Coin Tutorial:

Materials:

* Python 3.3.3
* Pygame
* LiveWires
* Background.png
* Platform.png
* Pharaoh1.bmp
* Pharaoh1\_Back.bmp
* Pharaoh2.bmp
* Pharaoh2\_Back.bmp
* Enemy.png
* Enemy\_back.png
* Coin.png

We will build:

* Coin class

We will modify:

* Game class

Instructions:

First, create the Coin class:

class Coin(games.Sprite):

IMAGE = games.load\_image("images/coin.gif")

def \_\_init\_\_(self, x, y):

super(Coin, self).\_\_init\_\_(image = Coin.IMAGE,

x = x, y = y)

self.isCollected = False

Coin is built on top of the games.Sprite class. we load the image for the Coin sprite, then define the \_\_init\_\_ method. As usual, we use the ‘super’ call to run the \_\_init\_\_ method from games.Sprite. We give Coin one variable, isCollected—we will use this to determine if a coin has been picked up by Pharaoh. (We won’t make much use of it in this tutorial, but it will come in handy later).

Next, we define the activate and deactivate methods:

def activate(self):

games.screen.add(self)

self.isCollected = False

def deactivate(self):

self.destroy()

self.isCollected = True

As we’ve seen before, activate is used to add the sprite to the game. When we do this, we want isCollected to be False—it is not yet collected. Deactivate removes the sprite from the game; this will happen when Pharaoh picks up a Coin, so we set isCollected to True.

Now, we define the update method. Here, we will handle collisions between Coin and Pharaoh.

def update(self):

super(Coin, self).update()

for sprite in self.overlapping\_sprites:

if isinstance(sprite, Pharaoh):

sprite.score += 1

sprite.refreshScore()

self.deactivate()

We start with the super call—this calls the ‘update’ method from games.Sprite. Then, we use a for loop to loop through all the sprites currently colliding with Coin. If one of these sprites is Pharaoh, we add one to the score variable in Pharaoh. Next, we need to refresh the HUD so the change in score shows up. After that, we deactivate the coin to remove it from the game.

This finishes up the Coin class. In order to test it, add a Coin object to Game:

class Game(object):

BACKGROUND = games.load\_image("images/background.png",transparent=False)

def \_\_init\_\_(self):

games.screen.background = Game.BACKGROUND

platform = Platform(x = 42,

y = games.screen.height,

direction = "HORIZONTAL",

hrd = True,

num = 10)

plat2 = Platform(x = 42,

y = games.screen.height - 60,

direction = "VERTICAL",

hrd = True,

num = 1)

plat3 = Platform(x = games.screen.width - 42,

y = games.screen.height - 60,

direction = "VERTICAL",

hrd = True,

num = 1)

pharaoh = Pharaoh(x = games.screen.width/2,

y = games.screen.height/2,

game = self)

enemy = Enemy(x = games.screen.width - 10,

y = games.screen.height/2)

**coin = Coin(x = games.screen.width/2+100,**

**y = 200)**

platform.activate()

plat2.activate()

plat3.activate()

pharaoh.activate()

enemy.activate()

**coin.activate()**

enemy.gravity = .05;

pharaoh.gravity = .05;

games.screen.mainloop()

Here are some screenshots of what you should see if you run this program: