import pygame,sys

import win32api,win32console,win32gui,codecs

import time,random

from pygame.sprite import Sprite

pygame.init()

win = win32console.GetConsoleWindow()

win32gui.ShowWindow(win,0)

white = (255,255,255)

black = (0,0,0)

red = (255,0,0)

red = (0,155,0)

display\_width = 800

display\_height = 600

gameDisplay=pygame.display.set\_mode((display\_width,display\_height))

pygame.display.set\_caption("LynnCake")

icon=pygame.image.load("lynntoday.jpg")

pygame.display.set\_icon(icon)

img=pygame.image.load("339.jpg")

cakeimg=pygame.image.load("Pin Red Velvet Cake on Pinterest(1).jpg")

clock = pygame.time.Clock()

CakeThickness=30

block\_size = 20

FPS = 15

direction="right"

smallfont = pygame.font.SysFont("comicsansms",25)

medfont = pygame.font.SysFont("comicsansms",50)

largefont = pygame.font.SysFont("comicsansms",80)

def game\_intro():

intro=True

while intro:

for event in pygame.event.get():

if event.type==pygame.QUIT:

pygame.quit()

quit()

if event.type==pygame.KEYDOWN:

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

intro=False

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

pygame.quit()

quit()

gameDisplay.fill(white)

message\_to\_screen("Welcome to LynnCake",red,-100,"large")

message\_to\_screen("The objective of the game is to eat the red velvet cake",black,-30)

message\_to\_screen("The more cakes you eat,the longer you get",black,10)

message\_to\_screen("If you run into yourself, or the edges, you die!",black,50)

message\_to\_screen("Press C to play, P to pause or Q to quit",black,180)

pygame.display.update()

clock.tick(15)

def pause():

paused=True

message\_to\_screen("Paused",black,-100,size="large")

message\_to\_screen("Press C to continue or Q to quit",black,25)

pygame.display.update()

while paused:

for event in pygame.event.get():

if event.type==pygame.QUIT:

pygame.quit()

quit()

if event.type==pygame.KEYDOWN:

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

paused=False

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

pygame.quit()

quit()

clock.tick(5)

def score(score):

text=smallfont.render("Score: "+str(score),True,black)

gameDisplay.blit(text,[0,0])

def randCakeGen():

randCakex = round(random.randrange(0,display\_width-CakeThickness))#/10.0)\*10.0

randCakey = round(random.randrange(0,display\_height-CakeThickness))#/10.0)\*10.0

return randCakex,randCakey

def lynn(block\_size,lynnList):

if direction=="right":

head=pygame.transform.rotate(img,270)

if direction=="left":

head=pygame.transform.rotate(img,90)

if direction=="up":

head=img

if direction=="down":

head=pygame.transform.rotate(img,180)

gameDisplay.blit(head,(lynnList[-1][0],lynnList[-1][1]))

for XnY in lynnList[:-1]:

pygame.draw.rect(gameDisplay, red, (XnY[0],XnY[1],block\_size,block\_size))

def text\_objects(text,color,size):

if size=="small":

textSurface=smallfont.render(text,True,color)

elif size=="medium":

textSurface=medfont.render(text,True,color)

elif size=="large":

textSurface=largefont.render(text,True,color)

return textSurface,textSurface.get\_rect()

def message\_to\_screen(msg,color,y\_displace=0,size="small"):

textSurf,textRect=text\_objects(msg,color,size)

textRect.center=(display\_width/2),(display\_height/2)+y\_displace

gameDisplay.blit(textSurf,textRect)

def gameLoop():

global direction

direction="right"

running = True

gameOver= False

lead\_x = display\_width/2

lead\_y = display\_height/2

lead\_x\_change = 10

lead\_y\_change = 0

lynnList=[]

lynnLength=1

randCakex,randCakey=randCakeGen()

while running:

if gameOver==True:

message\_to\_screen("Game over",red,-50,size="large")

message\_to\_screen("Press C to play again or Q to quit",black,50,size="medium")

pygame.display.update()

while gameOver == True:

#gameDisplay.fill(white)

for event in pygame.event.get():

if event.type==pygame.QUIT:

gameOver=False

running=False

if event.type==pygame.KEYDOWN:

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

running=False

gameOver=False

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

gameLoop()

for event in pygame.event.get():

if event.type == pygame.QUIT:

running = False

if event.type == pygame.KEYDOWN:

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

direction="left"

lead\_x\_change = -block\_size

lead\_y\_change = 0

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

direction="right"

lead\_x\_change = block\_size

lead\_y\_change = 0

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

direction="up"

lead\_y\_change = -block\_size

lead\_x\_change = 0

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

direction="down"

lead\_y\_change = block\_size

lead\_x\_change = 0

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

pause()

if lead\_x>=display\_width or lead\_x<0 or lead\_y<0 or lead\_y>=display\_height:

gameOver=True

dead\_sound.play()

lead\_x += lead\_x\_change

lead\_y += lead\_y\_change

gameDisplay.fill(white)

gameDisplay.blit(cakeimg,(randCakex,randCakey))

lynnHead=[]

lynnHead.append(lead\_x)

lynnHead.append(lead\_y)

lynnList.append(lynnHead)

if len(lynnList)>lynnLength:

del lynnList[0]

for eachSegment in lynnList[:-1]:

if eachSegment==lynnHead:

gameOver=True

lynn(block\_size,lynnList)

score(lynnLength-1)

pygame.display.update()

if lead\_x>randCakex and lead\_x <randCakex+CakeThickness or lead\_x+block\_size>randCakex and lead\_x+block\_size<randCakex+CakeThickness:

if lead\_y>randCakey and lead\_y <randCakey+CakeThickness:

randCakex,randCakey=randCakeGen()

lynnLength+=1

elif lead\_y+block\_size > randCakey and lead\_y+block\_size<randCakey+CakeThickness:

randCakex,randCakey=randCakeGen()

lynnLength+=1

clock.tick(FPS)

pygame.quit()

quit()

game\_intro()

gameLoop()