import uvage

#GLobal Variables ----------------------------------------------------------------

score = 0

game\_on = False

game\_over = True

game\_over\_screen = True

player\_name = 'z'

scoreboard\_values = []

#Set-up Game Objects ----------------------------------------------------------------

camera = uvage.Camera(800, 800)

scoreboard = uvage.from\_color(400, 400, 'white', 400, 400)

button = uvage.from\_color(600, 600, 'red', 200,200)

score\_table = []

#Inter-Session Progress ----------------------------------------------------------------

def store\_highscore():

global player\_name, score

highscore\_file = open('highscore.txt', 'a')

highscore\_file.write('\n' + player\_name + ',' + *str*(score))

highscore\_file.close()

return

def clear\_highscore():

highscore\_file = open('highscore.txt', 'w')

print(*file* = highscore\_file)

highscore\_file.close()

return

def retrieve\_highscorelist():

highscore\_file = open('highscore.txt', 'r')

highscore = []

for row in highscore\_file.read().split('\n'):

highscore.append(row.split(','))

highscore\_file.close()

highscore.sort(*key*=lambda *x*: *int*(x[1]), *reverse* = True)

return highscore

def create\_score\_table(*values*):

scoreboard\_values = values

rank = 1

entries = 0

while entries < 10 and entries <= (len(scoreboard\_values) - 2):

text = "{} | Score:{}".format(scoreboard\_values[rank][0], scoreboard\_values[rank][1])

entry\_diplay = uvage.from\_text(scoreboard.x, scoreboard.top - 20 + 40 \* rank, text, 40, "black")

score\_table.append(entry\_diplay)

rank += 1

entries += 1

return

# Collectibles --------------------------------

# Snake --------------------------------

# Restart -----------------------------------------

def restart\_game():

return

# Game Code ----------------------------------------------------------------

def game():

global game\_over, game\_on, game\_over\_screen

if game\_over == True:

store\_highscore()

scoreboard\_values = retrieve\_highscorelist()

print(scoreboard\_values)

create\_score\_table(scoreboard\_values)

game\_over\_screen = True

game\_on == False

game\_over = False

if game\_over\_screen == True:

camera.draw(scoreboard)

for row in score\_table:

camera.draw(row)

camera.draw(button)

if button.contains(camera.mouse) and camera.mouseclick == True:

restart\_game()

game\_on = True

game\_over\_screen == False

camera.display()

return

#Run Game ----------------------------------------------------------------

uvage.timer\_loop(30, game)