**Animation Part I**

The basic sequence of any animation scheme is a simple loop:

1. erase image
2. change coordinates of image
3. draw image at new coordinates
4. repeat

In this lesson we will take an image and apply this animation loop. First we start with a basic image loaded and displayed on the screen:

import pygame

from pygame.locals import \*

# set the dimensions of the screen

screen\_width = 640

screen\_height = 520

size = screen\_width, screen\_height

screen = pygame.display.set\_mode(size)

green = 0, 255, 0 # create a RGB tuple

screen.fill(green) # paint screen green

# load yoshi pic and get dimensions

pic = pygame.image.load("mario yoshi.png")

pic\_width, pic\_height = pic.get\_size()

# draw pic to the screen

screen.blit(pic,(100,0)) # add pic to screen

pygame.display.flip() # flip screen onto window

We notice that the image is quite large. Before we do our animation, we will reduce the image using the python.transform.scale() method. Place the following line immediately after the line that loads the image file:

pic = pygame.transform.scale(pic,(100,100))

This scales the image down to a more reasonable 100 pixels by 100 pixels.

Now we want to create a game loop that moves the image to the right. Use the code below to replace the drawing portion of the above code (i.e. the last 3 lines of code in your program).

gameOn = True

while gameOn:

for event in pygame.event.get():

if event.type == QUIT:

gameOn = False

break

elif event.type == KEYDOWN:

pass

pygame.display.flip() # flip screen onto window

pygame.quit()

This is incomplete but we’ll fill it in next. We will write our animation code inside the KEYDOWN part of the if statement, but for now it is empty – this is the meaning of the word “pass”—do nothing until we figure out what to write there. Although pass does nothing, we need to put it in simply as an indent marker so the interpreter doesn’t complain (try removing it and see what happens).

For the animation part, we will follow our steps. The code goes in the KEYDOWN part of the if statement. First you must erase the “pass” statement (or nothing we write here will get done!). In its place, we do the following:

1. erase image

We do this by filling the screen over with green paint:

screen.fill(green) # paint screen green

1. change coordinates of image

Let’s call the coordinates yoshiX and yoshiY. To move yoshi to the right, we write:

yoshiX = yoshiX + 10

1. draw image at new coordinates

screen.blit(pic,(yoshiX,yoshiY)) # put Yoshi at new coords

pygame.display.flip() # flip screen on to window

1. repeat

The code repeats already since it is in a while loop. We are done... or are we?

Of course, we are not finished. But you can run the code and see what happens. When you press any key, Yoshi is supposed to move to the right. But the code crashes. The error message is: “yoshiX not defined”.

The cure is to define yoshiX above the loop by writing:

yoshiX = 10 # put yoshi on the left

yoshiY = 200 # and a bit down

gameOn = True

while gameOn:

Now we run the code and it should work. When we press a key, Yoshi moves to the right. It’s not pretty, but it works.

Exercise:

Adapt your code so that when you press the right arrow, Yoshi moves to the right, and when you press the left arrow key, Yoshi moves to the left.