#########################################

# File Name: image\_rotation.py

# Description: Demonstrates how to rotate an image while keeping its center and size

# See example here: http://www.pygame.org/wiki/RotateCenter?parent=CookBook

# Author: ICS2O

# Date: 12/12/2017

#########################################

import pygame

pygame.init()

WIDTH = 800

HEIGHT = 600

gameWindow=pygame.display.set\_mode((WIDTH,HEIGHT))

BLACK = ( 0, 0, 0)

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

# functions #

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

def redrawGameWindow():

gameWindow.blit(ship, (shipX,shipY))

pygame.display.update()

def rotate(image, angle):

# the image MUST be square and the background should be removed

# open ship.png to inspect it

# use Photoshop or alternatives to remove the background of your pictures

ORIGINALrect = image.get\_rect()

rotatedImage = pygame.transform.rotate(image,angle)

rotatedRect = ORIGINALrect.copy()

rotatedRect.center = rotatedImage.get\_rect().center

rotatedImage = rotatedImage.subsurface(rotatedRect).copy()

return rotatedImage

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

# main program #

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

ORIGINALship = pygame.image.load("Pendulum(1).png")

ORIGINALship = pygame.transform.scale(ORIGINALship,(400,400))

backGround = pygame.image.load("BackGround.jpg")

backGround = pygame.transform.scale(backGround,(800,600))

ship = ORIGINALship.copy() # keep the original image intact, so it does not get distorted

shipX = 10

shipY = 100

shipAngle = 0

rotationStep = 10

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

print "Hit ESC to end the program."

clock = pygame.time.Clock()

FPS = 30

inPlay = True

while inPlay:

gameWindow.blit(backGround,(0,0))

redrawGameWindow()

#pygame.draw.circle(gameWindow, (93,183,235),(10,100),175,10)

clock.tick(FPS)

pygame.event.get()

keys = pygame.key.get\_pressed()

if keys[pygame.K\_ESCAPE]:

inPlay = False

shipAngle = shipAngle + rotationStep

ship = rotate(ORIGINALship,shipAngle)

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

pygame.quit()

Gucci input code:

import pygame

print "Enter your username"

pygame.init()

WIDTH = 800

HEIGHT= 600

gameWindow=pygame.display.set\_mode((WIDTH,HEIGHT))

GRIDSIZE=10

RED =(255, 0, 0)

GREEN=( 0,255, 0)

BLUE =( 0, 0,255)

CYAN =( 0,255,255)

WHITE=(255,255,255)

BLACK=( 0, 0, 0)

GREY =(128,128,128)

pygame.mixer.music.load("DoubleAgent.mp3")

def grid():

for x in range(0,WIDTH,GRIDSIZE):

pygame.draw.line(gameWindow, CYAN, (x,0),(x,HEIGHT),1)

for y in range(0,HEIGHT,GRIDSIZE):

pygame.draw.line(gameWindow, CYAN, (0,y),(WIDTH,y),1)

for x in range(0,WIDTH,10\*GRIDSIZE):

pygame.draw.line(gameWindow, GREY, (x,0),(x,HEIGHT),2)

for y in range(0,HEIGHT,10\*GRIDSIZE):

pygame.draw.line(gameWindow, GREY, (0,y),(WIDTH,y),2)

def backGround():

picture = pygame.image.load('Background(1).jpg')

picture=pygame.transform.scale(picture,(800,600))

gameWindow.blit(picture,(0,0))

running = True

def main():

backGround()

#grid()

font = pygame.font.Font(None, 32)

clock = pygame.time.Clock()

input\_box = pygame.Rect(460, 230, 400, 50)

color\_inactive = pygame.Color('lightskyblue3')

color\_active = pygame.Color('dodgerblue2')

color = color\_inactive

active = False

main.text = ''

done = False

while not done:

for event in pygame.event.get():

if event.type == pygame.QUIT:

done = True

if event.type == pygame.MOUSEBUTTONDOWN:

# If the user clicked on the input\_box rect.

if input\_box.collidepoint(event.pos):

# Toggle the active variable.

active = not active

else:

active = False

# Change the current color of the input box.

color = color\_active if active else color\_inactive

if event.type == pygame.KEYDOWN:

if active:

if event.key == pygame.K\_RETURN:

print 'your username has been set to:', main.text

done = True

elif event.key == pygame.K\_BACKSPACE:

main.text = main.text[:-1]

else:

main.text += event.unicode

# Render the current text.

txt\_surface = font.render(main.text, True, color)

# Resize the box if the text is too long.

width = max(300, txt\_surface.get\_width()+10)

input\_box.w = width

backGround()

# Blit the text.

gameWindow.blit(txt\_surface, (input\_box.x+5, input\_box.y+5))

# Blit the input\_box rect.

pygame.draw.rect(gameWindow, color, input\_box, 2)

pygame.display.flip()

clock.tick(30)

#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# PROGRAM BEGINS HERE

#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

count=0

pygame.mixer.music.play(-1)

pygame.mixer.music.set\_volume(0.5)

while running:

if count==0:

main()

print (main.text)

count=1

gameWindow.fill((255,255,255))

pygame.event.get()

keys=pygame.key.get\_pressed()

if keys[pygame.K\_ESCAPE]:

running= False

pygame.display.update()

pygame.quit()

#########################################################################

DECEMBER 19

############################################################################

#########################################  
# File Name: image\_rotation.py  
# Description: Demonstrates how to rotate an image while keeping its center and size  
# See example here: http://www.pygame.org/wiki/RotateCenter?parent=CookBook  
# Author: ICS2O  
# Date: 12/12/2017  
#########################################  
import pygame  
pygame.init()  
WIDTH = 800  
HEIGHT = 600  
gameWindow=pygame.display.set\_mode((WIDTH,HEIGHT))  
  
BLACK = ( 0, 0, 0)  
WHITE = (255,255,255)  
#---------------------------------------#  
# functions #  
#---------------------------------------#  
def redrawGameWindow():  
 gameWindow.blit(ship, (shipX,shipY))  
 pygame.display.update()  
  
def rotate(image, angle):  
 # the image MUST be square and the background should be removed  
 # open ship.png to inspect it  
 # use Photoshop or alternatives to remove the background of your pictures   
 ORIGINALrect = image.get\_rect()  
 rotatedImage = pygame.transform.rotate(image,angle)  
 rotatedRect = ORIGINALrect.copy()  
 rotatedRect.center = rotatedImage.get\_rect().center  
 rotatedImage = rotatedImage.subsurface(rotatedRect).copy()  
 return rotatedImage  
  
def play():  
 gameMenu = pygame.image.load("GameMenu.jpg")  
 gameMenu = pygame.transform.scale(gameMenu,(800,600))  
 gameWindow.blit(gameMenu,(0,0))  
 pygame.display.update()  
def tutorial():  
 gameWindow.fill(WHITE)  
#Music/Pictures  
#---------------------------------------------#  
ORIGINALship = pygame.image.load("Pendulum(2).png")  
ORIGINALship = pygame.transform.scale(ORIGINALship,(400,400))  
backGround = pygame.image.load("BackGround(2).jpg")  
backGround = pygame.transform.scale(backGround,(800,600))  
pygame.mixer.music.load("DoubleAgent.mp3")  
ship = ORIGINALship.copy() # keep the original image intact, so it does not get distorted   
shipX = 10  
shipY = 100  
shipAngle = 0  
rotationStep = 10  
  
#---------------------------------------#  
print "Hit ESC to end the program."  
clock = pygame.time.Clock()  
FPS = 30  
  
pygame.mixer.music.play(-1)  
pygame.mixer.music.set\_volume(0.5)  
inPlay = True  
(mouseX, mouseY) = pygame.mouse.get\_pos()  
while inPlay:  
 gameWindow.blit(backGround,(0,0))  
 redrawGameWindow()  
 clock.tick(FPS)  
 pygame.event.get()   
 keys = pygame.key.get\_pressed()   
 if keys[pygame.K\_ESCAPE]:   
 inPlay = False  
 for event in pygame.event.get():  
 if event.type == pygame.MOUSEBUTTONDOWN and mouseX <=710 and mouseX>=390 and mouseY >=170 and mouseY <= 240:  
 play()  
 break  
 elif event.type == pygame.MOUSEBUTTONDOWN and mouseX <=730 and mouseX>=400 and mouseY >=260 and mouseY <= 320:  
 tutorial()  
 elif event.type == pygame.MOUSEBUTTONDOWN and mouseX <=710 and mouseX>=390 and mouseY >=340 and mouseY <= 410:  
 break  
 shipAngle = shipAngle + rotationStep  
 ship = rotate(ORIGINALship,shipAngle)  
 mouseX, mouseY = pygame.mouse.get\_pos()  
  
#---------------------------------------#   
pygame.quit()  
 ######################

#DECEMBER 19 LPG 2

#########################################

# File Name: image\_rotation.py

# Description: Demonstrates how to rotate an image while keeping its center and size

# See example here: http://www.pygame.org/wiki/RotateCenter?parent=CookBook

# Author: ICS2O

# Date: 12/12/2017

#########################################

import pygame

pygame.init()

WIDTH = 800

HEIGHT = 600

gameWindow=pygame.display.set\_mode((WIDTH,HEIGHT))

BLACK = ( 0, 0, 0)

WHITE = (255,255,255)

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

# functions #

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

def redrawGameWindow():

gameWindow.blit(ship, (shipX,shipY))

pygame.display.update()

def rotate(image, angle):

# the image MUST be square and the background should be removed

# open ship.png to inspect it

# use Photoshop or alternatives to remove the background of your pictures

ORIGINALrect = image.get\_rect()

rotatedImage = pygame.transform.rotate(image,angle)

rotatedRect = ORIGINALrect.copy()

rotatedRect.center = rotatedImage.get\_rect().center

rotatedImage = rotatedImage.subsurface(rotatedRect).copy()

return rotatedImage

def play():

gameMenu = pygame.image.load("GameMenu.jpg")

gameMenu = pygame.transform.scale(gameMenu,(800,600))

gameWindow.blit(gameMenu,(0,0))

pygame.display.update()

pygame.mixer.music.load("DoubleAgent.mp3")

count=0

pygame.mixer.music.play(-1)

pygame.mixer.music.set\_volume(0.5)

def backGround():

picture = pygame.image.load('Background(1).jpg')

picture=pygame.transform.scale(picture,(800,600))

gameWindow.blit(picture,(0,0))

running = True

def main():

backGround()

#grid()

font = pygame.font.Font(None, 32)

clock = pygame.time.Clock()

input\_box = pygame.Rect(460, 230, 400, 50)

color\_inactive = pygame.Color('lightskyblue3')

color\_active = pygame.Color('dodgerblue2')

color = color\_inactive

active = False

main.text = ''

done = False

while not done:

for event in pygame.event.get():

if event.type == pygame.QUIT:

done = True

if event.type == pygame.MOUSEBUTTONDOWN:

# If the user clicked on the input\_box rect.

if input\_box.collidepoint(event.pos):

# Toggle the active variable.

active = not active

else:

active = False

# Change the current color of the input box.

color = color\_active if active else color\_inactive

if event.type == pygame.KEYDOWN:

if active:

if event.key == pygame.K\_RETURN:

print 'your username has been set to:', main.text

done = True

elif event.key == pygame.K\_BACKSPACE:

main.text = main.text[:-1]

else:

main.text += event.unicode

# Render the current text.

txt\_surface = font.render(main.text, True, color)

# Resize the box if the text is too long.

width = max(300, txt\_surface.get\_width()+10)

input\_box.w = width

backGround()

# Blit the text.

gameWindow.blit(txt\_surface, (input\_box.x+5, input\_box.y+5))

# Blit the input\_box rect.

pygame.draw.rect(gameWindow, color, input\_box, 2)

pygame.display.flip()

clock.tick(30)

while running:

if count==0:

main()

print (main.text)

count=1

gameWindow.fill((255,255,255))

pygame.event.get()

keys=pygame.key.get\_pressed()

if keys[pygame.K\_ESCAPE]:

running= False

pygame.display.update()

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

def mainMenu():

gameWindow.blit(backGround,(0,0))

redrawGameWindow()

clock.tick(FPS)

pygame.event.get()

keys = pygame.key.get\_pressed()

if keys[pygame.K\_ESCAPE]:

inPlay = False

for event in pygame.event.get():

if event.type == pygame.MOUSEBUTTONDOWN and mouseX <=710 and mouseX>=390 and mouseY >=170 and mouseY <= 240:

play()

break

elif event.type == pygame.MOUSEBUTTONDOWN and mouseX <=730 and mouseX>=400 and mouseY >=260 and mouseY <= 320:

tutorial()

break

elif event.type == pygame.MOUSEBUTTONDOWN and mouseX <=710 and mouseX>=390 and mouseY >=340 and mouseY <= 410:

break

shipAngle = shipAngle + rotationStep

ship = rotate(ORIGINALship,shipAngle)

mouseX, mouseY = pygame.mouse.get\_pos()

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

def gameMenu():

def gamePlay():

def tutorial():

def shop():

def tutorial():

gameWindow.fill(WHITE)

#Music/Pictures

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

ORIGINALship = pygame.image.load("Pendulum(2).png")

ORIGINALship = pygame.transform.scale(ORIGINALship,(400,400))

backGround = pygame.image.load("BackGround(2).jpg")

backGround = pygame.transform.scale(backGround,(800,600))

pygame.mixer.music.load("DoubleAgent.mp3")

ship = ORIGINALship.copy() # keep the original image intact, so it does not get distorted

shipX = 10

shipY = 100

shipAngle = 0

rotationStep = 10

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

print "Hit ESC to end the program."

clock = pygame.time.Clock()

FPS = 30

pygame.mixer.music.play(-1)

pygame.mixer.music.set\_volume(0.5)

inPlay = True

inplay2=True

(mouseX, mouseY) = pygame.mouse.get\_pos()

while inPlay:

main()

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

pygame.quit()

Gucci input(1)

**import** pygame

**print "Enter your username"**

pygame.init()

WIDTH = 800

HEIGHT= 600

gameWindow=pygame.display.set\_mode((WIDTH,HEIGHT))

GRIDSIZE=10

RED =(255, 0, 0)

GREEN=( 0,255, 0)

BLUE =( 0, 0,255)

CYAN =( 0,255,255)

WHITE=(255,255,255)

BLACK=( 0, 0, 0)

GREY =(128,128,128)

pygame.mixer.music.load(**"DoubleAgent.mp3"**)

**def** backGround():

picture = pygame.image.load(**'Background(1).jpg'**)

picture=pygame.transform.scale(picture,(800,600))

gameWindow.blit(picture,(0,0))

running = True

**def** main():

backGround()

*#grid()*

font = pygame.font.Font(None, 32)

clock = pygame.time.Clock()

input\_box = pygame.Rect(460, 230, 400, 50)

color\_inactive = pygame.Color(**'lightskyblue3'**)

color\_active = pygame.Color(**'dodgerblue2'**)

color = color\_inactive

active = False

main.text = **''**

done = False

**while not** done:

**for** event **in** pygame.event.get():

**if** event.type == pygame.QUIT:

done = True

**if** event.type == pygame.MOUSEBUTTONDOWN:

*# If the user clicked on the input\_box rect.*

**if** input\_box.collidepoint(event.pos):

*# Toggle the active variable.*

active = **not** active

**else**:

active = False

*# Change the current color of the input box.*

color = color\_active **if** active **else** color\_inactive

**if** event.type == pygame.KEYDOWN:

**if** active:

**if** event.key == pygame.K\_RETURN:

**print 'your username has been set to:'**, main.text

done = True

**elif** event.key == pygame.K\_BACKSPACE:

main.text = main.text[:-1]

**else**:

main.text += event.unicode

*# Render the current text.*

txt\_surface = font.render(main.text, True, color)

*# Resize the box if the text is too long.*

width = max(300, txt\_surface.get\_width()+10)

input\_box.w = width

backGround()

*# Blit the text.*

gameWindow.blit(txt\_surface, (input\_box.x+5, input\_box.y+5))

*# Blit the input\_box rect.*

pygame.draw.rect(gameWindow, color, input\_box, 2)

pygame.display.flip()

clock.tick(30)

**def** gameMenu():

gameMenu = pygame.image.load(**"GameMenu.jpg"**)

gameMenu = pygame.transform.scale(gameMenu, (800, 600))

gameWindow.blit(gameMenu,(0,0))

font = pygame.font.Font(None, 32)

clock = pygame.time.Clock()

input\_box = pygame.Rect(460, 230, 400, 50)

color\_inactive = pygame.Color(**'lightskyblue3'**)

color\_active = pygame.Color(**'dodgerblue2'**)

color = color\_inactive

active = False

gameMenu.song = **''**

done = False

**while not** done:

**for** event **in** pygame.event.get():

**if** event.type == pygame.QUIT:

done = True

**if** event.type == pygame.MOUSEBUTTONDOWN:

*# If the user clicked on the input\_box rect.*

**if** input\_box.collidepoint(event.pos):

*# Toggle the active variable.*

active = **not** active

**else**:

active = False

*# Change the current color of the input box.*

color = color\_active **if** active **else** color\_inactive

**if** event.type == pygame.KEYDOWN:

**if** active:

**if** event.key == pygame.K\_RETURN:

done = True

**elif** event.key == pygame.K\_BACKSPACE:

gameMenu.song = gameMenu.song[:-1]

**else**:

gameMenu.song += event.unicode

*# Render the current text.*

txt\_surface = font.render(main.text, True, color)

*# Resize the box if the text is too long.*

width = max(300, txt\_surface.get\_width() + 10)

input\_box.w = width

backGround()

*# Blit the text.*

gameWindow.blit(txt\_surface, (input\_box.x + 5, input\_box.y + 5))

*# Blit the input\_box rect.*

pygame.draw.rect(gameWindow, color, input\_box, 2)

pygame.display.flip()

clock.tick(30)

*#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*# PROGRAM BEGINS HERE*

*#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

count=0

pygame.mixer.music.play(-1)

pygame.mixer.music.set\_volume(0.5)

**while** running:

**if** count==0:

main()

**print** (main.text)

count=1

gameMenu()

pygame.event.get()

keys=pygame.key.get\_pressed()

**if** keys[pygame.K\_ESCAPE]:

running= False

pygame.display.update()

pygame.quit()

DECEMBER 27

**import pygame  
  
print "Enter your username"  
pygame.init()  
WIDTH = 800  
HEIGHT = 600  
gameWindow = pygame.display.set\_mode((WIDTH, HEIGHT))  
  
GRIDSIZE = 10  
RED = (255, 0, 0)  
GREEN = (0, 255, 0)  
BLUE = (0, 0, 255)  
CYAN = (0, 255, 255)  
WHITE = (255, 255, 255)  
BLACK = (0, 0, 0)  
GREY = (128, 128, 128)  
  
#pygame.mixer.music.load("DoubleAgent.mp3")  
songs = ["PutOnYourSundayClothes.mp3",'Dreams.mp3', 'WeStayedUpAllNight.mp3', "Masquerade.mp3","Nevada.mp3","Predawn.mp3","Grafiore.mp3","Howling.mp3","PandorA.mp3","Phototropic.mp3","DoubleAgent.mp3"]  
  
pygame.mixer.music.load("DoubleAgent.mp3")  
  
def backGround():  
 picture = pygame.image.load('Background(1).jpg')  
 picture = pygame.transform.scale(picture, (800, 600))  
 gameWindow.blit(picture, (0, 0))  
  
  
def background1():  
 gameMenu = pygame.image.load("GameMenu.jpg")  
 gameMenu = pygame.transform.scale(gameMenu, (800, 600))  
 gameWindow.blit(gameMenu, (0, 0))  
  
  
running = True  
  
  
def main():  
 backGround()  
 # grid()  
 font = pygame.font.Font(None, 32)  
 clock = pygame.time.Clock()  
 input\_box = pygame.Rect(460, 230, 400, 54)  
 color\_inactive = pygame.Color('lightskyblue3')  
 color\_active = pygame.Color('dodgerblue2')  
 color = color\_inactive  
 active = False  
 main.text = ''  
 done = False  
 while not done:  
 for event in pygame.event.get():  
 if event.type == pygame.QUIT:  
 done = True  
 if event.type == pygame.MOUSEBUTTONDOWN:  
 # If the user clicked on the input\_box rect.  
 if input\_box.collidepoint(event.pos):  
 # Toggle the active variable.  
 active = not active  
 else:  
 active = False  
 # Change the current color of the input box.  
 color = color\_active if active else color\_inactive  
 if event.type == pygame.KEYDOWN:  
 if active:  
 if event.key == pygame.K\_RETURN:  
 print 'your username has been set to:', main.text  
 done = True  
 elif event.key == pygame.K\_BACKSPACE:  
 main.text = main.text[:-1]  
 else:  
 main.text += event.unicode  
  
 # Render the current text.  
 txt\_surface = font.render(main.text, True, color)  
 # Resize the box if the text is too long.  
 width = max(300, txt\_surface.get\_width() + 10)  
 input\_box.w = width  
 backGround()  
 # Blit the text.  
 gameWindow.blit(txt\_surface, (input\_box.x + 5, input\_box.y + 5))  
 # Blit the input\_box rect.  
 pygame.draw.rect(gameWindow, color, input\_box, 2)  
 pygame.display.flip()  
 clock.tick(30)  
  
  
def gameMenu():  
 pygame.mixer.music.stop  
 background1()  
 font = pygame.font.Font(None, 32)  
 clock = pygame.time.Clock()  
 input\_box = pygame.Rect(425, 230, 310, 60)  
 color\_inactive = pygame.Color('lightskyblue3')  
 color\_active = pygame.Color('dodgerblue2')  
 color = color\_inactive  
 active = False  
 gameMenu.song = ''  
 done = False  
 while not done:  
  
 for event in pygame.event.get():  
 if event.type == pygame.QUIT:  
 done = True  
 if event.type == pygame.MOUSEBUTTONDOWN:  
 # If the user clicked on the input\_box rect.  
 if input\_box.collidepoint(event.pos):  
 # Toggle the active variable.  
 active = not active  
 else:  
 active = False  
 # Change the current color of the input box.  
 color = color\_active if active else color\_inactive  
 if event.type == pygame.KEYDOWN:  
 if active:  
 if event.key == pygame.K\_RETURN:  
 done = True  
 elif event.key == pygame.K\_BACKSPACE:  
 gameMenu.song = gameMenu.song[:-1]  
 else:  
 gameMenu.song += event.unicode  
  
 # Render the current text.  
 txt\_surface = font.render(gameMenu.song, True, color)  
 # Resize the box if the text is too long.  
 width = max(270, txt\_surface.get\_width() + 10)  
 input\_box.w = width  
 # Blit the text.  
 gameWindow.blit(txt\_surface, (input\_box.x + 5, input\_box.y + 5))  
 # Blit the input\_box rect.  
 pygame.draw.rect(gameWindow, color, input\_box, 2)  
 pygame.display.flip()  
 clock.tick(30)  
  
coins = 0  
lives = 0  
r,g,b = 0  
def shop():  
 shopBackground = pygame.image.load()  
  
def game():  
 screen()  
 pygame.mixer.music.load(songs[int(gameMenu.song)])  
 pygame.mixer.music.play(1)  
 pygame.mixer.music.set\_volume(0.5)  
  
count = 0  
pygame.mixer.music.play(-1)  
pygame.mixer.music.set\_volume(0.5)  
  
while running:  
 if count == 0:  
 main()  
 print (main.text)  
 count = 1  
 gameMenu()  
 pygame.mixer.music.load(songs[int(gameMenu.song)])  
 pygame.mixer.music.play(-1)  
 pygame.mixer.music.set\_volume(0.5)  
'''  
 if gameMenu.song == "1":  
 screen1  
 elif gameMenu.song == "2":  
 screen2  
 elif gameMenu.song == "3":  
 screen3  
 elif gameMenu.song == "4":  
 screen4  
 elif gameMenu.song == "5":  
 screen5  
 elif gameMenu.song == "6":  
 screen6  
 elif gameMenu.song == "7":  
 screen7  
 elif gameMenu.song == "8":  
 screen8  
 elif gameMenu.song == "9":  
 screen9  
 elif gameMenu.song == "10":  
 screen10  
 game()  
'''  
 pygame.event.get()  
 keys = pygame.key.get\_pressed()  
 if keys[pygame.K\_ESCAPE]:  
 running = False  
 pygame.display.update()  
  
pygame.quit()**

Example game (make it into a function)

**import pygame, os, random**

**from pygame.locals import \***

**wix = 800**

**wiy = 600**

**def msg(screen, text, color=(55, 55, 55), size=36, pos=(-1, -1)):**

**if pos[0] == -1: pos = (screen.get\_rect().centerx, pos[1])**

**if pos[1] == -1: pos = (pos[0], screen.get\_rect().centery)**

**font = pygame.font.Font(None, size)**

**text = font.render(text, 1, color)**

**textpos = text.get\_rect()**

**textpos.centerx = pos[0]**

**textpos.centery = pos[1]**

**screen.blit(text, textpos)**

**class button():**

**x = 0**

**y = -wiy // 5**

**h = wix // 4 - 1**

**l = wiy // 5**

**enclick = True**

**def pos(self, n):**

**self.x = n \* wix // 4**

**def update(self, screen):**

**if self.enclick:**

**pygame.draw.rect(screen, (0, 0, 0), [self.x, self.y, self.h, self.l])**

**else:**

**pygame.draw.rect(screen, (180, 180, 180), [self.x, self.y, self.h, self.l])**

**def click(self, ps):**

**if ps[0] in range(self.x, self.x + self.h):**

**if ps[1] in range(self.y, self.y + self.l):**

**self.enclick = False**

**return 0**

**return 1**

**def game():**

**pygame.init()**

**pygame.mixer.get\_init()**

**clock = pygame.time.Clock()**

**screen = pygame.display.set\_mode((wix, wiy))**

**lost = 0**

**time = 0**

**delt = 60**

**sb = []**

**speed = 3**

**score = 0**

**playAgain = False**

**while lost == 0:**

**for i in range(10):**

**sb.append(button())**

**sb[-1].pos(random.randrange(4)) # (i)**

**if lost != 0: break**

**for j in range(wiy // (5 \* speed) + 1):**

**time += 1 / delt**

**clock.tick(delt)**

**screen.fill((224, 224, 255))**

**if lost != 0: break**

**for k in range(len(sb)):**

**try:**

**sb[k].y += speed**

**sb[k].update(screen)**

**if sb[k].y > wiy - sb[k].l and sb[k].enclick == True: lost = 1**

**except:**

**pass**

**for event in pygame.event.get():**

**if event.type == QUIT or \**

**(event.type == KEYDOWN and event.key == K\_ESCAPE):**

**pygame.quit()**

**elif event.type == MOUSEBUTTONDOWN:**

**lost = sb[score].click(pygame.mouse.get\_pos())**

**score += 1**

**msg(screen, "SCORE " + str(score), color=(0, 128, 255), pos=(-1, 30))**

**pygame.display.update()**

**speed += 1**

**pygame.mixer.music.stop()**

**msg(screen, "YOU LOSE ", color=(110, 128, 225), size=100, pos=(-1, -1))**

**msg(screen, ' press BACKSPACE to play again', color=(110, 128, 225), size=50, pos=(400, 20))**

**pygame.display.update()**

**pygame.time.wait(4000)**

**# can't get this part to work for them to play the game a second time**

**inPlay = True**

**while inPlay:**

**for event in pygame.event.get():**

**if event.type == pygame.KEYDOWN:**

**if event.key == pygame.K\_BACKSPACE:**

**inPlay = False**

**game()**

**pygame.display.update()**

**pygame.time.wait(4000)**

**pygame.quit()**

**quit()**

Jan 5

**import** pygame

**print "Enter your username"**

pygame.init()

WIDTH = 800

HEIGHT = 600

gameWindow = pygame.display.set\_mode((WIDTH, HEIGHT))

GRIDSIZE = 10

RED = (255, 0, 0)

GREEN = (0, 255, 0)

BLUE = (0, 0, 255)

CYAN = (0, 255, 255)

WHITE = (255, 255, 255)

BLACK = (0, 0, 0)

GREY = (128, 128, 128)

*#pygame.mixer.music.load("DoubleAgent.mp3")*

songs = [**"PutOnYourSundayClothes.mp3"**,**'Dreams.mp3'**, **'WeStayedUpAllNight.mp3'**, **"Masquerade.mp3"**,**"Nevada.mp3"**,**"Predawn.mp3"**,**"Grafiore.mp3"**,**"Howling.mp3"**,**"PandorA.mp3"**,**"Phototropic.mp3"**,**"DoubleAgent.mp3"**]

pygame.mixer.music.load(**"DoubleAgent.mp3"**)

**def** backGround():

picture = pygame.image.load(**'Background(1).jpg'**)

picture = pygame.transform.scale(picture, (800, 600))

gameWindow.blit(picture, (0, 0))

**def** background1():

gameMenu = pygame.image.load(**"GameMenu.jpg"**)

gameMenu = pygame.transform.scale(gameMenu, (800, 600))

gameWindow.blit(gameMenu, (0, 0))

running = True

**def** main():

backGround()

*# grid()*

font = pygame.font.Font(None, 32)

clock = pygame.time.Clock()

input\_box = pygame.Rect(460, 230, 400, 54)

color\_inactive = pygame.Color(**'lightskyblue3'**)

color\_active = pygame.Color(**'dodgerblue2'**)

color = color\_inactive

active = False

main.text = **''**

done = False

**while not** done:

**for** event **in** pygame.event.get():

**if** event.type == pygame.QUIT:

done = True

**if** event.type == pygame.MOUSEBUTTONDOWN:

*# If the user clicked on the input\_box rect.*

**if** input\_box.collidepoint(event.pos):

*# Toggle the active variable.*

active = **not** active

**else**:

active = False

*# Change the current color of the input box.*

color = color\_active **if** active **else** color\_inactive

**if** event.type == pygame.KEYDOWN:

**if** active:

**if** event.key == pygame.K\_RETURN:

**print 'your username has been set to:'**, main.text

done = True

**elif** event.key == pygame.K\_BACKSPACE:

main.text = main.text[:-1]

**else**:

main.text += event.unicode

*# Render the current text.*

txt\_surface = font.render(main.text, True, color)

*# Resize the box if the text is too long.*

width = max(300, txt\_surface.get\_width() + 10)

input\_box.w = width

backGround()

*# Blit the text.*

gameWindow.blit(txt\_surface, (input\_box.x + 5, input\_box.y + 5))

*# Blit the input\_box rect.*

pygame.draw.rect(gameWindow, color, input\_box, 2)

pygame.display.flip()

clock.tick(30)

**def** gameMenu():

pygame.mixer.music.stop

background1()

font = pygame.font.Font(None, 32)

clock = pygame.time.Clock()

input\_box = pygame.Rect(425, 230, 310, 60)

color\_inactive = pygame.Color(**'lightskyblue3'**)

color\_active = pygame.Color(**'dodgerblue2'**)

color = color\_inactive

active = False

gameMenu.song = **''**

done = False

**while not** done:

**for** event **in** pygame.event.get():

**if** event.type == pygame.QUIT:

done = True

**if** event.type == pygame.MOUSEBUTTONDOWN:

*# If the user clicked on the input\_box rect.*

**if** input\_box.collidepoint(event.pos):

*# Toggle the active variable.*

active = **not** active

**else**:

active = False

*# Change the current color of the input box.*

color = color\_active **if** active **else** color\_inactive

**if** event.type == pygame.KEYDOWN:

**if** active:

**if** event.key == pygame.K\_RETURN:

done = True

**elif** event.key == pygame.K\_BACKSPACE:

gameMenu.song = gameMenu.song[:-1]

**else**:

gameMenu.song += event.unicode

*# Render the current text.*

txt\_surface = font.render(gameMenu.song, True, color)

*# Resize the box if the text is too long.*

width = max(270, txt\_surface.get\_width() + 10)

input\_box.w = width

*# Blit the text.*

gameWindow.blit(txt\_surface, (input\_box.x + 5, input\_box.y + 5))

*# Blit the input\_box rect.*

pygame.draw.rect(gameWindow, color, input\_box, 2)

pygame.display.flip()

clock.tick(30)

coins = 0

lives = 0

r,g,b = 0,0,0

**def** MusicScreen():

musicbackground1 = pygame.image.load(**"MusicBackGround(1).png"**)

musicbackground2 = pygame.image.load(**"MusicBackGround(2).png"**)

musicbackground3 = pygame.image.load(**"MusicBackGround(3).png"**)

musicbackground4 = pygame.image.load(**"MusicBackGround(4).png"**)

musicbackground5 = pygame.image.load(**"MusicBackGround(5).png"**)

musicbackground6 = pygame.image.load(**"MusicBackGround(6).png"**)

musicbackground7 = pygame.image.load(**"MusicBackGround(7).png"**)

musicbackground8 = pygame.image.load(**"MusicBackGround(8).png"**)

musicbackground9 = pygame.image.load(**"MusicBackGround(9).png"**)

musicbackground10 = pygame.image.load(**"MusicBackGround(10).png"**)

musicbackground1 = pygame.transform.scale(musicbackground1,(400,400))

musicbackground2 = pygame.transform.scale(musicbackground2, (400, 400))

musicbackground3 = pygame.transform.scale(musicbackground3, (400, 400))

musicbackground4 = pygame.transform.scale(musicbackground4, (400, 400))

musicbackground5 = pygame.transform.scale(musicbackground5, (400, 400))

musicbackground6 = pygame.transform.scale(musicbackground6, (400, 400))

musicbackground7 = pygame.transform.scale(musicbackground7, (400, 400))

musicbackground8 = pygame.transform.scale(musicbackground8, (400, 400))

musicbackground9 = pygame.transform.scale(musicbackground9, (400, 400))

musicbackground10 = pygame.transform.scale(musicbackground10, (400, 400))

**if** gameMenu.song == **"1"**:

gameWindow.blit(musicbackground1,(0, 0))

**elif** gameMenu.song == **"2"**:

gameWindow.blit(musicbackground2, (0, 0))

**elif** gameMenu.song == **"3"**:

gameWindow.blit(musicbackground3, (0, 0))

**elif** gameMenu.song == **"4"**:

gameWindow.blit(musicbackground4, (0, 0))

**elif** gameMenu.song == **"5"**:

gameWindow.blit(musicbackground5, (0, 0))

**elif** gameMenu.song == **"6"**:

gameWindow.blit(musicbackground6, (0, 0))

**elif** gameMenu.song == **"7"**:

gameWindow.blit(musicbackground7, (0, 0))

**elif** gameMenu.song == **"8"**:

gameWindow.blit(musicbackground8, (0, 0))

**elif** gameMenu.song == **"9"**:

gameWindow.blit(musicbackground9, (0, 0))

**elif** gameMenu.song == **"10"**:

gameWindow.blit(musicbackground10, (0, 0))

**def** msg(screen, text, color=(55, 55, 55), size=36, pos=(-1, -1)):

**if** pos[0] == -1: pos = (screen.get\_rect().centerx, pos[1])

**if** pos[1] == -1: pos = (pos[0], screen.get\_rect().centery)

font = pygame.font.Font(None, size)

text = font.render(text, 1, color)

textpos = text.get\_rect()

textpos.centerx = pos[0]

textpos.centery = pos[1]

screen.blit(text, textpos)

**class** button():

x = 0

y = -HEIGHT // 5

h = WIDTH // 4 - 1

l = HEIGHT // 5

enclick = True

**def** pos(self, n):

self.x = n \* wix // 4

**def** update(self, screen):

**if** self.enclick:

pygame.draw.rect(screen, (0, 0, 0), [self.x, self.y, self.h, self.l])

**else**:

pygame.draw.rect(screen, (180, 180, 180), [self.x, self.y, self.h, self.l])

**def** click(self, ps):

**if** ps[0] **in** range(self.x, self.x + self.h):

**if** ps[1] **in** range(self.y, self.y + self.l):

self.enclick = False

**return** 0

**return** 1

**def** game():

MusicScreen()

pygame.mixer.music.load(songs[int(gameMenu.song)])

pygame.mixer.music.play(1)

pygame.mixer.music.set\_volume(0.5)

clock = pygame.time.Clock()

screen = pygame.display.set\_mode((WIDTH, HEIGHT))

lost = 0

time = 0

delt = 60

sb = []

speed = 3

score = 0

playAgain = False

**while** lost == 0:

**for** i **in** range(10):

sb.append(button())

sb[-1].pos(random.randrange(4)) *# (i)*

**if** lost != 0: **break**

**for** j **in** range(wiy // (5 \* speed) + 1):

time += 1 / delt

clock.tick(delt)

screen.fill((224, 224, 255))

**if** lost != 0: **break**

**for** k **in** range(len(sb)):

**try**:

sb[k].y += speed

sb[k].update(screen)

**if** sb[k].y > wiy - sb[k].l **and** sb[k].enclick == True: lost = 1

**except**:

**pass**

**for** event **in** pygame.event.get():

**if** event.type == QUIT **or** \

(event.type == KEYDOWN **and** event.key == K\_ESCAPE):

pygame.quit()

**elif** event.type == MOUSEBUTTONDOWN:

lost = sb[score].click(pygame.mouse.get\_pos())

score += 1

msg(screen, **"SCORE "** + str(score), color=(0, 128, 255), pos=(-1, 30))

pygame.display.update()

speed += 1

pygame.mixer.music.stop()

msg(screen, **"YOU LOSE "**, color=(110, 128, 225), size=100, pos=(-1, -1))

msg(screen, **' press BACKSPACE to play again'**, color=(110, 128, 225), size=50, pos=(400, 20))

pygame.display.update()

pygame.time.wait(4000)

count = 0

pygame.mixer.music.play(-1)

pygame.mixer.music.set\_volume(0.5)

**while** running:

**if** count == 0:

main()

**print** (main.text)

count = 1

gameMenu()

pygame.mixer.music.load(songs[int(gameMenu.song)])

pygame.mixer.music.play(-1)

pygame.mixer.music.set\_volume(0.5)

game()

pygame.event.get()

keys = pygame.key.get\_pressed()

**if** keys[pygame.K\_ESCAPE]:

running = False

pygame.display.update()

pygame.quit()

Jan 8th

import pygame  
import random  
from pygame.locals import \*  
  
print "Enter your username"  
pygame.init()  
WIDTH = 800  
HEIGHT = 600  
wix = 800  
wiy = 600  
gameWindow = pygame.display.set\_mode((WIDTH, HEIGHT))  
  
GRIDSIZE = 10  
RED = (255, 0, 0)  
GREEN = (0, 255, 0)  
BLUE = (0, 0, 255)  
CYAN = (0, 255, 255)  
WHITE = (255, 255, 255)  
BLACK = (0, 0, 0)  
GREY = (128, 128, 128)  
  
  
songs = ["PutOnYourSundayClothes.mp3",'Dreams.mp3', 'WeStayedUpAllNight.mp3', "Masquerade.mp3","Nevada.mp3","Predawn.mp3","Grafiore.mp3","Howling.mp3","PandorA.mp3","Phototropic.mp3","DoubleAgent.mp3"]  
  
pygame.mixer.music.load("DoubleAgent.mp3")  
  
def backGround():  
 picture = pygame.image.load('Background(1).jpg').convert\_alpha()  
 picture = pygame.transform.scale(picture, (800, 600))  
 gameWindow.blit(picture, (0, 0))  
  
def background1():  
 gameMenu = pygame.image.load("GameMenu.jpg").convert\_alpha()  
 gameMenu = pygame.transform.scale(gameMenu, (800, 600))  
 gameWindow.blit(gameMenu, (0, 0))  
  
running = True  
  
  
def main():  
 backGround()  
 # grid()  
 font = pygame.font.Font(None, 32)  
 clock = pygame.time.Clock()  
 input\_box = pygame.Rect(460, 230, 400, 54)  
 color\_inactive = pygame.Color('lightskyblue3')  
 color\_active = pygame.Color('dodgerblue2')  
 color = color\_inactive  
 active = False  
 main.text = ''  
 done = False  
 while not done:  
 for event in pygame.event.get():  
 if event.type == pygame.QUIT:  
 done = True  
 if event.type == pygame.MOUSEBUTTONDOWN:  
 # If the user clicked on the input\_box rect.  
 if input\_box.collidepoint(event.pos):  
 # Toggle the active variable.  
 active = not active  
 else:  
 active = False  
 # Change the current color of the input box.  
 color = color\_active if active else color\_inactive  
 if event.type == pygame.KEYDOWN:  
 if active:  
 if event.key == pygame.K\_RETURN:  
 print 'your username has been set to:', main.text  
 done = True  
 elif event.key == pygame.K\_BACKSPACE:  
 main.text = main.text[:-1]  
 else:  
 main.text += event.unicode  
  
 # Render the current text.  
 txt\_surface = font.render(main.text, True, color)  
 # Resize the box if the text is too long.  
 width = max(300, txt\_surface.get\_width() + 10)  
 input\_box.w = width  
 backGround()  
 # Blit the text.  
 gameWindow.blit(txt\_surface, (input\_box.x + 5, input\_box.y + 5))  
 # Blit the input\_box rect.  
 pygame.draw.rect(gameWindow, color, input\_box, 2)  
 pygame.display.flip()  
 clock.tick(30)  
  
  
def redrawGameWindow():  
 gameWindow.blit(ship, (shipX,shipY))  
 pygame.display.update()  
  
def rotate(image, angle):  
 ORIGINALrect = image.get\_rect()  
 rotatedImage = pygame.transform.rotate(image,angle)  
 rotatedRect = ORIGINALrect.copy()  
 rotatedRect.center = rotatedImage.get\_rect().center  
 rotatedImage = rotatedImage.subsurface(rotatedRect).copy()  
 return rotatedImage  
  
#--------------------------------------------------------------#  
def mainMenu():  
 gameWindow.blit(backGround,(0,0))  
 redrawGameWindow()  
 clock.tick(FPS)  
 pygame.event.get()  
 keys = pygame.key.get\_pressed()  
 mouseX, mouseY = pygame.mouse.get\_pos()  
 if keys[pygame.K\_ESCAPE]:  
 inPlay = False  
 for event in pygame.event.get():  
 if event.type == pygame.MOUSEBUTTONDOWN and mouseX <=710 and mouseX>=390 and mouseY >=170 and mouseY <= 240:  
 play()  
 break  
 elif event.type == pygame.MOUSEBUTTONDOWN and mouseX <=730 and mouseX>=400 and mouseY >=260 and mouseY <= 320:  
 tutorial()  
 break  
 elif event.type == pygame.MOUSEBUTTONDOWN and mouseX <=710 and mouseX>=390 and mouseY >=340 and mouseY <= 410:  
 break  
 shipAngle = shipAngle + rotationStep  
 ship = rotate(ORIGINALship,shipAngle)  
  
#---------------------------------------------#  
ORIGINALship = pygame.image.load("Pendulum(2).png").convert\_alpha()  
ORIGINALship = pygame.transform.scale(ORIGINALship,(400,400))  
backGround = pygame.image.load("BackGround(2).jpg").convert\_alpha()  
backGround = pygame.transform.scale(backGround,(800,600))  
pygame.mixer.music.load("DoubleAgent.mp3")  
ship = ORIGINALship.copy() # keep the original image intact, so it does not get distorted  
shipX = 10  
shipY = 100  
shipAngle = 0  
rotationStep = 10  
  
#---------------------------------------#  
print "Hit ESC to end the program."  
clock = pygame.time.Clock()  
FPS = 30  
  
def gameMenu():  
 pygame.mixer.music.stop  
 background1()  
 font = pygame.font.Font(None, 32)  
 clock = pygame.time.Clock()  
 input\_box = pygame.Rect(425, 230, 310, 60)  
 color\_inactive = pygame.Color('lightskyblue3')  
 color\_active = pygame.Color('dodgerblue2')  
 color = color\_inactive  
 active = False  
 gameMenu.song = ''  
 done = False  
 while not done:  
  
 for event in pygame.event.get():  
 if event.type == pygame.QUIT:  
 done = True  
 if event.type == pygame.MOUSEBUTTONDOWN:  
 # If the user clicked on the input\_box rect.  
 if input\_box.collidepoint(event.pos):  
 # Toggle the active variable.  
 active = not active  
 else:  
 active = False  
 # Change the current color of the input box.  
 color = color\_active if active else color\_inactive  
 if event.type == pygame.KEYDOWN:  
 if active:  
 if event.key == pygame.K\_RETURN:  
 done = True  
 elif event.key == pygame.K\_BACKSPACE:  
 gameMenu.song = gameMenu.song[:-1]  
 else:  
 gameMenu.song += event.unicode  
  
 # Render the current text.  
 txt\_surface = font.render(gameMenu.song, True, color)  
 # Resize the box if the text is too long.  
 width = max(270, txt\_surface.get\_width() + 10)  
 input\_box.w = width  
 # Blit the text.  
 gameWindow.blit(txt\_surface, (input\_box.x + 5, input\_box.y + 5))  
 # Blit the input\_box rect.  
 pygame.draw.rect(gameWindow, color, input\_box, 2)  
 pygame.display.flip()  
 clock.tick(30)  
  
coins = 0  
lives = 0  
r,g,b = 0,0,0  
  
#def play(mouseX,mouseY):  
  
  
def MusicScreen():  
 musicbackground1 = pygame.image.load("MusicBackGround(1).png").convert\_alpha()  
 musicbackground2 = pygame.image.load("MusicBackGround(2).png").convert\_alpha()  
 musicbackground3 = pygame.image.load("MusicBackGround(3).png").convert\_alpha()  
 musicbackground4 = pygame.image.load("MusicBackGround(4).png").convert\_alpha()  
 musicbackground5 = pygame.image.load("MusicBackGround(5).png").convert\_alpha()  
 musicbackground6 = pygame.image.load("MusicBackGround(6).png").convert\_alpha()  
 musicbackground7 = pygame.image.load("MusicBackGround(7).png").convert\_alpha()  
 musicbackground8 = pygame.image.load("MusicBackGround(8).png").convert\_alpha()  
 musicbackground9 = pygame.image.load("MusicBackGround(9).png").convert\_alpha()  
 musicbackground10 = pygame.image.load("MusicBackGround(10).png").convert\_alpha()  
 musicbackground1 = pygame.transform.scale(musicbackground1,(400,400))  
 musicbackground2 = pygame.transform.scale(musicbackground2, (400, 400))  
 musicbackground3 = pygame.transform.scale(musicbackground3, (400, 400))  
 musicbackground4 = pygame.transform.scale(musicbackground4, (400, 400))  
 musicbackground5 = pygame.transform.scale(musicbackground5, (400, 400))  
 musicbackground6 = pygame.transform.scale(musicbackground6, (400, 400))  
 musicbackground7 = pygame.transform.scale(musicbackground7, (400, 400))  
 musicbackground8 = pygame.transform.scale(musicbackground8, (400, 400))  
 musicbackground9 = pygame.transform.scale(musicbackground9, (400, 400))  
 musicbackground10 = pygame.transform.scale(musicbackground10, (400, 400))  
 if gameMenu.song == "1":  
 gameWindow.blit(musicbackground1,(0, 0))  
 '''if play():  
 return True  
 else:  
 return False'''  
 elif gameMenu.song == "2":  
 gameWindow.blit(musicbackground2, (0, 0))  
 elif gameMenu.song == "3":  
 gameWindow.blit(musicbackground3, (0, 0))  
 elif gameMenu.song == "4":  
 gameWindow.blit(musicbackground4, (0, 0))  
 elif gameMenu.song == "5":  
 gameWindow.blit(musicbackground5, (0, 0))  
 elif gameMenu.song == "6":  
 gameWindow.blit(musicbackground6, (0, 0))  
 elif gameMenu.song == "7":  
 gameWindow.blit(musicbackground7, (0, 0))  
 elif gameMenu.song == "8":  
 gameWindow.blit(musicbackground8, (0, 0))  
 elif gameMenu.song == "9":  
 gameWindow.blit(musicbackground9, (0, 0))  
 elif gameMenu.song == "10":  
 gameWindow.blit(musicbackground10, (0, 0))  
  
def msg(screen, text, color=(55, 55, 55), size=36, pos=(-1, -1)):  
 if pos[0] == -1: pos = (screen.get\_rect().centerx, pos[1])  
 if pos[1] == -1: pos = (pos[0], screen.get\_rect().centery)  
 font = pygame.font.Font(None, size)  
 text = font.render(text, 1, color)  
 textpos = text.get\_rect()  
 textpos.centerx = pos[0]  
 textpos.centery = pos[1]  
 screen.blit(text, textpos)  
  
class button():  
 x = 0  
 y = -HEIGHT // 5  
 h = WIDTH // 4 - 1  
 l = HEIGHT // 5  
 enclick = True  
  
 def pos(self, n):  
 self.x = n \* wix // 4  
  
 def update(self, screen):  
 if self.enclick:  
 pygame.draw.rect(screen, (0, 0, 0), [self.x, self.y, self.h, self.l])  
 else:  
 pygame.draw.rect(screen, (180, 180, 180), [self.x, self.y, self.h, self.l])  
  
 def click(self, ps):  
 if ps[0] in range(self.x, self.x + self.h):  
 if ps[1] in range(self.y, self.y + self.l):  
 self.enclick = False  
 return 0  
 return 1  
  
def game():  
 pygame.mixer.music.load(songs[int(gameMenu.song)])  
 pygame.mixer.music.play(1)  
 pygame.mixer.music.set\_volume(0.5)  
 MusicScreen()  
 clock = pygame.time.Clock()  
 screen = pygame.display.set\_mode((WIDTH, HEIGHT))  
 lost = 0  
 time = 0  
 delt = 60  
 sb = []  
 speed = 3  
 score = 0  
 playAgain = False  
 while lost == 0:  
 for i in range(10):  
 sb.append(button())  
 sb[-1].pos(random.randrange(4)) # (i)  
 if lost != 0: break  
 for j in range(wiy // (5 \* speed) + 1):  
 time += 1 / delt  
 clock.tick(delt)  
 screen.fill((224, 224, 255))  
 if lost != 0: break  
 for k in range(len(sb)):  
 try:  
 sb[k].y += speed  
 sb[k].update(screen)  
 if sb[k].y > wiy - sb[k].l and sb[k].enclick == True: lost = 1  
 except:  
 pass  
 for event in pygame.event.get():  
 if event.type == QUIT or \  
 (event.type == KEYDOWN and event.key == K\_ESCAPE):  
 pygame.quit()  
 elif event.type == MOUSEBUTTONDOWN:  
 lost = sb[score].click(pygame.mouse.get\_pos())  
 score += 1  
 msg(screen, "SCORE " + str(score), color=(0, 128, 255), pos=(-1, 30))  
 pygame.display.update()  
 speed += 1  
 pygame.mixer.music.stop()  
  
 msg(screen, "YOU LOSE ", color=(110, 128, 225), size=100, pos=(-1, -1))  
 msg(screen, ' press BACKSPACE to play again', color=(110, 128, 225), size=50, pos=(400, 20))  
  
 pygame.display.update()  
 pygame.time.wait(4000)  
  
def tutorial():   
 pygame.mixer.music.load("Double Agent.mp3")  
 pygame.mixer.music.play(1)  
 pygame.mixer.music.set\_volume(0.5)  
 clock = pygame.time.Clock()  
 screen = pygame.display.set\_mode((WIDTH, HEIGHT))  
 msg(screen, "Press q,w,e,r on the keyboard when the blocks land on the bottom. If the block is outside, then 10 health will be taken from the hp bar. The score will add up each time.")  
 lost = 0  
 time = 0  
 delt = 60  
 sb = []  
 speed = 1  
 score = 0  
 playAgain = False  
 while lost == 0:  
 for i in range(10):  
 sb.append(button())  
 sb[-1].pos(random.randrange(4)) # (i)  
 if lost != 0: break  
 for j in range(wiy // (5 \* speed) + 1):  
 time += 1 / delt  
 clock.tick(delt)  
 screen.fill((224, 224, 255))  
 if lost != 0: break  
 for k in range(len(sb)):  
 try:  
 sb[k].y += speed  
 sb[k].update(screen)  
 if sb[k].y > wiy - sb[k].l and sb[k].enclick == True: lost = 1  
 except:  
 pass  
 for event in pygame.event.get():  
 if event.type == QUIT or \  
 (event.type == KEYDOWN and event.key == K\_ESCAPE):  
 pygame.quit()  
 elif event.type == MOUSEBUTTONDOWN:  
 lost = sb[score].click(pygame.mouse.get\_pos())  
 score += 1  
 msg(screen, "SCORE " + str(score), color=(0, 128, 255), pos=(-1, 30))  
 pygame.display.update()  
 speed += 1  
 pygame.mixer.music.stop()  
  
 msg(screen, "YOU LOSE ", color=(110, 128, 225), size=100, pos=(-1, -1))  
 msg(screen, ' press BACKSPACE to play again', color=(110, 128, 225), size=50, pos=(400, 20))  
 pygame.display.update()  
 pygame.time.wait(4000)  
  
  
  
count = 0  
pygame.mixer.music.play(-1)  
pygame.mixer.music.set\_volume(0.5)  
  
while running:  
 if count == 0:  
 count = 1  
 main()  
 print (main.text)  
 mainMenu()  
 gameMenu()  
 pygame.mixer.music.load(songs[int(gameMenu.song)])  
 pygame.mixer.music.play(-1)  
 pygame.mixer.music.set\_volume(0.5)  
 game()  
 pygame.event.get()  
 keys = pygame.key.get\_pressed()  
 if keys[pygame.K\_ESCAPE]:  
 running = False  
 pygame.display.update()  
  
pygame.quit()

Jan 11

**import pygame**

**import random**

**from pygame.locals import \***

**pygame.init()**

**print "Enter your username"**

**pygame.init()**

**WIDTH = 800**

**HEIGHT = 600**

**wix = 800**

**wiy = 600**

**gameWindow = pygame.display.set\_mode((WIDTH, HEIGHT))**

**GRIDSIZE = 10**

**RED = (255, 0, 0)**

**GREEN = (0, 255, 0)**

**BLUE = (0, 0, 255)**

**CYAN = (0, 255, 255)**

**WHITE = (255, 255, 255)**

**BLACK = (0, 0, 0)**

**GREY = (128, 128, 128)**

**songs = ["PutOnYourSundayClothes.mp3",'Dreams.mp3', 'WeStayedUpAllNight.mp3', "Masquerade.mp3","Nevada.mp3","Predawn.mp3","Grafiore.mp3","Howling.mp3","PandorA.mp3","Phototropic.mp3","DoubleAgent.mp3"]**

**pygame.mixer.music.load("DoubleAgent.mp3")**

**def backGround():**

**picture = pygame.image.load("Background(1).jpg").convert\_alpha()**

**picture = pygame.transform.scale(picture, (800, 600))**

**gameWindow.blit(picture, (0, 0))**

**def background1():**

**gameMenu = pygame.image.load("GameMenu.jpg").convert\_alpha()**

**gameMenu = pygame.transform.scale(gameMenu, (800, 600))**

**gameWindow.blit(gameMenu, (0, 0))**

**running = True**

**def grid():**

**for x in range(0, WIDTH, 10):**

**pygame.draw.line(gameWindow, BLACK, (x, 0), (x, HEIGHT), 1)**

**for y in range(0, HEIGHT, 10):**

**pygame.draw.line(gameWindow, BLACK, (0, y), (WIDTH, y), 1)**

**for x in range(0, WIDTH, 10 \* 10):**

**pygame.draw.line(gameWindow, BLACK, (x, 0), (x, HEIGHT), 2)**

**for y in range(0, HEIGHT, 10 \* 10):**

**pygame.draw.line(gameWindow, BLACK, (0, y), (WIDTH, y), 2)**

**def main():**

**backGround()**

**font = pygame.font.Font(None, 32)**

**clock = pygame.time.Clock()**

**input\_box = pygame.Rect(460, 230, 400, 54)**

**color\_inactive = pygame.Color('lightskyblue3')**

**color\_active = pygame.Color('dodgerblue2')**

**color = color\_inactive**

**active = False**

**main.text = ''**

**done = False**

**while not done:**

**for event in pygame.event.get():**

**if event.type == pygame.QUIT:**

**done = True**

**if event.type == pygame.MOUSEBUTTONDOWN:**

***# If the user clicked on the input\_box rect.***

**if input\_box.collidepoint(event.pos):**

***# Toggle the active variable.***

**active = not active**

**else:**

**active = False**

***# Change the current color of the input box.***

**color = color\_active if active else color\_inactive**

**if event.type == pygame.KEYDOWN:**

**if active:**

**if event.key == pygame.K\_RETURN:**

**print 'Your username has been set to:', main.text**

**done = True**

**elif event.key == pygame.K\_BACKSPACE:**

**main.text = main.text[:-1]**

**else:**

**main.text += event.unicode**

***# Render the current text.***

**txt\_surface = font.render(main.text, True, color)**

***# Resize the box if the text is too long.***

**width = max(300, txt\_surface.get\_width() + 10)**

**input\_box.w = width**

**backGround()**

***# Blit the text.***

**gameWindow.blit(txt\_surface, (input\_box.x + 5, input\_box.y + 5))**

***# Blit the input\_box rect.***

**pygame.draw.rect(gameWindow, color, input\_box, 2)**

**pygame.display.flip()**

**clock.tick(30)**

**def redrawGameWindow():**

**gameWindow.blit(ship, (shipX,shipY))**

**'''background = pygame.image.load("BackGround(2).jpg")**

**background = pygame.transform.scale(background, (800, 600))**

**gameWindow.blit(background, (0, 0))'''**

**pygame.display.update()**

**def rotate(image, angle):**

**ORIGINALrect = image.get\_rect()**

**rotatedImage = pygame.transform.rotate(image,angle)**

**rotatedRect = ORIGINALrect.copy()**

**rotatedRect.center = rotatedImage.get\_rect().center**

**rotatedImage = rotatedImage.subsurface(rotatedRect).copy()**

**return rotatedImage**

***#--------------------------------------------------------------#***

**def mainMenu():**

**background2 = pygame.image.load("BackGround(2).jpg")**

**background2 = pygame.transform.scale(background2, (800, 600))**

**gameWindow.blit(background2, (0, 0))**

**pygame.display.update()**

**clock.tick(FPS)**

**pygame.event.get()**

**keys = pygame.key.get\_pressed()**

**inPlay = True**

**global ship, shipY, shipX, shipAngle**

**ORIGINALship = pygame.image.load("Pendulum(2).png")**

**ORIGINALship = pygame.transform.scale(ORIGINALship, (400, 400))**

***# backGround = pygame.image.load("BackGround(2).jpg")***

***# backGround = pygame.transform.scale(backGround, (800, 600))***

**ship = ORIGINALship.copy()**

**shipX = -10**

**shipY = 100**

**shipAngle = 0**

**rotationStep = 4**

**while inPlay:**

**redrawGameWindow()**

***# gameWindow.blit(backGround, (0, 0))***

**clock.tick(FPS)**

**pygame.event.get()**

**shipAngle = shipAngle + rotationStep**

**ship = rotate(ORIGINALship, shipAngle)**

**pygame.display.update()**

**if keys[pygame.K\_ESCAPE]:**

**inPlay = False**

**for event in pygame.event.get():**

**if event.type == pygame.MOUSEBUTTONDOWN and mouseX <=710 and mouseX>=390 and mouseY >=170 and mouseY <= 240:**

**inPlay = False**

**elif event.type == pygame.MOUSEBUTTONDOWN and mouseX <=730 and mouseX>=400 and mouseY >=260 and mouseY <= 320:**

**transition(800,600)**

**tutorial()**

**inPlay = False**

**elif event.type == pygame.MOUSEBUTTONDOWN and mouseX <=710 and mouseX>=390 and mouseY >=340 and mouseY <= 410:**

**pygame.quit()**

**mouseX, mouseY = pygame.mouse.get\_pos()**

***#---------------------------------------------#***

***#---------------------------------------#***

**print "Hit ESC to end the program."**

**clock = pygame.time.Clock()**

**FPS = 30**

**def gameMenu():**

**pygame.mixer.music.stop**

**background1()**

**font = pygame.font.Font(None, 32)**

**clock = pygame.time.Clock()**

**input\_box = pygame.Rect(425, 230, 310, 60)**

**color\_inactive = pygame.Color('lightskyblue3')**

**color\_active = pygame.Color('dodgerblue2')**

**color = color\_inactive**

**active = False**

**gameMenu.song = ''**

**done = False**

**pygame.init()**

**while not done:**

**for event in pygame.event.get():**

**if event.type == pygame.QUIT:**

**done = True**

**if event.type == pygame.MOUSEBUTTONDOWN and mouseX <= 790 and mouseX >= 610 and mouseY >= 520 and mouseY <= 600:**

**done = True**

**if event.type == pygame.MOUSEBUTTONDOWN:**

***# If the user clicked on the input\_box rect.***

**if input\_box.collidepoint(event.pos):**

***# Toggle the active variable.***

**active = not active**

**else:**

**active = False**

***# Change the current color of the input box.***

**color = color\_active if active else color\_inactive**

**if event.type == pygame.KEYDOWN:**

**if active:**

**if event.key == pygame.K\_RETURN:**

**done = True**

**elif event.key == pygame.K\_BACKSPACE:**

**gameMenu.song = gameMenu.song[:-1]**

**else:**

**gameMenu.song += event.unicode**

**mouseX, mouseY = pygame.mouse.get\_pos()**

***# Render the current text.***

**txt\_surface = font.render(gameMenu.song, True, color)**

***# Resize the box if the text is too long.***

**width = max(270, txt\_surface.get\_width() + 10)**

**input\_box.w = width**

***# Blit the text.***

**gameWindow.blit(txt\_surface, (input\_box.x + 5, input\_box.y + 5))**

***# Blit the input\_box rect.***

**pygame.draw.rect(gameWindow, color, input\_box, 2)**

**pygame.display.flip()**

**clock.tick(30)**

***#grid()***

**def transition(width,height):**

**fade = pygame.Surface((width,height))**

**fade.fill((0,0,0))**

**for alpha in range(50):**

**fade.set\_alpha(alpha)**

**gameWindow.blit(fade,(0,0))**

**pygame.display.update()**

**pygame.time.delay(10)**

***#def play(mouseX,mouseY):***

**def MusicScreen():**

**musicbackground1 = pygame.image.load("MusicBackGround(1).png").convert\_alpha()**

**musicbackground2 = pygame.image.load("MusicBackGround(2).png").convert\_alpha()**

**musicbackground3 = pygame.image.load("MusicBackGround(3).png").convert\_alpha()**

**musicbackground4 = pygame.image.load("MusicBackGround(4).png").convert\_alpha()**

**musicbackground5 = pygame.image.load("MusicBackGround(5).png").convert\_alpha()**

**musicbackground6 = pygame.image.load("MusicBackGround(6).png").convert\_alpha()**

**musicbackground7 = pygame.image.load("MusicBackGround(7).png").convert\_alpha()**

**musicbackground8 = pygame.image.load("MusicBackGround(8).png").convert\_alpha()**

**musicbackground9 = pygame.image.load("MusicBackGround(9).png").convert\_alpha()**

**musicbackground10 = pygame.image.load("MusicBackGround(10).png").convert\_alpha()**

**musicbackground1 = pygame.transform.scale(musicbackground1,(400,400))**

**musicbackground2 = pygame.transform.scale(musicbackground2, (400, 400))**

**musicbackground3 = pygame.transform.scale(musicbackground3, (400, 400))**

**musicbackground4 = pygame.transform.scale(musicbackground4, (400, 400))**

**musicbackground5 = pygame.transform.scale(musicbackground5, (400, 400))**

**musicbackground6 = pygame.transform.scale(musicbackground6, (400, 400))**

**musicbackground7 = pygame.transform.scale(musicbackground7, (400, 400))**

**musicbackground8 = pygame.transform.scale(musicbackground8, (400, 400))**

**musicbackground9 = pygame.transform.scale(musicbackground9, (400, 400))**

**musicbackground10 = pygame.transform.scale(musicbackground10, (400, 400))**

**if gameMenu.song == "1":**

**gameWindow.blit(musicbackground1,(0, 0))**

**'''if play():**

**return True**

**else:**

**return False'''**

**elif gameMenu.song == "2":**

**gameWindow.blit(musicbackground2, (0, 0))**

**elif gameMenu.song == "3":**

**gameWindow.blit(musicbackground3, (0, 0))**

**elif gameMenu.song == "4":**

**gameWindow.blit(musicbackground4, (0, 0))**

**elif gameMenu.song == "5":**

**gameWindow.blit(musicbackground5, (0, 0))**

**elif gameMenu.song == "6":**

**gameWindow.blit(musicbackground6, (0, 0))**

**elif gameMenu.song == "7":**

**gameWindow.blit(musicbackground7, (0, 0))**

**elif gameMenu.song == "8":**

**gameWindow.blit(musicbackground8, (0, 0))**

**elif gameMenu.song == "9":**

**gameWindow.blit(musicbackground9, (0, 0))**

**elif gameMenu.song == "10":**

**gameWindow.blit(musicbackground10, (0, 0))**

**def msg(screen, text, color=(55, 55, 55), size=36, pos=(-1, -1)):**

**if pos[0] == -1: pos = (screen.get\_rect().centerx, pos[1])**

**if pos[1] == -1: pos = (pos[0], screen.get\_rect().centery)**

**font = pygame.font.Font(None, size)**

**text = font.render(text, 1, color)**

**textpos = text.get\_rect()**

**textpos.centerx = pos[0]**

**textpos.centery = pos[1]**

**screen.blit(text, textpos)**

**class button():**

**x = 0**

**y = -HEIGHT // 5**

**h = WIDTH // 4 - 1**

**l = HEIGHT // 5**

**enclick = True**

**def pos(self, n):**

**self.x = n \* wix // 4**

**def update(self, screen):**

**if self.enclick:**

**pygame.draw.rect(screen, (0, 0, 0), [self.x, self.y, self.h, self.l])**

**else:**

**pygame.draw.rect(screen, (180, 180, 180), [self.x, self.y, self.h, self.l])**

**def click(self, ps):**

**if ps[0] in range(self.x, self.x + self.h):**

**if ps[1] in range(self.y, self.y + self.l):**

**self.enclick = False**

**return 0**

**return 1**

***#def health():***

**def game():**

**pygame.mixer.music.load(songs[int(gameMenu.song)])**

**pygame.mixer.music.play(1)**

**pygame.mixer.music.set\_volume(0.5)**

***#MusicScreen()***

**clock = pygame.time.Clock()**

**screen = pygame.display.set\_mode((WIDTH, HEIGHT))**

**ingame = 0**

**time = 0**

**delt = 60**

**block = []**

**speed = 3**

**score = 0**

**while ingame == 0:**

**for i in range(35):**

**block.append(button())**

**block[-1].pos(random.randrange(4)) *# (i)***

**if ingame != 0: break**

**for j in range(wiy // (5 \* speed)+1):**

**time += 1 / delt**

**clock.tick(delt)**

**screen.fill((224, 224, 255))**

**if ingame != 0: break**

**for k in range(len(block)):**

**try:**

**block[k].y += speed**

**block[k].update(screen)**

**if block[k].y > wiy - block[k].l and block[k].enclick == True:**

**ingame = 1**

**except:**

**pass**

**for event in pygame.event.get():**

**if event.type == QUIT or \**

**(event.type == KEYDOWN and event.key == K\_ESCAPE):**

**pygame.quit()**

**elif event.type == MOUSEBUTTONDOWN:**

**ingame = block[score].click(pygame.mouse.get\_pos())**

**score += 1**

**msg(screen, "SCORE " + str(score), color=(0, 128, 255), pos=(-1, 30))**

**pygame.draw.line(gameWindow, GREY, (200, 0), (200, 600), 1)**

**pygame.draw.line(gameWindow, GREY, (400, 0), (400, 600), 1)**

**pygame.draw.line(gameWindow, GREY, (600, 0), (600, 600), 1)**

**pygame.draw.line(gameWindow, GREY, (800, 0), (800, 600), 1)**

**pygame.display.update()**

**if speed <= 9:**

**speed += 1**

***#clock.tick(100)***

**msg(screen, str(main.text), color=(110, 128, 225), size=75, pos=(400,200))**

**msg(screen, " YOU LOSE ", color=(110, 128, 225), size=100, pos=(-1, -1))**

**pygame.mixer.music.stop()**

**msg(screen, "SCORE " + str(score-1), color=(0, 128, 255), pos=(-1, 50))**

**msg(screen, ' press BACKSPACE to play again', color=(110, 128, 225), size=50, pos=(400, 20))**

**'''if keys[pygame.K\_BACKSPACE]:**

**ingame = 0'''**

***#pygame.display.update()***

***#pygame.time.wait(4000)***

**def tutorial():**

**pygame.mixer.music.load("DoubleAgent.mp3")**

**pygame.mixer.music.play(1)**

**pygame.mixer.music.set\_volume(0.5)**

**clock = pygame.time.Clock()**

**screen = pygame.display.set\_mode((WIDTH, HEIGHT))**

**ingame = 0**

**time = 0**

**delt = 60**

**block = []**

**speed = 1**

**score = 0**

**playAgain = False**

**while ingame == 0:**

**for i in range(25):**

**block.append(button())**

**block[-1].pos(random.randrange(4)) *# (i)***

**if ingame != 0: break**

**for j in range(wiy // (5 \* speed) + 1):**

**time += 1 / delt**

**clock.tick(delt)**

**screen.fill((224, 224, 255))**

***#msg(screen,"Press A,S,J,K on the keyboard when the blocks land on the bottom. If the block is outside, then 10 health will be taken from the hp bar. The score will add up each time.")***

**if ingame != 0: break**

**for k in range(len(block)):**

**try:**

**block[k].y += speed**

**block[k].update(screen)**

**if block[k].y > wiy - block[k].l and block[k].enclick == True:**

**ingame = 1**

**except:**

**pass**

**for event in pygame.event.get():**

**if event.type == QUIT or \**

**(event.type == KEYDOWN and event.key == K\_ESCAPE):**

**pygame.quit()**

**elif event.type == MOUSEBUTTONDOWN:**

**ingame = block[score].click(pygame.mouse.get\_pos())**

**score += 1**

**msg(screen, "SCORE " + str(score), color=(0, 128, 255), pos=(-1, 30))**

**pygame.draw.line(gameWindow, GREY, (200, 0), (200, 600), 1)**

**pygame.draw.line(gameWindow, GREY, (400, 0), (400, 600), 1)**

**pygame.draw.line(gameWindow, GREY, (600, 0), (600, 600), 1)**

**pygame.draw.line(gameWindow, GREY, (800, 0), (800, 600), 1)**

**pygame.display.update()**

**if speed <= 9:**

**speed += 1**

**msg(screen, "YOU LOSE ", color=(110, 128, 225), size=100, pos=(-1, -1))**

**pygame.mixer.music.stop()**

**msg(screen, ' press BACKSPACE to play again', color=(110, 128, 225), size=50, pos=(400, 20))**

**pygame.display.update()**

**pygame.time.wait(4000)**

**count = 0**

**pygame.mixer.music.play(-1)**

**pygame.mixer.music.set\_volume(0.5)**

**pygame.event.get()**

**keys = pygame.key.get\_pressed()**

**while running:**

**if count == 0:**

**count = 1**

**main()**

**print (main.text)**

**mainMenu()**

**gameMenu()**

**transition(800,600)**

**'''pygame.mixer.music.load(songs[int(gameMenu.song)])**

**pygame.mixer.music.play(-1)**

**pygame.mixer.music.set\_volume(0.5)'''**

**game()**

**pygame.event.get()**

**keys = pygame.key.get\_pressed()**

**if keys[pygame.K\_ESCAPE]:**

**running = False**

**pygame.display.update()**

**pygame.quit()**

Keyboard mode

**import** pygame, os, random

**from** pygame.locals **import** \*

pygame.init()

wix = 800

wiy = 600

WIDTH = 800

HEIGHT = 600

BLUE = (0, 0, 77)

GREY = (0, 0, 0)

WHITE = (255,255,255)

BLACK = (0,0,0)

gameWindow = pygame.display.set\_mode((WIDTH, HEIGHT))

**def** grid():

**for** x **in** range(0, WIDTH, 10):

pygame.draw.line(gameWindow, BLACK, (x, 0), (x, HEIGHT), 1)

**for** y **in** range(0, HEIGHT, 10):

pygame.draw.line(gameWindow, BLACK, (0, y), (WIDTH, y), 1)

**for** x **in** range(0, WIDTH, 10 \* 10):

pygame.draw.line(gameWindow, BLACK, (x, 0), (x, HEIGHT), 2)

**for** y **in** range(0, HEIGHT, 10 \* 10):

pygame.draw.line(gameWindow, BLACK, (0, y), (WIDTH, y), 2)

**def** msg(screen, text, color=(55, 55, 55), size=36, pos=(-1, -1)):

**if** pos[0] == -1: pos = (screen.get\_rect().centerx, pos[1])

**if** pos[1] == -1: pos = (pos[0], screen.get\_rect().centery)

font = pygame.font.Font(None, size)

text = font.render(text, 1, color)

textpos = text.get\_rect()

textpos.centerx = pos[0]

textpos.centery = pos[1]

screen.blit(text, textpos)

bottomBlock = pygame.image.load(**"BottomBlock.png"**).convert\_alpha()

bottomBlock = pygame.transform.scale(bottomBlock, (195, 120))

font = pygame.font.SysFont(**"Monospaced"**,130)

graphics = font.render(**"A"**, 1, WHITE)

graphics1 = font.render(**"S"**, 1, WHITE)

graphics2 = font.render(**"J"**, 1, WHITE)

graphics3 = font.render(**"K"**, 1, WHITE)

xPosition = []

**class** button():

x=0

y=-wiy//5

h=wix//4-1

l=wiy//5

enpress = True

**def** pos(self,n):

self.x=n\*wix//4

**def** update(self,screen):

**if** self.enpress:

pygame.draw.rect(screen,(0,0,0),[self.x,self.y,self.h,self.l])

**global** selfY

selfY = self.y

**print** self.y

*#xPosition.append(self.x)*

*#print xPosition'''*

**else** :

pygame.draw.rect(screen,(180,180,180),[self.x,self.y,self.h,self.l])

**'''def press(self,ps):**

**if ps[0] in range(self.x,self.x+self.h):**

**if ps[1] in range (self.y,self.y+self.l):**

**self.enpress = False**

**return 0**

**return 1'''**

pygame.init()

clock = pygame.time.Clock()

screen = pygame.display.set\_mode((wix, wiy))

time = 0

delt = 60

block = []

speed = 3

score = 0

clock = pygame.time.Clock()

x1 = 200

y1 = 0

x2 = 200

y2 = 600

cnt = 0

ingame = 0

**while** ingame == 0:

**for** i **in** range(10):

a = random.randrange(4)

block.append(button())

block[-1].pos(a)

*#print a*

**if** ingame != 0:

**break**

**for** j **in** range(wiy // (5 \* speed) + 1):

time += 1 / delt

clock.tick(delt)

screen.fill((224, 224, 255))

**if** ingame != 0:

**break**

**for** k **in** range(len(block)):

**try**:

block[k].y += speed

block[k].update(screen)

**'''if block[k].y > wiy - block[k].l and block[k].enpress == True:**

**ingame = 1'''**

**except**:

**pass**

**for** event **in** pygame.event.get():

**if** (event.type == KEYDOWN **and** event.key == K\_ESCAPE):

pygame.quit()

**elif** event.type == pygame.K\_a **and** a == 0 **and** selfY <= 260:

**print "gucci"**

*#msg(screen, "SCORE " + str(score), color=(0, 128, 255), pos=(-1, 30))*

pygame.draw.line(gameWindow, GREY, (200, 0), (200, 600), 1)

pygame.draw.line(gameWindow, GREY, (400, 0), (400, 600), 1)

pygame.draw.line(gameWindow, GREY, (600, 0), (600, 600), 1)

pygame.draw.line(gameWindow, GREY, (800, 0), (800, 600), 1)

gameWindow.blit(bottomBlock, (0, 460))

gameWindow.blit(bottomBlock, (200, 460))

gameWindow.blit(bottomBlock, (400, 460))

gameWindow.blit(bottomBlock, (600, 460))

gameWindow.blit(graphics,(60, 480))

gameWindow.blit(graphics1,(260, 480))

gameWindow.blit(graphics2,(460, 480))

gameWindow.blit(graphics3,(660, 480))

pygame.display.update()

*#grid()*

*#speed += 1*

msg(screen, **"YOU LOSE "**, color=(110, 128, 225), size=100, pos=(-1, -1))

pygame.display.update()

*# pygame.time.wait(4000)*

pygame.quit()

Keyboard mode 2

**import** pygame, os, random

**from** pygame.locals **import** \*

pygame.init()

wix = 800

wiy = 600

WIDTH = 800

HEIGHT = 600

BLUE = (0, 0, 77)

GREY = (0, 0, 0)

WHITE = (255,255,255)

gameWindow = pygame.display.set\_mode((WIDTH, HEIGHT))

**def** msg(screen, text, color=(55, 55, 55), size=36, pos=(-1, -1)):

**if** pos[0] == -1: pos = (screen.get\_rect().centerx, pos[1])

**if** pos[1] == -1: pos = (pos[0], screen.get\_rect().centery)

font = pygame.font.Font(None, size)

text = font.render(text, 1, color)

textpos = text.get\_rect()

textpos.centerx = pos[0]

textpos.centery = pos[1]

screen.blit(text, textpos)

bottomBlock = pygame.image.load(**"BottomBlock.png"**).convert\_alpha()

bottomBlock = pygame.transform.scale(bottomBlock, (195, 120))

font = pygame.font.SysFont(**"Monospaced"**,130)

graphics = font.render(**"A"**, 1, WHITE)

graphics1 = font.render(**"S"**, 1, WHITE)

graphics2 = font.render(**"J"**, 1, WHITE)

graphics3 = font.render(**"K"**, 1, WHITE)

**'''class button():**

**x=0**

**y=-wiy//5**

**h=wix//4-1**

**l=wiy//5**

**enclick = True**

**def pos(self,n):**

**self.x=n\*wix//4**

**def update(self,screen):**

**if self.enclick :**

**pygame.draw.rect(screen,(0,0,0),[self.x,self.y,self.h,self.l])**

**else :**

**pygame.draw.rect(screen,(180,180,180),[self.x,self.y,self.h,self.l])**

**'''**

pygame.init()

clock = pygame.time.Clock()

screen = pygame.display.set\_mode((wix, wiy))

lost = 0

time = 0

delt = 60

sb = []

speey = 3

score = 0

clock = pygame.time.Clock()

cnt = 0

running = True

blocks = []

x,y = 0,0

**while** running:

screen.fill((255,255,255))

**for** event **in** pygame.event.get():

**if** (event.type == KEYDOWN **and** event.key == K\_ESCAPE):

pygame.quit()

pygame.draw.rect(screen,GREY,(0,y,200,100))

y += 1

**if** y == 600:

y = 0

msg(screen, **"SCORE "** + str(score), color=(0, 128, 255), pos=(-1, 30))

pygame.draw.line(gameWindow, GREY, (200, 0), (200, 600), 1)

pygame.draw.line(gameWindow, GREY, (400, 0), (400, 600), 1)

pygame.draw.line(gameWindow, GREY, (600, 0), (600, 600), 1)

pygame.draw.line(gameWindow, GREY, (800, 0), (800, 600), 1)

gameWindow.blit(bottomBlock, (0, 460))

gameWindow.blit(bottomBlock, (200, 460))

gameWindow.blit(bottomBlock, (400, 460))

gameWindow.blit(bottomBlock, (600, 460))

gameWindow.blit(graphics,(60, 480))

gameWindow.blit(graphics1,(260, 480))

gameWindow.blit(graphics2,(460, 480))

gameWindow.blit(graphics3,(660, 480))

pygame.display.update()

msg(screen, **"YOU LOSE "**, color=(110, 128, 225), size=100, pos=(-1, -1))

pygame.display.update()

*# pygame.time.wait(4000)*

pygame.quit()

Jan 16

**import pygame, os, sys**

**import random**

**from pygame.locals import \***

**pygame.init()**

**print "Enter your username"**

**pygame.init()**

**WIDTH = 800**

**HEIGHT = 600**

**wix = 800**

**wiy = 600**

**gameWindow = pygame.display.set\_mode((WIDTH, HEIGHT))**

**GRIDSIZE = 10**

**RED = (255, 0, 0)**

**GREEN = (0, 255, 0)**

**BLUE = (0, 0, 255)**

**CYAN = (0, 255, 255)**

**WHITE = (255, 255, 255)**

**BLACK = (0, 0, 0)**

**GREY = (128, 128, 128)**

**YELLOW = (255,255,0)**

**PINK = (255, 102, 128)**

**WHITE2 = (224, 224, 255)**

**dark\_WHITE = (224,244,255)**

**songs = ["PutOnYourSundayClothes.mp3",'Dreams.mp3', 'WeStayedUpAllNight.mp3', "Masquerade.mp3","Nevada.mp3","Predawn.mp3","Grafiore.mp3","Howling.mp3","PandorA.mp3","Phototropic.mp3","DoubleAgent.mp3"]**

**pygame.mixer.music.load("DoubleAgent.mp3")**

**def backGround():**

**picture = pygame.image.load("Background(1).jpg").convert\_alpha()**

**picture = pygame.transform.scale(picture, (800, 600))**

**gameWindow.blit(picture, (0, 0))**

**def background1():**

**gameMenu = pygame.image.load("GameMenu.jpg").convert\_alpha()**

**gameMenu = pygame.transform.scale(gameMenu, (800, 600))**

**gameWindow.blit(gameMenu, (0, 0))**

**running = True**

**def grid():**

**for x in range(0, WIDTH, 10):**

**pygame.draw.line(gameWindow, BLACK, (x, 0), (x, HEIGHT), 1)**

**for y in range(0, HEIGHT, 10):**

**pygame.draw.line(gameWindow, BLACK, (0, y), (WIDTH, y), 1)**

**for x in range(0, WIDTH, 10 \* 10):**

**pygame.draw.line(gameWindow, BLACK, (x, 0), (x, HEIGHT), 2)**

**for y in range(0, HEIGHT, 10 \* 10):**

**pygame.draw.line(gameWindow, BLACK, (0, y), (WIDTH, y), 2)**

**def main():**

**global mouseMode,keyboardMode**

**mouseMode = False**

**keyboardMode = False**

**backGround()**

**font = pygame.font.Font(None, 32)**

**clock = pygame.time.Clock()**

**input\_box = pygame.Rect(460, 230, 400, 54)**

**color\_inactive = pygame.Color('lightskyblue3')**

**color\_active = pygame.Color('dodgerblue2')**

**color = color\_inactive**

**mouse = pygame.image.load("Mouse.png").convert\_alpha()**

**mouse = pygame.transform.scale(mouse,(70,70))**

**keyboard = pygame.image.load("Keyboard.png").convert\_alpha()**

**keyboard = pygame.transform.scale(keyboard,(150,100))**

**active = False**

**main.text = ''**

**done = False**

**drawRect = False**

**drawRect2 = False**

**w = 0**

**l = 0**

**x = 0**

**y = 0**

**while not done:**

**backGround()**

**gameWindow.blit(mouse,(475,320))**

**gameWindow.blit(keyboard,(600,300))**

**mouseX, mouseY = pygame.mouse.get\_pos()**

**if drawRect == True or drawRect2 == True:**

**border = pygame.image.load("Border.png").convert\_alpha()**

**border = pygame.transform.scale(border,(w,l))**

**gameWindow.blit(border, (x, y))**

**for event in pygame.event.get():**

**if mouseX >= 490 and mouseX <= 530 and mouseY >= 320 and mouseY <= 390:**

**w = 250**

**l = 430**

**x = 385**

**y = 170**

**drawRect = True**

**if event.type == pygame.MOUSEBUTTONDOWN:**

**mouseMode = True**

**print "Mouse Mode"**

**else:**

**drawRect = False**

**if mouseX >= 600 and mouseX <= 750 and mouseY >= 330 and mouseY <= 390:**

**w = 550**

**l = 500**

**x = 400**

**y = 150**

**drawRect2 = True**

**if event.type == pygame.MOUSEBUTTONDOWN:**

**keyboardMode = True**

**print "Keyboard mode"**

**else:**

**drawRect2 = False**

**if event.type == pygame.QUIT:**

**done = True**

**if event.type == pygame.MOUSEBUTTONDOWN:**

***# If the user clicked on the input\_box rect.***

**if input\_box.collidepoint(event.pos):**

***# Toggle the active variable.***

**active = not active**

**else:**

**active = False**

***# Change the current color of the input box.***

**color = color\_active if active else color\_inactive**

**if event.type == pygame.KEYDOWN:**

**if active:**

**if event.key == pygame.K\_RETURN:**

**print 'Your username has been set to:', main.text**

**done = True**

**elif event.key == pygame.K\_BACKSPACE:**

**main.text = main.text[:-1]**

**else:**

**main.text += event.unicode**

***# Render the current text.***

**txt\_surface = font.render(main.text, True, color)**

***# Resize the box if the text is too long.***

**width = max(300, txt\_surface.get\_width() + 10)**

**input\_box.w = width**

***#backGround()***

***# Blit the text.***

**gameWindow.blit(txt\_surface, (input\_box.x + 5, input\_box.y + 5))**

**msg(gameWindow, 'BONK BONK', color=(180,180,180), size=100, pos=(400,50))**

***# Blit the input\_box rect.***

***#grid()***

**pygame.draw.rect(gameWindow, color, input\_box, 2)**

**pygame.display.update()**

**clock.tick(30)**

**def redrawGameWindow():**

**gameWindow.blit(ship, (shipX,shipY))**

**'''background = pygame.image.load("BackGround(2).jpg").convert\_alpha()**

**background = pygame.transform.scale(background, (800, 600))**

**gameWindow.blit(background, (0, 0))'''**

**pygame.display.update()**

**def rotate(image, angle):**

**ORIGINALrect = image.get\_rect()**

**rotatedImage = pygame.transform.rotate(image,angle)**

**rotatedRect = ORIGINALrect.copy()**

**rotatedRect.center = rotatedImage.get\_rect().center**

**rotatedImage = rotatedImage.subsurface(rotatedRect).copy()**

**return rotatedImage**

***#--------------------------------------------------------------#***

***#--------------------------------------------------------------#***

**def mainMenu():**

**global tutorials**

**tutorials = False**

**background2 = pygame.image.load("BackGround(2).jpg").convert\_alpha()**

**background2 = pygame.transform.scale(background2, (800, 600))**

**gameWindow.blit(background2, (0, 0))**

**pygame.display.update()**

**clock = pygame.time.Clock()**

**clock.tick(FPS)**

**pygame.event.get()**

**keys = pygame.key.get\_pressed()**

**inPlay = True**

**global ship, shipY, shipX, shipAngle**

**ORIGINALship = pygame.image.load("Pendulum(2).png").convert\_alpha()**

**ORIGINALship = pygame.transform.scale(ORIGINALship, (400, 400))**

***# backGround = pygame.image.load("BackGround(2).jpg")***

***# backGround = pygame.transform.scale(backGround, (800, 600))***

**ship = ORIGINALship.copy()**

**shipX = -10**

**shipY = 100**

**shipAngle = 0**

**rotationStep = 4**

**active = False**

**while inPlay:**

**redrawGameWindow()**

***# gameWindow.blit(backGround, (0, 0))***

**clock.tick(FPS)**

**pygame.event.get()**

**shipAngle = shipAngle + rotationStep**

**ship = rotate(ORIGINALship, shipAngle)**

***#pygame.display.update()***

**if keys[pygame.K\_ESCAPE]:**

**inPlay = False**

**for event in pygame.event.get():**

**if event.type == pygame.MOUSEBUTTONDOWN and mouseX <=710 and mouseX>=390 and mouseY >=170 and mouseY <= 240:**

**transition(800,600)**

**inPlay = False**

**if event.type == pygame.MOUSEBUTTONDOWN and mouseX <=730 and mouseX>=400 and mouseY >=260 and mouseY <= 320:**

**tutorials = True**

**transition(800,600)**

**inPlay = False**

**if event.type == pygame.MOUSEBUTTONDOWN and mouseX <=710 and mouseX>=390 and mouseY >=340 and mouseY <= 410:**

**transition(800,600)**

**pygame.quit()**

**mouseX, mouseY = pygame.mouse.get\_pos()**

***#grid()***

***#---------------------------------------------#***

**def shop():**

**font = pygame.font.SysFont("Monospaced", 30)**

**graphics = font.render("10 Bonkcoins", 1, WHITE)**

**graphics1 = font.render("10 Bonkcoins", 1, WHITE)**

**graphics2 = font.render("10 Bonkcoins", 1, WHITE)**

**graphics3 = font.render("10 Bonkcoins", 1, WHITE)**

**graphics4 = font.render("10 Bonkcoins", 1, WHITE)**

**graphics5 = font.render("20 Bonkcoins", 1, WHITE)**

**shopBackground = pygame.image.load("ShopBackground.jpg").convert\_alpha()**

**shopBackground = pygame.transform.scale(shopBackground,(800,600))**

**bonkCoins = pygame.image.load("Coins.png").convert\_alpha()**

**bonkCoins = pygame.transform.scale(bonkCoins,(100,100))**

**backButton = pygame.image.load("BackButton.png").convert\_alpha()**

**backButton = pygame.transform.scale(backButton,(150,75))**

**inShop = True**

**while inShop:**

**mouseX, mouseY = pygame.mouse.get\_pos()**

**gameWindow.blit(shopBackground,(0,0))**

**pygame.draw.rect(gameWindow,BLUE, (50,100,200,100))**

**pygame.draw.rect(gameWindow, RED, (300, 100, 200, 100))**

**pygame.draw.rect(gameWindow, YELLOW, (550, 100, 200, 100))**

**pygame.draw.rect(gameWindow, GREEN, (50, 300, 200, 100))**

**pygame.draw.rect(gameWindow, PINK, (300, 300, 200, 100))**

**pygame.draw.rect(gameWindow, WHITE2, (550, 300, 200, 100))**

**gameWindow.blit(bonkCoins,(600,0))**

**gameWindow.blit(graphics, (75, 225))**

**gameWindow.blit(graphics1, (325, 225))**

**gameWindow.blit(graphics2, (575, 225))**

**gameWindow.blit(graphics3, (75, 425))**

**gameWindow.blit(graphics4, (325, 425))**

**gameWindow.blit(graphics5, (575, 425))**

**gameWindow.blit(backButton,(0,500))**

**pygame.display.update()**

**for event in pygame.event.get():**

**if (event.type == KEYDOWN and event.key == K\_ESCAPE):**

**inShop = False**

**pygame.quit()**

**elif event.type == pygame.MOUSEBUTTONDOWN and mouseX >= 0 and mouseX <= 150 and mouseY >= 500 and mouseY <= 570:**

**inshop = False**

***#break***

***#---------------------------------------#***

**print "Hit ESC to end the program."**

**clock = pygame.time.Clock()**

**FPS = 30**

**def gameMenu():**

**pygame.mixer.music.stop**

**background1()**

**font = pygame.font.Font(None, 32)**

**clock = pygame.time.Clock()**

**input\_box = pygame.Rect(425, 230, 310, 60)**

**color\_inactive = pygame.Color('lightskyblue3')**

**color\_active = pygame.Color('dodgerblue2')**

**color = color\_inactive**

**active = False**

**gameMenu.song = ''**

**done = False**

**true = False**

**global openShop**

**openShop = False**

**pygame.init()**

**while not done:**

**for event in pygame.event.get():**

**mouseX1, mouseY1 = pygame.mouse.get\_pos()**

**if event.type == pygame.QUIT:**

**done = True**

**if event.type == pygame.MOUSEBUTTONDOWN and mouseX1 <= 790 and mouseX1 >= 610 and mouseY1 >= 520 and mouseY1 <= 600:**

**transition(800,600)**

**done = True**

**pygame.quit()**

**if event.type == pygame.MOUSEBUTTONDOWN and mouseX1 >= 490 and mouseX1 <= 790 and mouseY1 >= 10 and mouseY1 <= 70:**

**openShop = True**

**if event.type == pygame.MOUSEBUTTONDOWN:**

***# If the user clicked on the input\_box rect.***

**if input\_box.collidepoint(event.pos):**

***# Toggle the active variable.***

**active = not active**

**else:**

**active = False**

***# Change the current color of the input box.***

**color = color\_active if active else color\_inactive**

**if event.type == pygame.KEYDOWN:**

**if active:**

**if event.key == pygame.K\_RETURN:**

**done = True**

**elif event.key == pygame.K\_BACKSPACE:**

**gameMenu.song = gameMenu.song[:-1]**

**else:**

**gameMenu.song += event.unicode**

***# Render the current text.***

**txt\_surface = font.render(gameMenu.song, True, color)**

***# Resize the box if the text is too long.***

**width = max(270, txt\_surface.get\_width() + 10)**

**input\_box.w = width**

**background1()**

***# Blit the text.***

**gameWindow.blit(txt\_surface, (input\_box.x + 5, input\_box.y + 5))**

***# Blit the input\_box rect.***

**pygame.draw.rect(gameWindow, color, input\_box, 2)**

**pygame.display.update()**

**clock.tick(30)**

***#grid()***

**def transition(width,height):**

**fade = pygame.Surface((width,height))**

**fade.fill((0,0,0))**

**for alpha in range(50):**

**fade.set\_alpha(alpha)**

**gameWindow.blit(fade,(0,0))**

**pygame.display.update()**

**pygame.time.delay(10)**

**def MusicScreen():**

**musicbackground1 = pygame.image.load("MusicBackGround(1).png").convert\_alpha()**

**musicbackground2 = pygame.image.load("MusicBackGround(2).png").convert\_alpha()**

**musicbackground3 = pygame.image.load("MusicBackGround(3).png").convert\_alpha()**

**musicbackground4 = pygame.image.load("MusicBackGround(4).png").convert\_alpha()**

**musicbackground5 = pygame.image.load("MusicBackGround(5).png").convert\_alpha()**

**musicbackground6 = pygame.image.load("MusicBackGround(6).png").convert\_alpha()**

**musicbackground7 = pygame.image.load("MusicBackGround(7).png").convert\_alpha()**

**musicbackground8 = pygame.image.load("MusicBackGround(8).png").convert\_alpha()**

**musicbackground9 = pygame.image.load("MusicBackGround(9).png").convert\_alpha()**

**musicbackground10 = pygame.image.load("MusicBackGround(10).png").convert\_alpha()**

**musicbackground1 = pygame.transform.scale(musicbackground1,(400,400))**

**musicbackground2 = pygame.transform.scale(musicbackground2, (600, 400))**

**musicbackground3 = pygame.transform.scale(musicbackground3, (550, 400))**

**musicbackground4 = pygame.transform.scale(musicbackground4, (600, 400))**

**musicbackground5 = pygame.transform.scale(musicbackground5, (650, 400))**

**musicbackground6 = pygame.transform.scale(musicbackground6, (525, 400))**

**musicbackground7 = pygame.transform.scale(musicbackground7, (450, 400))**

**musicbackground8 = pygame.transform.scale(musicbackground8, (550, 400))**

**musicbackground9 = pygame.transform.scale(musicbackground9, (600, 400))**

**musicbackground10 = pygame.transform.scale(musicbackground10, (525, 400))**

**running = True**

**global speedNum**

**speedNum = 4**

**drawRect,drawRect2,drawRect3 = False,False,False**

**while running:**

**mouseX, mouseY = pygame.mouse.get\_pos()**

**background1()**

**if gameMenu.song == "1":**

**gameWindow.blit(musicbackground2, (100, 75))**

**if drawRect == True or drawRect2 == True or drawRect3 == True:**

**box\_surface\_rect = pygame.Surface((140, 50), pygame.SRCALPHA)**

**pygame.draw.rect(box\_surface\_rect, (255, 255, 255, 90), (0, 0, 80, 50))**

**gameWindow.blit(box\_surface\_rect, (x, y))**

**for event in pygame.event.get():**

**if event.type == pygame.MOUSEBUTTONDOWN and mouseX >= 320 and mouseX <= 490 and mouseY >= 340 and mouseY <= 400:**

**running = False**

**if mouseX >= 260 and mouseX <= 350 and mouseY >= 220 and mouseY <= 270:**

**drawRect = True**

**x = 265**

**y = 220**

**if event.type == pygame.MOUSEBUTTONDOWN:**

**print "Easy"**

**speedNum = 2**

**else:**

**drawRect = False**

**if mouseX >= 370 and mouseX <= 450 and mouseY >= 220 and mouseY <= 270:**

**drawRect2 = True**

**x = 370**

**y = 220**

**if event.type == pygame.MOUSEBUTTONDOWN:**

**print "Normal"**

**speedNum = 4**

**else:**

**drawRect2 = False**

**if mouseX >= 470 and mouseX <= 550 and mouseY >= 220 and mouseY <= 270:**

**drawRect3 = True**

**x = 470**

**y = 220**

**if event.type == pygame.MOUSEBUTTONDOWN:**

**print "Hard"**

**speedNum = 6**

**else:**

**drawRect3 = False**

**elif gameMenu.song == "2":**

**gameWindow.blit(musicbackground3, (125, 100))**

**if drawRect == True or drawRect2 == True or drawRect3 == True:**

**box\_surface\_rect = pygame.Surface((140, 50), pygame.SRCALPHA)**

**pygame.draw.rect(box\_surface\_rect, (255, 255, 255, 90), (0, 0, 80, 50))**

**gameWindow.blit(box\_surface\_rect, (x, y))**

**for event in pygame.event.get():**

**if event.type == pygame.MOUSEBUTTONDOWN and mouseX >= 340 and mouseX <= 460 and mouseY >= 300 and mouseY <= 400:**

**running = False**

**if mouseX >= 250 and mouseX <= 330 and mouseY >= 230 and mouseY <= 280:**

**drawRect = True**

**x = 250**

**y = 230**

**if event.type == pygame.MOUSEBUTTONDOWN:**

**print "Easy"**

**speedNum = 2**

**else:**

**drawRect = False**

**if mouseX >= 360 and mouseX <= 440 and mouseY >= 230 and mouseY <= 280:**

**drawRect2 = True**

**x = 360**

**y = 230**

**if event.type == pygame.MOUSEBUTTONDOWN:**

**print "Normal"**

**speedNum = 4**

**else:**

**drawRect2 = False**

**if mouseX >= 470 and mouseX <= 550 and mouseY >= 230 and mouseY <= 280:**

**drawRect3 = True**

**x = 470**

**y = 230**

**if event.type == pygame.MOUSEBUTTONDOWN:**

**print "Hard"**

**speedNum = 6**

**else:**

**drawRect3 = False**

**elif gameMenu.song == "3":**

**gameWindow.blit(musicbackground4, (90, 75))**

**if drawRect == True or drawRect2 == True or drawRect3 == True:**

**box\_surface\_rect = pygame.Surface((140, 50), pygame.SRCALPHA)**

**pygame.draw.rect(box\_surface\_rect, (255, 255, 255, 90), (0, 0, 90, 40))**

**gameWindow.blit(box\_surface\_rect, (x, y))**

**for event in pygame.event.get():**

**if event.type == pygame.MOUSEBUTTONDOWN and mouseX >= 340 and mouseX <= 450 and mouseY >= 340 and mouseY <= 380:**

**running = False**

**if mouseX >= 240 and mouseX <= 330 and mouseY >= 220 and mouseY <= 260:**

**drawRect = True**

**x = 240**

**y = 220**

**if event.type == pygame.MOUSEBUTTONDOWN:**

**print "Easy"**

**speedNum = 2**

**else:**

**drawRect = False**

**if mouseX >= 350 and mouseX <= 440 and mouseY >= 220 and mouseY <= 260:**

**drawRect2 = True**

**x = 350**

**y = 220**

**if event.type == pygame.MOUSEBUTTONDOWN:**

**print "Normal"**

**speedNum = 4**

**else:**

**drawRect2 = False**

**if mouseX >= 470 and mouseX <= 550 and mouseY >= 220 and mouseY <= 260:**

**drawRect3 = True**

**x = 470**

**y = 220**

**if event.type == pygame.MOUSEBUTTONDOWN:**

**print "Hard"**

**speedNum = 6**

**else:**

**drawRect3 = False**

**elif gameMenu.song == "4":**

**gameWindow.blit(musicbackground1, (200, 75))**

**if drawRect == True or drawRect2 == True or drawRect3 == True:**

**box\_surface\_rect = pygame.Surface((140, 50), pygame.SRCALPHA)**

**pygame.draw.rect(box\_surface\_rect, (255, 255, 255, 90), (0, 0, 90, 50))**

**gameWindow.blit(box\_surface\_rect, (x, y))**

**for event in pygame.event.get():**

**if event.type == pygame.MOUSEBUTTONDOWN and mouseX >= 300 and mouseX <= 500 and mouseY >= 350 and mouseY <= 420:**

**running = False**

**if mouseX >= 220 and mouseX <= 310 and mouseY >= 210 and mouseY <= 260:**

**drawRect = True**

**x = 220**

**y = 210**

**if event.type == pygame.MOUSEBUTTONDOWN:**

**print "Easy"**

**speedNum = 2**

**else:**

**drawRect = False**

**if mouseX >= 350 and mouseX <= 440 and mouseY >= 210 and mouseY <= 260:**

**drawRect2 = True**

**x = 350**

**y = 210**

**if event.type == pygame.MOUSEBUTTONDOWN:**

**print "Normal"**

**speedNum = 4**

**else:**

**drawRect2 = False**

**if mouseX >= 480 and mouseX <= 570 and mouseY >= 210 and mouseY <= 260:**

**drawRect3 = True**

**x = 480**

**y = 210**

**if event.type == pygame.MOUSEBUTTONDOWN:**

**print "Hard"**

**speedNum = 6**

**else:**

**drawRect3 = False**

**elif gameMenu.song == "5":**

**gameWindow.blit(musicbackground5, (65, 75))**

**if drawRect == True or drawRect2 == True or drawRect3 == True:**

**box\_surface\_rect = pygame.Surface((140, 50), pygame.SRCALPHA)**

**pygame.draw.rect(box\_surface\_rect, (255, 255, 255, 90), (0, 0, 85, 40))**

**gameWindow.blit(box\_surface\_rect, (x, y))**

**for event in pygame.event.get():**

**if event.type == pygame.MOUSEBUTTONDOWN and mouseX >= 350 and mouseX <= 450 and mouseY >= 340 and mouseY <= 380:**

**running = False**

**if mouseX >= 230 and mouseX <= 310 and mouseY >= 220 and mouseY <= 260:**

**drawRect = True**

**x = 230**

**y = 220**

**if event.type == pygame.MOUSEBUTTONDOWN:**

**print "Easy"**

**speedNum = 2**

**else:**

**drawRect = False**

**if mouseX >= 360 and mouseX <= 440 and mouseY >= 220 and mouseY <= 260:**

**drawRect2 = True**

**x = 360**

**y = 220**

**if event.type == pygame.MOUSEBUTTONDOWN:**

**print "Normal"**

**speedNum = 4**

**else:**

**drawRect2 = False**

**if mouseX >= 480 and mouseX <= 560 and mouseY >= 220 and mouseY <= 260:**

**drawRect3 = True**

**x = 480**

**y = 220**

**if event.type == pygame.MOUSEBUTTONDOWN:**

**print "Hard"**

**speedNum = 6**

**else:**

**drawRect3 = False**

**elif gameMenu.song == "6":**

**gameWindow.blit(musicbackground6, (125, 100))**

**if drawRect == True or drawRect2 == True or drawRect3 == True:**

**box\_surface\_rect = pygame.Surface((140, 50), pygame.SRCALPHA)**

**pygame.draw.rect(box\_surface\_rect, (255, 255, 255, 90), (0, 0, 100, 47))**

**gameWindow.blit(box\_surface\_rect, (x, y))**

**for event in pygame.event.get():**

**if event.type == pygame.MOUSEBUTTONDOWN and mouseX >= 320 and mouseX <= 460 and mouseY >= 370 and mouseY <= 420:**

**running = False**

**if mouseX >= 220 and mouseX <= 320 and mouseY >= 240 and mouseY <= 290:**

**drawRect = True**

**x = 223**

**y = 243**

**if event.type == pygame.MOUSEBUTTONDOWN:**

**print "Easy"**

**speedNum = 2**

**else:**

**drawRect = False**

**if mouseX >= 340 and mouseX <= 440 and mouseY >= 240 and mouseY <= 290:**

**drawRect2 = True**

**x = 343**

**y = 243**

**if event.type == pygame.MOUSEBUTTONDOWN:**

**print "Normal"**

**speedNum = 4**

**else:**

**drawRect2 = False**

**if mouseX >= 460 and mouseX <= 560 and mouseY >= 240 and mouseY <= 290:**

**drawRect3 = True**

**x = 463**

**y = 243**

**if event.type == pygame.MOUSEBUTTONDOWN:**

**print "Hard"**

**speedNum = 6**

**else:**

**drawRect3 = False**

**elif gameMenu.song == "7":**

**gameWindow.blit(musicbackground7, (175, 100))**

**if drawRect == True or drawRect2 == True or drawRect3 == True:**

**box\_surface\_rect = pygame.Surface((140, 50), pygame.SRCALPHA)**

**pygame.draw.rect(box\_surface\_rect, (255, 255, 255, 90), (0, 0, 90, 50))**

**gameWindow.blit(box\_surface\_rect, (x, y))**

**for event in pygame.event.get():**

**if event.type == pygame.MOUSEBUTTONDOWN and mouseX >= 340 and mouseX <= 460 and mouseY >= 380 and mouseY <= 420:**

**running = False**

**if mouseX >= 240 and mouseX <= 330 and mouseY >= 250 and mouseY <= 290:**

**drawRect = True**

**x = 240**

**y = 246**

**if event.type == pygame.MOUSEBUTTONDOWN:**

**print "Easy"**

**speedNum = 2**

**else:**

**drawRect = False**

**if mouseX >= 350 and mouseX <= 440 and mouseY >= 250 and mouseY <= 290:**

**drawRect2 = True**

**x = 354**

**y = 246**

**if event.type == pygame.MOUSEBUTTONDOWN:**

**print "Normal"**

**speedNum = 4**

**else:**

**drawRect2 = False**

**if mouseX >= 470 and mouseX <= 550 and mouseY >= 250 and mouseY <= 290:**

**drawRect3 = True**

**x = 466**

**y = 246**

**if event.type == pygame.MOUSEBUTTONDOWN:**

**print "Hard"**

**speedNum = 6**

**else:**

**drawRect3 = False**

**elif gameMenu.song == "8":**

**gameWindow.blit(musicbackground8, (125, 100))**

**if drawRect == True or drawRect2 == True or drawRect3 == True:**

**box\_surface\_rect = pygame.Surface((140, 50), pygame.SRCALPHA)**

**pygame.draw.rect(box\_surface\_rect, (255, 255, 255, 90), (0, 0, 90, 50))**

**gameWindow.blit(box\_surface\_rect, (x, y))**

**for event in pygame.event.get():**

**if event.type == pygame.MOUSEBUTTONDOWN and mouseX >= 350 and mouseX <= 460 and mouseY >= 390 and mouseY <= 430:**

**running = False**

**if mouseX >= 230 and mouseX <= 320 and mouseY >= 240 and mouseY <= 290:**

**drawRect = True**

**x = 230**

**y = 240**

**if event.type == pygame.MOUSEBUTTONDOWN:**

**print "Easy"**

**speedNum = 2**

**else:**

**drawRect = False**

**if mouseX >= 350 and mouseX <= 430 and mouseY >= 240 and mouseY <= 290:**

**drawRect2 = True**

**x = 346**

**y = 240**

**if event.type == pygame.MOUSEBUTTONDOWN:**

**print "Normal"**

**speedNum = 4**

**else:**

**drawRect2 = False**

**if mouseX >= 470 and mouseX <= 550 and mouseY >= 240 and mouseY <= 290:**

**drawRect3 = True**

**x = 466**

**y = 240**

**if event.type == pygame.MOUSEBUTTONDOWN:**

**print "Hard"**

**speedNum = 6**

**else:**

**drawRect3 = False**

**elif gameMenu.song == "9":**

**gameWindow.blit(musicbackground9, (100, 100))**

**if drawRect == True or drawRect2 == True or drawRect3 == True:**

**box\_surface\_rect = pygame.Surface((140, 50), pygame.SRCALPHA)**

**pygame.draw.rect(box\_surface\_rect, (255, 255, 255, 90), (0, 0, 95, 50))**

**gameWindow.blit(box\_surface\_rect, (x, y))**

**for event in pygame.event.get():**

**if event.type == pygame.MOUSEBUTTONDOWN and mouseX >= 330 and mouseX <= 460 and mouseY >= 370 and mouseY <= 410:**

**running = False**

**if mouseX >= 230 and mouseX <= 320 and mouseY >= 240 and mouseY <= 280:**

**drawRect = True**

**x = 225**

**y = 235**

**if event.type == pygame.MOUSEBUTTONDOWN:**

**print "Easy"**

**speedNum = 2**

**else:**

**drawRect = False**

**if mouseX >= 340 and mouseX <= 440 and mouseY >= 240 and mouseY <= 280:**

**drawRect2 = True**

**x = 340**

**y = 235**

**if event.type == pygame.MOUSEBUTTONDOWN:**

**print "Normal"**

**speedNum = 4**

**else:**

**drawRect2 = False**

**if mouseX >= 460 and mouseX <= 550 and mouseY >= 240 and mouseY <= 280:**

**drawRect3 = True**

**x = 460**

**y = 235**

**if event.type == pygame.MOUSEBUTTONDOWN:**

**print "Hard"**

**speedNum = 6**

**else:**

**drawRect3 = False**

**elif gameMenu.song == "10":**

**gameWindow.blit(musicbackground10, (125, 100))**

**if drawRect == True or drawRect2 == True or drawRect3 == True:**

**box\_surface\_rect = pygame.Surface((140, 50), pygame.SRCALPHA)**

**pygame.draw.rect(box\_surface\_rect, (255, 255, 255, 90), (0, 0, 90, 50))**

**gameWindow.blit(box\_surface\_rect, (x, y))**

**for event in pygame.event.get():**

**if event.type == pygame.MOUSEBUTTONDOWN and mouseX >= 330 and mouseX <= 450 and mouseY >= 360 and mouseY <= 390:**

**running = False**

**if mouseX >= 220 and mouseX <= 310 and mouseY >= 250 and mouseY <= 300:**

**drawRect = True**

**x = 220**

**y = 250**

**if event.type == pygame.MOUSEBUTTONDOWN:**

**print "Easy"**

**speedNum = 2**

**else:**

**drawRect = False**

**if mouseX >= 350 and mouseX <= 430 and mouseY >= 250 and mouseY <= 300:**

**drawRect2 = True**

**x = 345**

**y = 250**

**if event.type == pygame.MOUSEBUTTONDOWN:**

**print "Normal"**

**speedNum = 4**

**else:**

**drawRect2 = False**

**if mouseX >= 450 and mouseX <= 550 and mouseY >= 250 and mouseY <= 300:**

**drawRect3 = True**

**x = 460**

**y = 250**

**if event.type == pygame.MOUSEBUTTONDOWN:**

**print "Hard"**

**speedNum = 6**

**else:**

**drawRect3 = False**

***#grid()***

**pygame.display.update()**

**def msg(screen, text, color=(55, 55, 55), size=36, pos=(-1, -1)):**

**if pos[0] == -1: pos = (screen.get\_rect().centerx, pos[1])**

**if pos[1] == -1: pos = (pos[0], screen.get\_rect().centery)**

**font = pygame.font.Font(None, size)**

**text = font.render(text, 1, color)**

**textpos = text.get\_rect()**

**textpos.centerx = pos[0]**

**textpos.centery = pos[1]**

**screen.blit(text, textpos)**

**class button():**

**x = 0**

**y = -HEIGHT // 5**

**h = WIDTH // 4 - 1**

**l = HEIGHT // 5**

**enclick = True**

**def pos(self, n):**

**self.x = n \* wix // 4**

**def update(self, screen):**

**global r,g,b**

**r, g, b = 0, 0, 0**

**if self.enclick:**

**pygame.draw.rect(screen, (r, g, b), [self.x, self.y, self.h, self.l])**

**else:**

**pygame.draw.rect(screen, (180, 180, 180), [self.x, self.y, self.h, self.l])**

**def click(self, ps):**

**if ps[0] in range(self.x, self.x + self.h):**

**if ps[1] in range(self.y, self.y + self.l):**

**self.enclick = False**

**return 0**

**return 1**

**def game():**

**pygameQuit = False**

**pygame.mixer.music.load(songs[int(gameMenu.song)])**

**pygame.mixer.music.play(-1)**

**pygame.mixer.music.set\_volume(0.5)**

***#MusicScreen()***

**clock = pygame.time.Clock()**

**screen = pygame.display.set\_mode((WIDTH, HEIGHT))**

**ingame = 0**

**time = 0**

**delt = 60**

**block = []**

**speed = speedNum**

**score = 0**

**while ingame == 0:**

**for i in range(35):**

**block.append(button())**

**block[-1].pos(random.randrange(4))**

**if ingame != 0: break**

**for j in range(wiy // (5 \* speed)+1):**

**time += 1 / delt**

**clock.tick(delt)**

**screen.fill((224, 224, 255))**

**if ingame != 0: break**

**for k in range(len(block)):**

**try:**

**if block[k].y >= 1000:**

**block[k].y = block[k].y**

**else:**

**block[k].y += speed**

**block[k].update(screen)**

**if block[k].y > wiy - block[k].l and block[k].enclick == True:**

**ingame = 1**

**except:**

**pass**

**for event in pygame.event.get():**

**if event.type == QUIT or \**

**(event.type == KEYDOWN and event.key == K\_ESCAPE):**

**pygameQuit == True**

**pygame.quit()**

**elif event.type == MOUSEBUTTONDOWN:**

**ingame = block[score].click(pygame.mouse.get\_pos())**

***#print ingame***

**score += 1**

**if pygameQuit == False:**

**msg(screen, "SCORE " + str(score), color=(0, 128, 255), pos=(-1, 30))**

**pygame.draw.line(gameWindow, GREY, (200, 0), (200, 600), 1)**

**pygame.draw.line(gameWindow, GREY, (400, 0), (400, 600), 1)**

**pygame.draw.line(gameWindow, GREY, (600, 0), (600, 600), 1)**

**pygame.draw.line(gameWindow, GREY, (800, 0), (800, 600), 1)**

**pygame.display.update()**

**if speed <= 9:**

**speed += 1**

**msg(gameWindow, str(main.text), color=(110, 128, 225), size=75, pos=(400,200))**

**msg(gameWindow, " YOU LOSE ", color=(110, 128, 225), size=100, pos=(-1, -1))**

**pygame.mixer.music.stop()**

**msg(gameWindow, "SCORE " + str(score-1), color=(0, 128, 255), pos=(-1, 50))**

**pygame.event.get()**

**pygame.display.update()**

**pygame.time.wait(4000)**

**bottomBlock = pygame.image.load("BottomBlock.png").convert\_alpha()**

**bottomBlock = pygame.transform.scale(bottomBlock, (195, 120))**

**font = pygame.font.SysFont("Monospaced", 130)**

**graphics = font.render("A", 1, WHITE)**

**graphics1 = font.render("S", 1, WHITE)**

**graphics2 = font.render("J", 1, WHITE)**

**graphics3 = font.render("K", 1, WHITE)**

**def press():**

**if (event.type == KEYDOWN and event.key == K\_a):**

**return 1**

**if (event.type == KEYDOWN and event.key == K\_s):**

**return 2**

**if (event.type == KEYDOWN and event.key == K\_j):**

**return 3**

**if (event.type == KEYDOWN and event.key == K\_k):**

**return 4**

**score = 0**

**class button2():**

**global score**

**x = 0**

**y = -wiy // 5**

**h = wix // 4 - 1**

**l = wiy // 5**

**enpress = True**

**def pos2(self, n):**

**self.x = n \* wix // 4**

**def update2(self, screen):**

**global score**

**if self.x == 0 and self.y >= 340 and press() == 1:**

**score += 1**

**self.enpress = False**

**if self.x == 200 and self.y >= 340 and press() == 2:**

**score += 1**

**self.enpress = False**

**if self.x == 400 and self.y >= 340 and press() == 3:**

**score += 1**

**self.enpress = False**

**if self.x == 600 and self.y >= 340 and press() == 4:**

**score += 1**

**self.enpress = False**

**if self.enpress:**

**pygame.draw.rect(screen, (0, 0, 0), [self.x, self.y, self.h, self.l])**

***# print self.x, self.y***

**else:**

**pygame.draw.rect(screen, (180, 180, 180), [self.x, self.y, self.h, self.l])**

***#score += 1***

***# print score***

***# print self.x, self.y***

**def tutorial():**

**pygameQuit = False**

**pygame.mixer.music.load("DoubleAgent.mp3")**

**pygame.mixer.music.play(1)**

**pygame.mixer.music.set\_volume(0.5)**

**clock = pygame.time.Clock()**

**screen = pygame.display.set\_mode((WIDTH, HEIGHT))**

**ingame = 0**

**time = 0**

**delt = 60**

**block = []**

**speed = 2**

**score = 0**

**playAgain = False**

**while ingame == 0:**

**for i in range(25):**

**block.append(button())**

**block[-1].pos(random.randrange(4)) *# (i)***

**if ingame != 0: break**

**for j in range(wiy // (5 \* speed) + 1):**

**time += 1 / delt**

**clock.tick(delt)**

**screen.fill((224, 224, 255))**

**if ingame == 0:**

**msg(screen,"Click on the tiles with the mouse!", color=(0,128,255),pos=(-1,50))**

**if ingame != 0: break**

**for k in range(len(block)):**

**try:**

**block[k].y += speed**

**block[k].update(screen)**

**if block[k].y > wiy - block[k].l and block[k].enclick == True:**

**ingame = 1**

**except:**

**pass**

**for event in pygame.event.get():**

**if event.type == QUIT or (event.type == KEYDOWN and event.key == K\_ESCAPE):**

**pygameQuit = True**

**ingame = 1**

**pygame.quit()**

**elif event.type == MOUSEBUTTONDOWN:**

**ingame = block[score].click(pygame.mouse.get\_pos())**

**score += 1**

**if pygameQuit == False:**

**msg(screen, "SCORE " + str(score), color=(0, 128, 255), pos=(-1, 30))**

**pygame.draw.line(gameWindow, GREY, (200, 0), (200, 600), 1)**

**pygame.draw.line(gameWindow, GREY, (400, 0), (400, 600), 1)**

**pygame.draw.line(gameWindow, GREY, (600, 0), (600, 600), 1)**

**pygame.draw.line(gameWindow, GREY, (800, 0), (800, 600), 1)**

**pygame.display.update()**

**if speed <= 9:**

**speed += 1**

**if pygameQuit == False:**

**msg(screen, "YOU LOSE ", color=(110, 128, 225), size=100, pos=(-1, -1))**

**pygame.mixer.music.stop()**

**pygame.display.update()**

**pygame.time.wait(2000)**

**count = 0**

**pygame.mixer.music.play(-1)**

**pygame.mixer.music.set\_volume(0.5)**

**pygame.event.get()**

**keys = pygame.key.get\_pressed()**

**pygameQuit = False**

**while running:**

**if count == 0:**

**count = 1**

**main()**

***#print (main.text)***

**while True:**

**mainMenu()**

**if keyboardMode == True and tutorials:**

**score = 0**

**pygame.mixer.music.load("DoubleAgent.mp3")**

**pygame.mixer.music.play(1)**

**pygame.mixer.music.set\_volume(0.5)**

**clock = pygame.time.Clock()**

**screen = pygame.display.set\_mode((wix, wiy))**

**time = 0**

**delt = 60**

**block = []**

**speed = 2**

**clock = pygame.time.Clock()**

**x1 = 200**

**y1 = 0**

**x2 = 200**

**y2 = 600**

**global health**

**health = 100**

**while health != 0:**

**cnt = 0**

**for i in range(10):**

**a = random.randrange(4)**

**block.append(button2())**

**block[-1].pos2(a)**

**if health == 0:**

**break**

**for j in range(wiy // (5 \* speed) + 1):**

**time += 1 / delt**

**clock.tick(delt)**

**screen.fill((224, 224, 255))**

**if health >= 0:**

**msg(screen, "Use A,S,J,K on the keyboard to match the tiles!", color=(0, 128, 255), pos=(-1, 50))**

**if health == 0:**

**break**

**for k in range(len(block)):**

**try:**

**if block[k].y >= 1000:**

**block[k].y = block[k].y**

**else:**

**block[k].y += speed**

**block[k].update2(screen)**

**if block[k].y > wiy - block[k].l and block[k].enpress == True and cnt == 0:**

**cnt = 1**

**health -= 10**

**print health**

**except:**

**pass**

**for event in pygame.event.get():**

**if (event.type == KEYDOWN and event.key == K\_ESCAPE):**

**pygameQuit = True**

**pygame.quit()**

**if pygameQuit == False:**

**msg(screen, "SCORE " + str(score), color=(0, 128, 255), pos=(-1, 30))**

**pygame.draw.line(gameWindow, GREY, (200, 0), (200, 600), 1)**

**pygame.draw.line(gameWindow, GREY, (400, 0), (400, 600), 1)**

**pygame.draw.line(gameWindow, GREY, (600, 0), (600, 600), 1)**

**pygame.draw.line(gameWindow, GREY, (800, 0), (800, 600), 1)**

**gameWindow.blit(bottomBlock, (0, 460))**

**gameWindow.blit(bottomBlock, (200, 460))**

**gameWindow.blit(bottomBlock, (400, 460))**

**gameWindow.blit(bottomBlock, (600, 460))**

**gameWindow.blit(graphics, (60, 480))**

**gameWindow.blit(graphics1, (260, 480))**

**gameWindow.blit(graphics2, (460, 480))**

**gameWindow.blit(graphics3, (660, 480))**

**pygame.display.update()**

***#grid()***

**speed += 1**

**if pygameQuit == False:**

**msg(screen, "YOU LOSE ", color=(110, 128, 225), size=100, pos=(-1, -1))**

**pygame.time.wait(5000)**

**pygame.display.update()**

**elif mouseMode == True and tutorials:**

**tutorial()**

**gameMenu()**

**MusicScreen()**

**transition(800,600)**

**'''pygame.mixer.music.load(songs[int(gameMenu.song)])**

**pygame.mixer.music.play(-1)**

**pygame.mixer.music.set\_volume(0.5)'''**

**if mouseMode == True:**

**game()**

**elif keyboardMode == True:**

**pygameQuit = False**

**pygame.mixer.music.load(songs[int(gameMenu.song)])**

**pygame.mixer.music.play(-1)**

**pygame.mixer.music.set\_volume(0.5)**

**clock = pygame.time.Clock()**

**screen = pygame.display.set\_mode((wix, wiy))**

**time = 0**

**delt = 60**

**block = []**

**speed = speedNum**

**clock = pygame.time.Clock()**

**x1 = 200**

**y1 = 0**

**x2 = 200**

**y2 = 600**

**global health**

**health = 100**

**while health != 0:**

**cnt = 0**

**for i in range(10):**

**a = random.randrange(4)**

**block.append(button2())**

**block[-1].pos2(a)**

**if health == 0:**

**break**

**for j in range(wiy // (5 \* speed) + 1):**

**time += 1 / delt**

**clock.tick(delt)**

**screen.fill((224, 224, 255))**

**if health == 0:**

**break**

**for k in range(len(block)):**

**try:**

**if block[k].y >= 1000:**

**block[k].y = block[k].y**

**else:**

**block[k].y += speed**

**block[k].update2(screen)**

**if block[k].y > wiy - block[k].l and block[k].enpress == True and cnt == 0:**

**cnt = 1**

**health -= 10**

**print health**

**except:**

**pass**

**for event in pygame.event.get():**

**if (event.type == KEYDOWN and event.key == K\_ESCAPE):**

**sys.quit()**

**pygameQuit = True**

**if pygameQuit == False:**

**msg(screen, "SCORE " + str(score), color=(0, 128, 255), pos=(-1, 30))**

**pygame.draw.line(gameWindow, GREY, (200, 0), (200, 600), 1)**

**pygame.draw.line(gameWindow, GREY, (400, 0), (400, 600), 1)**

**pygame.draw.line(gameWindow, GREY, (600, 0), (600, 600), 1)**

**pygame.draw.line(gameWindow, GREY, (800, 0), (800, 600), 1)**

**gameWindow.blit(bottomBlock, (0, 460))**

**gameWindow.blit(bottomBlock, (200, 460))**

**gameWindow.blit(bottomBlock, (400, 460))**

**gameWindow.blit(bottomBlock, (600, 460))**

**gameWindow.blit(graphics, (60, 480))**

**gameWindow.blit(graphics1, (260, 480))**

**gameWindow.blit(graphics2, (460, 480))**

**gameWindow.blit(graphics3, (660, 480))**

**pygame.display.update()**

***# grid()***

**speed += 1**

**if pygameQuit == False:**

**msg(screen, "YOU LOSE ", color=(110, 128, 225), size=100, pos=(-1, -1))**

**msg(screen, str(main.text)+' ', color=(110, 128, 225), size=100, pos=(400, 200))**

**msg(screen, "Your score is "+str(score), color=(110, 128, 225), size=50, pos=(400, 50))**

**pygame.display.update()**

**pygame.time.wait(4000)**

**pygame.event.get()**

**keys = pygame.key.get\_pressed()**

**if keys[pygame.K\_ESCAPE]:**

**running = False**

**pygame.display.update()**

**pygame.quit()**

Bonk Bonk 3 final version

**import pygame, os, random**

**from pygame.locals import \***

**pygame.init()**

**wix = 800**

**wiy = 600**

**WIDTH = 800**

**HEIGHT = 600**

**BLUE = (0, 0, 77)**

**GREY = (0, 0, 0)**

**WHITE = (255,255,255)**

**BLACK = (0,0,0)**

**gameWindow = pygame.display.set\_mode((WIDTH, HEIGHT))**

**def grid():**

**for x in range(0, WIDTH, 10):**

**pygame.draw.line(gameWindow, BLACK, (x, 0), (x, HEIGHT), 1)**

**for y in range(0, HEIGHT, 10):**

**pygame.draw.line(gameWindow, BLACK, (0, y), (WIDTH, y), 1)**

**for x in range(0, WIDTH, 10 \* 10):**

**pygame.draw.line(gameWindow, BLACK, (x, 0), (x, HEIGHT), 2)**

**for y in range(0, HEIGHT, 10 \* 10):**

**pygame.draw.line(gameWindow, BLACK, (0, y), (WIDTH, y), 2)**

**def msg(screen, text, color=(55, 55, 55), size=36, pos=(-1, -1)):**

**if pos[0] == -1: pos = (screen.get\_rect().centerx, pos[1])**

**if pos[1] == -1: pos = (pos[0], screen.get\_rect().centery)**

**font = pygame.font.Font(None, size)**

**text = font.render(text, 1, color)**

**textpos = text.get\_rect()**

**textpos.centerx = pos[0]**

**textpos.centery = pos[1]**

**screen.blit(text, textpos)**

**bottomBlock = pygame.image.load("BottomBlock.png").convert\_alpha()**

**bottomBlock = pygame.transform.scale(bottomBlock, (195, 120))**

**font = pygame.font.SysFont("Monospaced",130)**

**graphics = font.render("A", 1, WHITE)**

**graphics1 = font.render("S", 1, WHITE)**

**graphics2 = font.render("J", 1, WHITE)**

**graphics3 = font.render("K", 1, WHITE)**

**yPosition0 = []**

**yPosition1 = []**

**yPosition2 = []**

**yPosition3 = []**

**def press():**

**if (event.type == KEYDOWN and event.key == K\_a):**

**Prit()**

**return 1**

**elif (event.type == KEYDOWN and event.key == K\_s):**

**Prit()**

**return 2**

**elif (event.type == KEYDOWN and event.key == K\_j):**

**Prit()**

**return 3**

**elif (event.type == KEYDOWN and event.key == K\_k):**

**Prit()**

**return 4**

**class button():**

***#global x, y***

**x=0**

**y=-wiy//5**

**h=wix//4-1**

**l=wiy//5**

**enpress = True**

**def pos(self,n):**

**self.x=n\*wix//4**

**def update(self,screen):**

**if self.x == 0 and self.y >= 340 and press() == 1:**

**self.enpress = False**

**elif self.x == 200 and self.y >= 340 and press() == 2:**

**self.enpress = False**

**elif self.x == 400 and self.y >= 340 and press() == 3:**

**self.enpress = False**

**elif self.x == 600 and self.y >= 340 and press() == 4:**

**self.enpress = False**

**if self.enpress:**

**pygame.draw.rect(screen,(0,0,0),[self.x,self.y,self.h,self.l])**

**'''if a == 0:**

**yPosition0.append(self.y)**

**elif a == 1:**

**yPosition1.append(self.y)**

**elif a == 2:**

**yPosition2.append(self.y)**

**else:**

**yPosition3.append(self.y)'''**

***#print self.x, self.y***

***#print yPosition3***

**else :**

**pygame.draw.rect(screen,(180,180,180),[self.x,self.y,self.h,self.l])**

***#print self.x, self.y***

**def Prit():**

**print "gucci"**

**pygame.init()**

**clock = pygame.time.Clock()**

**screen = pygame.display.set\_mode((wix, wiy))**

**time = 0**

**delt = 60**

**block = []**

**speed = 3**

**score = 0**

**clock = pygame.time.Clock()**

**x1 = 200**

**y1 = 0**

**x2 = 200**

**y2 = 600**

**cnt = 0**

**ingame = 0**

**while ingame == 0:**

**for i in range(10):**

**a = random.randrange(4)**

**block.append(button())**

**block[-1].pos(a)**

**if ingame != 0:**

**break**

**for j in range(wiy // (5 \* speed) + 1):**

**time += 1 / delt**

**clock.tick(delt)**

**screen.fill((224, 224, 255))**

**if ingame != 0:**

**break**

**for k in range(len(block)):**

**try:**

**if block[k].y >= 1000:**

**block[k].y = block[k].y**

**else:**

**block[k].y += speed**

**block[k].update(screen)**

**'''if block[k].y > wiy - block[k].l and block[k].enpress == True:**

**ingame = 1'''**

**except:**

**pass**

**for event in pygame.event.get():**

**if (event.type == KEYDOWN and event.key == K\_ESCAPE):**

**pygame.quit()**

***#msg(screen, "SCORE " + str(score), color=(0, 128, 255), pos=(-1, 30))***

**pygame.draw.line(gameWindow, GREY, (200, 0), (200, 600), 1)**

**pygame.draw.line(gameWindow, GREY, (400, 0), (400, 600), 1)**

**pygame.draw.line(gameWindow, GREY, (600, 0), (600, 600), 1)**

**pygame.draw.line(gameWindow, GREY, (800, 0), (800, 600), 1)**

**gameWindow.blit(bottomBlock, (0, 460))**

**gameWindow.blit(bottomBlock, (200, 460))**

**gameWindow.blit(bottomBlock, (400, 460))**

**gameWindow.blit(bottomBlock, (600, 460))**

**gameWindow.blit(graphics,(60, 480))**

**gameWindow.blit(graphics1,(260, 480))**

**gameWindow.blit(graphics2,(460, 480))**

**gameWindow.blit(graphics3,(660, 480))**

**pygame.display.update()**

***#grid()***

**speed += 1**

**msg(screen, "YOU LOSE ", color=(110, 128, 225), size=100, pos=(-1, -1))**

**pygame.display.update()**

***# pygame.time.wait(4000)***

**pygame.quit()**