Activity one 1-5

#Drawing of Colorful Shapes

#Gavriel Feuer

#9/17/13

#--------------------Notes--------------------------

import pygame

BLACK = ( 0, 0, 0)

WHITE = (255,255,255)

RED = (255, 0, 0)

BLUE = ( 0, 0,255)

#color added V

CYAN=(65,255,243)

GREEN = ( 0,255, 0)

pygame.init()

#size change V

size\_x = 300

size\_y = 500

size = [size\_x, size\_y]

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

pygame.display.set\_caption("Retro Screensaver")

clock=pygame.time.Clock()

radius = 50

#line wigth changed V

line\_width = 10

position\_x = 100

position\_y = 100

rungame=True

while rungame:

for event in pygame.event.get():

if event.type == pygame.QUIT:

rungame=False

screen.fill(WHITE)

pygame.draw.circle(screen, GREEN, [position\_x+5,position\_y+5], radius, line\_width)

#color changed V

pygame.draw.circle(screen, CYAN, [position\_x, 10+position\_y], radius, line\_width )

pygame.draw.circle(screen, BLACK, [1+position\_x, -10+position\_y], 10, 0 )

pygame.draw.line(screen,RED,[position\_x,25+position\_y],[position\_x,position\_y],10)

pygame.draw.line(screen,RED,[position\_x,8+position\_y],[-20+position\_x,10+position\_y],10)

pygame.draw.line(screen,RED,[position\_x,8+position\_y],[ 20+position\_x,10+position\_y],10)

#leg changeV

pygame.draw.line(screen,BLUE,[position\_x-2,300+position\_y],[-8+position\_x,500+position\_y],6)

pygame.draw.line(screen,BLUE,[position\_x+2,25+position\_y],[ 8+position\_x,45+position\_y],6)

clock.tick(60)

pygame.display.flip()

pygame.quit()

Topic 2

#Drawing of Colorful Shapes

#Gavriel Feuer

#9/17/13

#--------------------Notes--------------------------

import pygame

BLACK = ( 0, 0, 0)

WHITE = (255,255,255)

RED = (255, 0, 0)

BLUE = ( 0, 0,255)

GREEN = ( 0,255, 0)

#new colors

cyan=(0,225,225)

brown=(156,86,0)

something=(0,114,0)

pygame.init()

size\_x = 700

size\_y = 500

size = [size\_x, size\_y]

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

pygame.display.set\_caption("Retro Screensaver")

clock=pygame.time.Clock()

radius = 50

line\_width = 2

position\_x = 100

position\_y = 100

rungame=True

while rungame:

for event in pygame.event.get():

if event.type == pygame.QUIT:

rungame=False

#new color used

screen.fill(brown)

pygame.draw.circle(screen, GREEN, [position\_x+5,position\_y+5], radius, line\_width)

# new color used

pygame.draw.circle(screen, cyan, [position\_x, 10+position\_y], radius, line\_width )

#new color used

pygame.draw.circle(screen, something, [1+position\_x, -10+position\_y], 10, 0 )

pygame.draw.line(screen,RED,[position\_x,25+position\_y],[position\_x,position\_y],10)

pygame.draw.line(screen,RED,[position\_x,8+position\_y],[-20+position\_x,10+position\_y],4)

pygame.draw.line(screen,RED,[position\_x,8+position\_y],[ 20+position\_x,10+position\_y],4)

pygame.draw.line(screen,BLUE,[position\_x-2,25+position\_y],[-8+position\_x,45+position\_y],6)

pygame.draw.line(screen,BLUE,[position\_x+2,25+position\_y],[ 8+position\_x,45+position\_y],6)

clock.tick(60)

pygame.display.flip()

pygame.quit()

Topic 3

I changed the spped multiple times for the velocity part here are those changes

Original V

speed\_x = 3

speed\_y = 3

1

speed\_x = 3

speed\_y = 0

2

speed\_x = -3

speed\_y = 0

3

speed\_x = 0

speed\_y = -3

4

speed\_x = 0

speed\_y = 3

5

speed\_x = 6

speed\_y = 0

6

speed\_x = 3

speed\_y = -3

here is 7-9

#Drawing and Movement of Colorful Shapes

#Gavriel Feuer

#9/17/13

#--------------------Notes-------------------------

#--------------------------------------------------

import pygame

#Define color palette

# r g b

BLACK = ( 0, 0, 0)

WHITE = (255,255,255)

RED = (255, 0, 0)

GREEN = ( 0,255, 0)

BLUE = ( 0, 0,255)

#Initialize the pygame library

pygame.init()

#Set the size of the display canvas: Q-What are the units for the size?

size\_x = 700

size\_y = 500

size = [size\_x, size\_y]

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

pygame.display.set\_caption("Retro Screensaver")

#Set the clock to manage how fast the screen updates

clock=pygame.time.Clock()

#Set up the shape parameters

radius = 50

line\_width = 3

#set the starting position of our shape

position\_x = 100

position\_y = 100

#set the speed and direction of the shape

speed\_x = 3

speed\_y = 3

#This is the driving force for the program

rungame=True

#------------Main Program Loop -----------------

while rungame:

for event in pygame.event.get():

if event.type == pygame.QUIT:

rungame=False

#set the background color: Q- what would happen if this line was not included?

screen.fill(WHITE)

#Draw the shape: Learn to look at the reference manual. DIY

#http://www.pygame.org/docs/ref/draw.html

# Outer circle

pygame.draw.circle(screen, GREEN, [position\_x, 10+position\_y], radius, line\_width )

# Head

pygame.draw.circle(screen, BLACK, [1+position\_x, -10+position\_y], 10, 0 )

# Body

pygame.draw.line(screen,RED,[position\_x,25+position\_y],[position\_x,position\_y],10)

# Arms

pygame.draw.line(screen,RED,[position\_x,8+position\_y],[-20+position\_x,10+position\_y],4)

pygame.draw.line(screen,RED,[position\_x,8+position\_y],[ 20+position\_x,10+position\_y],4)

# Legs

pygame.draw.line(screen,BLUE,[position\_x-2,25+position\_y],[-8+position\_x,45+position\_y],6)

pygame.draw.line(screen,BLUE,[position\_x+2,25+position\_y],[ 8+position\_x,45+position\_y],6)

#Move the shape

position\_x = position\_x + speed\_x

position\_y = position\_y + speed\_y

#Bounce the ball if it hits a wall or an obstacle

# 7 V

# if position\_x > size\_x-radius or position\_x < 0+radius:

#speed\_x = -1\*speed\_x

# if position\_y > size\_y-radius or position\_y < 0+radius:

# speed\_y = -1\*speed\_y

#Set the clock speed [frames per second]

clock.tick(60)

#Update the screen with the new drawing

pygame.display.flip()

pygame.quit()

8

if position\_x > size\_x-radius or position\_x < 0+radius:

speed\_x = -1\*speed\_x

speed\_x+=1

if position\_y > size\_y-radius or position\_y < 0+radius:

speed\_y = -1\*speed\_y

speed\_y+=1

9

#I changed radius

Radius=50

rad\_new=Radius/2

if position\_x > size\_x-rad\_new or position\_x < 0+rad\_new:

speed\_x = -1\*speed\_x

speed\_x += 1

if position\_y > size\_y-radius or position\_y < 0+radius:

speed\_y = -1\*speed\_y

speed\_y += 1

topic 4

1

#Drawing and Movement with Sound

#Gavriel Feuer

#9/17/13

#--------------------Notes-------------------------

# -make sure the sound file is in the same folder as the bounce\_player\_sound.py file

#--------------------------------------------------

import pygame

#Define color palette

# r g b

BLACK = ( 0, 0, 0)

WHITE = (255,255,255)

RED = (255, 0, 0)

GREEN = ( 0,255, 0)

BLUE = ( 0, 0,255)

#Initialize the pygame library

pygame.init()

#Set the size of the display canvas: Q-What are the units for the size?

size\_x = 700

size\_y = 500

size = [size\_x, size\_y]

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

pygame.display.set\_caption("Retro Screensaver")

# Sounds

# new sound

bounce\_sound = pygame.mixer.Sound("./flyby.wav")

#Set the clock to manage how fast the screen updates

clock=pygame.time.Clock()

#Set up the shape parameters

radius = 50

line\_width = 3

#set the starting position of our shape

position\_x = 100

position\_y = 100

#set the speed and direction of the shape

speed\_x = 3

speed\_y = 3

#This is the driving force for the program

rungame=True

#------------Main Program Loop -----------------

while rungame:

for event in pygame.event.get():

if event.type == pygame.QUIT:

rungame=False

#set the background color: Q- what would happen if this line was not included?

screen.fill(WHITE)

#Draw the shape: Learn to look at the reference manual. DIY

#http://www.pygame.org/docs/ref/draw.html

# Outer circle

pygame.draw.circle(screen, GREEN, [position\_x, 10+position\_y], radius, line\_width)

# Head

pygame.draw.circle(screen, BLACK, [1+position\_x, -10+position\_y], 10, 0 )

# Body

pygame.draw.line(screen,RED,[position\_x,25+position\_y],[position\_x,position\_y],10)

# Arms

pygame.draw.line(screen,RED,[position\_x,8+position\_y],[-20+position\_x, 10+position\_y],4)

pygame.draw.line(screen,RED,[position\_x,8+position\_y],[ 20+position\_x, 10+position\_y],4)

# Legs

pygame.draw.line(screen,BLUE,[position\_x-2,25+position\_y],[-8+position\_x,45+position\_y],6)

pygame.draw.line(screen,BLUE,[position\_x+2,25+position\_y],[ 8+position\_x,45+position\_y],6)

#Move the shape

position\_x = position\_x + speed\_x

position\_y = position\_y + speed\_y

#Bounce the ball if it hits a wall or an obstacle

if position\_x > size\_x-radius or position\_x < 0+radius:

speed\_x = -1\*speed\_x

bounce\_sound.play()

if position\_y > size\_y-radius or position\_y < 0+radius:

speed\_y = -1\*speed\_y

bounce\_sound.play()

#Set the clock speed [frames per second]

clock.tick(60)

#Update the screen with the new drawing

pygame.display.flip()

pygame.quit()

2

#Drawing and Movement with Sound

#Gavriel Feuer

#9/17/13

#--------------------Notes-------------------------

# -make sure the sound file is in the same folder as the bounce\_player\_sound.py file

#--------------------------------------------------

import pygame

import numpy

#Define color palette

# r g b

BLACK = ( 0, 0, 0)

WHITE = (255,255,255)

RED = (255, 0, 0)

GREEN = ( 0,255, 0)

BLUE = ( 0, 0,255)

#Initialize the pygame library

pygame.init()

#Set the size of the display canvas: Q-What are the units for the size?

size\_x = 700

size\_y = 500

size = [size\_x, size\_y]

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

pygame.display.set\_caption("Retro Screensaver")

# Sounds

# new sound

bounce\_sound = pygame.mixer.Sound("./flyby.wav")

#Set the clock to manage how fast the screen updates

clock=pygame.time.Clock()

#Set up the shape parameters

radius = 50

line\_width = 3

#set the starting position of our shape

position\_x = 100

position\_y = 100

#set the speed and direction of the shape

speed\_x = 3

speed\_y = 3

#freq change v

Fs=44100

pygame.mixer.init(Fs,-16,1)

time=5

length=Fs\*time

##freq=600.0

freq1=800.0

##freq2=600.0

##freq3=460.0

#freq1=220.0\*(4.0/3.0)

#freq2=17.5

amplitude=500.0

tmp=[]

for t in range(int(length)):

v1=amplitude\*numpy.sin(t\*freq1/Fs\*2\*numpy.pi)

## v2=amplitude\*numpy.sin(t\*freq2/Fs\*2\*numpy.pi)

## v2=amp\*numpy.sin(t\*freq3/Fs\*2\*numpy.pi)

# v1=1+0.2\*numpy.sin(t\*freq1/Fs\*2\*numpy.pi)

# v2=amplitude\*numpy.sin(t\*freq2/Fs\*2\*numpy.pi\*v1)

# v=v1\*v2

v=v1

tmp.append(v1)

sound=pygame.sndarray.make\_sound(numpy.array(tmp,numpy.int16))

#This is the driving force for the program

rungame=True

#------------Main Program Loop -----------------

while rungame:

for event in pygame.event.get():

if event.type == pygame.QUIT:

rungame=False

#set the background color: Q- what would happen if this line was not included?

screen.fill(WHITE)

#Draw the shape: Learn to look at the reference manual. DIY

#http://www.pygame.org/docs/ref/draw.html

# Outer circle

pygame.draw.circle(screen, GREEN, [position\_x, 10+position\_y], radius, line\_width)

# Head

pygame.draw.circle(screen, BLACK, [1+position\_x, -10+position\_y], 10, 0 )

# Body

pygame.draw.line(screen,RED,[position\_x,25+position\_y],[position\_x,position\_y],10)

# Arms

pygame.draw.line(screen,RED,[position\_x,8+position\_y],[-20+position\_x, 10+position\_y],4)

pygame.draw.line(screen,RED,[position\_x,8+position\_y],[ 20+position\_x, 10+position\_y],4)

# Legs

pygame.draw.line(screen,BLUE,[position\_x-2,25+position\_y],[-8+position\_x,45+position\_y],6)

pygame.draw.line(screen,BLUE,[position\_x+2,25+position\_y],[ 8+position\_x,45+position\_y],6)

#Move the shape

position\_x = position\_x + speed\_x

position\_y = position\_y + speed\_y

#Bounce the ball if it hits a wall or an obstacle

if position\_x > size\_x-radius or position\_x < 0+radius:

speed\_x = -1\*speed\_x

bounce\_sound.play()

if position\_y > size\_y-radius or position\_y < 0+radius:

speed\_y = -1\*speed\_y

tmp.append(v1)

sound=pygame.sndarray.make\_sound(numpy.array(tmp,numpy.int16))

#Set the clock speed [frames per second]

clock.tick(60)

#Update the screen with the new drawing

pygame.display.flip()

pygame.quit()

3

#Generating sounds in Pygame using numpy

#Gavriel Feuer

#8/25/13

#--------------------------------------------------

##-Before Running this program make sure numpy is installed

#(https://pypi.python.org/pypi/numpy)

##-additional information on generating sounds is available here

#(http://web.media.mit.edu/~nvawter/otherProduct/keyboard5.py)

#--------------------------------------------------

import numpy

import pygame

Fs=44100

pygame.mixer.init(Fs,-16,1)

pygame.init()

pygame.display.set\_mode((800,600))

time=5

length=Fs\*time

##freq=600.0

#freq1=800.0

##freq2=600.0

#freq3=460.0

#here V

freq1=220.0\*(4.0/3.0)

#freq2=17.5

amplitude=500.0

tmp=[]

for t in range(int(length)):

# v1=amplitude\*numpy.sin(t\*freq1/Fs\*2\*numpy.pi)

## v2=amplitude\*numpy.sin(t\*freq2/Fs\*2\*numpy.pi)

#4 V

v2=amp\*numpy.sin(t\*freq3/Fs\*2\*numpy.pi)

# v1=1+0.2\*numpy.sin(t\*freq1/Fs\*2\*numpy.pi)

# v2=amplitude\*numpy.sin(t\*freq2/Fs\*2\*numpy.pi\*v1)

# v=v1\*v2

v=v1

tmp.append(v1)

sound=pygame.sndarray.make\_sound(numpy.array(tmp,numpy.int16))

rungame=True

while rungame:

for event in pygame.event.get():

if event.type == pygame.QUIT:

rungame=False

sound.play()

pygame.quit()

5

(done)

6

freq1=800.0+freq2=600.0=freq\_new=1400.0

7

tmp.append(v1+v2)

sound=pygame.sndarray.make\_sound(numpy.array(tmp,numpy.int16))

8

freq1=800.0\*freq2=600.0=freq\_new=480,000.0

and

tmp.append(v1\*v2)

sound=pygame.sndarray.make\_sound(numpy.array(tmp,numpy.int16))