Animation Part II

So far we have a poor man’s version of an animation. But frankly, if I may, it is terrible. I wouldn’t pay to play this game; I would pay to not have to play this game. It is that terrible.

In this lesson we will get Yoshi to move without continually hitting a key. We start where we left off in the last lesson. Here’s what you should have:

import pygame

from pygame.locals import \*

screen\_width = 640

screen\_height = 520

size = screen\_width, screen\_height

screen = pygame.display.set\_mode(size) # set the dimensions of screen

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

screen.fill(green) # paint screen green

# load yoshi pic and figure out dimensions of each slice

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

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

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

yoshiX = 20

yoshiY = 100

gameOn = True

while gameOn:

for event in pygame.event.get():

if event.type == QUIT:

sys.exit()

elif event.type == KEYDOWN:

if event.key == K\_RIGHT:

yoshiX = yoshiX + 10

elif event.key == K\_LEFT:

yoshiX = yoshiX - 10

screen.fill(green) # erase old pic

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

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

Now we add a variable called speed. Just above the “gameOn = True” statement, we put:

speed = 0

gameOn = True

The speed is how many pixels Yoshi will move at a time. Now we change this:

if event.key == K\_RIGHT:

yoshiX = yoshiX + 10

to this:

if event.key == K\_RIGHT:

speed = 10

We want to do the same (well, not exactly the same!) thing for the left key.

Finally, we will need to put back in the part that gets Yoshi to move. This part must be inside the game loop but not be inside the event part because Yoshi must move regardless of the events going on. We place the line above the line that paints the screen green, like so:

yoshiX = yoshiX + speed

screen.fill(green) # erase old pic

**Exercises**

1. Yoshi is a little fast. Slow him down.
2. It would be nice if we could make Yoshi stop. Here is my suggestion:

if Yoshi is stopped when you press an arrow key, make him go.

if Yoshi is going left when you press the right arrow key, make him stop.

If Yoshi is going right when you press the left arrow key, make him stop.

If Yoshi is going left when you press the left key, don’t do anything. Ditto for the right.

1. It would be nice if Yoshi actually faced left when he goes left. To do this, you’ll need to create a second image. So this line:

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

becomes this :

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

picLeft = pygame.transform.flip(pic,False,True)

Now that you have two images, adjust your code so that the correct one is being drawn – left when he’s going left, right when going right.

1. Add a fireball-firing capability, triggered by the space bar. I have included a fireball image and a sound file. Make sure that the fireball can fire in both directions. Note: If you are unsatisfied with the sound file that I have provided you with, [www.soundbible.com](http://www.soundbible.com) is an excellent resource where you can waste countless hours. Please do this on your own time!