
    if evnt.type == QUIT:
      running = False

    if evnt.type == KEYDOWN:
      if evnt.key == K_LEFT:
        KEY_LEFT = True
      if evnt.key == K_RIGHT:
        KEY_RIGHT = True

    if evnt.type == KEYUP:
      if evnt.key == K_LEFT:
        KEY_LEFT = False
      if evnt.key == K_RIGHT:
        KEY_RIGHT = False

  if KEY_LEFT == True:
    x = x - 1
  if KEY_RIGHT == True:
    x = x + 1

  drawScreen(x)
  display.flip()

quit()
```

But now there are more problems. Speed. Some computers will be faster than others. How do we control the speed?

Clock Ticks

time.wait() vs. <Clock object>.tick()

time.wait() pauses your program for a set number of milliseconds. We do this because if we don't our program will run too fast. We also do it so our program doesn't hog system resources. The only problem here is that our goal is to have a constant frame rate (number of loops/updates per second.) This loop:

```
while True:
    time.wait(20)
```

loops at a rate of 50 times per second (1000 / 20). As soon as I add anything else to the loop the frequency will drop. If my game logic + time to draw takes 5 ms, then I would have a frame rate of (1000 / (20 +5)) = 50 fps. This is made worse by the fact that on different machines or with more or less stuff going on in my game the time for game logic + drawing would be higher or lower.

<Clock object>.tick() – tick() allows you to specify the frame rate you want and wait for the amount of time you need to ensure it happens. Why an object? We need to remember when we updated last so we can figure out how long to wait.

Example #3

Add in a tick() function that will make the game run at 60 frames/second.

```
from pygame import *

init()

RED = (255, 0, 0)
BLACK = (0,0,0)

info = display.Info()
width = 500
height = 300

SIZE = (width, height)
screen = display.set_mode(SIZE)#,FULLSCREEN)

#some game states
KEY_RIGHT = False
KEY_LEFT = False

def drawScreen(location):
  draw.rect(screen, BLACK, (0, 0, width, height))
  draw.circle(screen, RED, (location, height - 100), 10)
  display.flip()

myClock = time.Clock()
running = True
x = width//2

while running:
  for evnt in event.get():
    if evnt.type == QUIT:
      running = False

    if evnt.type == KEYDOWN:
      if evnt.key == K_LEFT:
        KEY_LEFT = True
      if evnt.key == K_RIGHT:
        KEY_RIGHT = True

    if evnt.type == KEYUP:
      if evnt.key == K_LEFT:
        KEY_LEFT = False
      if evnt.key == K_RIGHT:
        KEY_RIGHT = False

  if KEY_LEFT == True:
    x = x - 1
  if KEY_RIGHT == True:
    x = x + 1

  drawScreen(x)
  display.flip()
  myClock.tick(60)

quit()
```

Much better!

Exercises:

1) Modify the last example to move the circle anywhere on the screen using all the arrow keys.

2) Modify the previous code so the circle does not move off the screen.

3) Write a program that moves an object from a starting position to a finishing position before stopping. Add some obstacles that the object has to move around. Use the arrow keys to guide the object.