

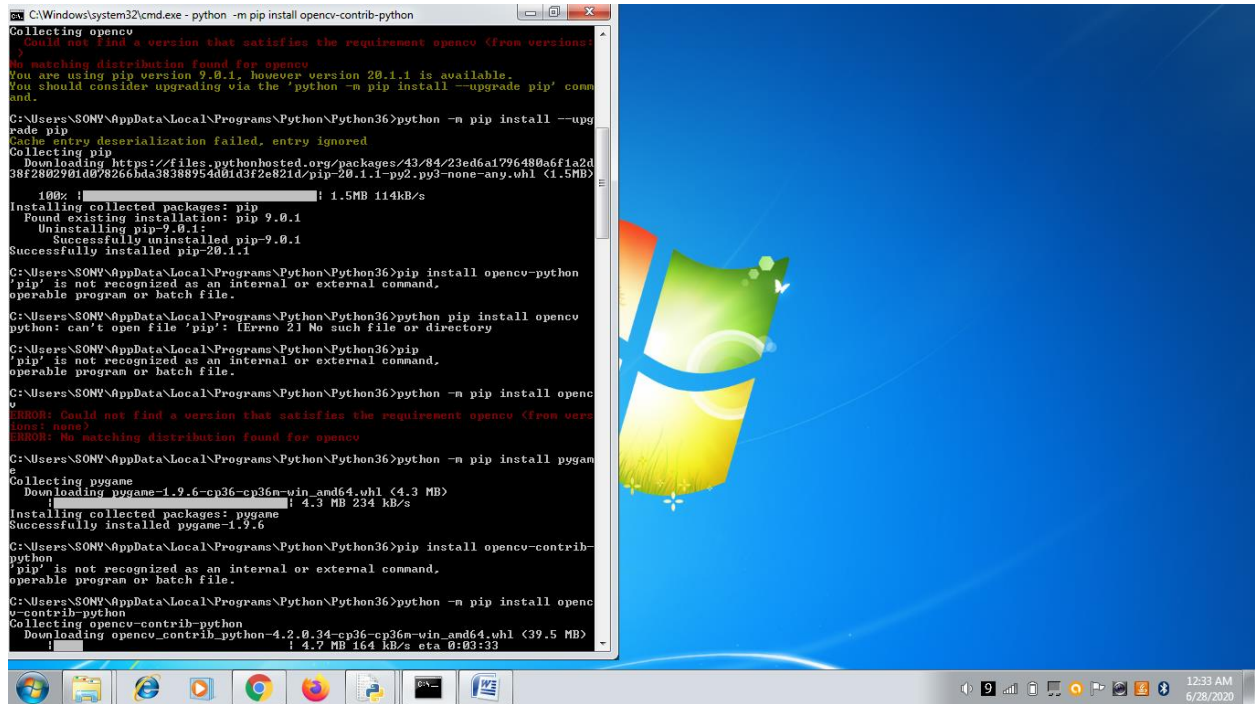
Projects using Python

pip install pygame

(Game Programming)

pip install opencv-contrib-python

(Image Processing)



Pygame –Snake Program

```
import pygame,sys
```

```
import time
```

```
import random
```

```
pygame.init()
```

```
white = (255,255,255)
```

```
black = (100,0,0)
```

```
red = (255,0,0)
```

```
window_width = 800
```

```
window_height = 600
```

```
gameDisplay = pygame.display.set_mode((window_width,window_height))  
pygame.display.set_caption('slither')
```

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

```
FPS = 5
```

```
blockSize = 20
```

```
noPixel = 0
```

```
'''
```

```
sizeGrd = window_width // blockSize
```

```
row = 0
```

```
col = 0
```

```
for nextline in range(sizeGrd):
```

```
'''
```

```
def myquit():
```

```
    ''' Self explanatory '''
```

```
    pygame.quit()
```

```
    sys.exit(0)
```

```
font = pygame.font.SysFont(None, 25, bold=True)
```

```
def drawGrid():
```

```
    sizeGrd = window_width // blockSize
```

```
def snake(blockSize, snakelist):
```

```
    #x = 250 - (segment_width + segment_margin) * i
```

```
    for size in snakelist:
```

```
        pygame.draw.rect(gameDisplay, black,[size[0]+5,size[1],blockSize,blockSize],2)
```

```
def message_to_screen(msg, color):  
    screen_text = font.render(msg, True, color)  
    gameDisplay.blit(screen_text, [window_width/2, window_height/2])  
  
def gameLoop():  
    gameExit = False  
    gameOver = False  
  
    lead_x = window_width/2  
    lead_y = window_height/2  
  
    change_pixels_of_x = 0  
    change_pixels_of_y = 0  
  
    snakelist = []  
    snakeLength = 1  
  
    randomAppleX = round(random.randrange(0, window_width-blockSize)/10.0)*10.0  
    randomAppleY = round(random.randrange(0, window_height-blockSize)/10.0)*10.0  
  
    while not gameExit:  
  
        while gameOver == True:  
            gameDisplay.fill(white)
```

```
message_to_screen("Game over, press c to play again or Q to quit", red)

pygame.display.update()
```

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

    if event.type == pygame.QUIT:

        gameOver = False

        gameExit = True

    if event.type == pygame.KEYDOWN:

        if event.key == pygame.K_q:

            gameExit = True

            gameOver = False

        if event.key == pygame.K_c:

            gameLoop()
```

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

    if event.type == pygame.QUIT:

        gameExit = True

    if event.type == pygame.KEYDOWN:

        if event.key == pygame.K_ESCAPE:

            myquit()

        leftArrow = event.key == pygame.K_LEFT

        rightArrow = event.key == pygame.K_RIGHT

        upArrow = event.key == pygame.K_UP
```

```
        downArrow = event.key == pygame.K_DOWN

    if leftArrow:
        change_pixels_of_x = -blockSize
        change_pixels_of_y = noPixel
    elif rightArrow:
        change_pixels_of_x = blockSize
        change_pixels_of_y = noPixel
    elif upArrow:
        change_pixels_of_y = -blockSize
        change_pixels_of_x = noPixel
    elif downArrow:
        change_pixels_of_y = blockSize
        change_pixels_of_x = noPixel

    if lead_x >= window_width or lead_x < 0 or lead_y >= window_height or lead_y < 0:
        gameOver = True

    lead_x += change_pixels_of_x
    lead_y += change_pixels_of_y

    gameDisplay.fill(white)

    AppleThickness = 20
```

```
print([int(randomAppleX),int(randomAppleY),AppleThickness,AppleThickness])

pygame.draw.rect(gameDisplay, red,
[randomAppleX,randomAppleY,AppleThickness,AppleThickness])
```

```
allspriteslist = []

allspriteslist.append(lead_x)

allspriteslist.append(lead_y)

snakelist.append(allspriteslist)
```

```
if len(snakelist) > snakeLength:

    del snakelist[0]
```

```
for eachSegment in snakelist[:-1]:

    if eachSegment == allspriteslist:

        gameOver = True
```

```
snake(blockSize, snakelist)
```

```
pygame.display.update()
```

```
if lead_x >= randomAppleX and lead_x <= randomAppleX + AppleThickness:

    if lead_y >= randomAppleY and lead_y <= randomAppleY + AppleThickness:

        randomAppleX = round(random.randrange(0, window_width-blockSize)/10.0)*10.0

        randomAppleY = round(random.randrange(0, window_height-blockSize)/10.0)*10.0

        snakeLength += 1
```

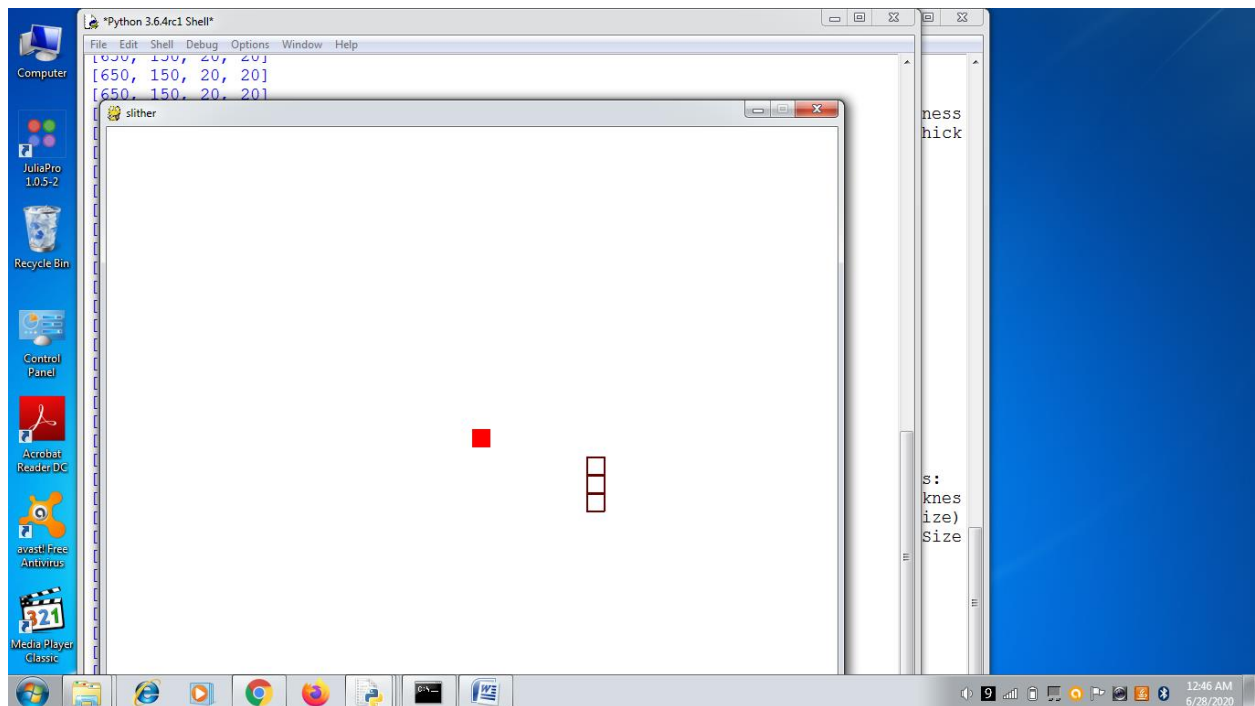
```
clock.tick(FPS)
```

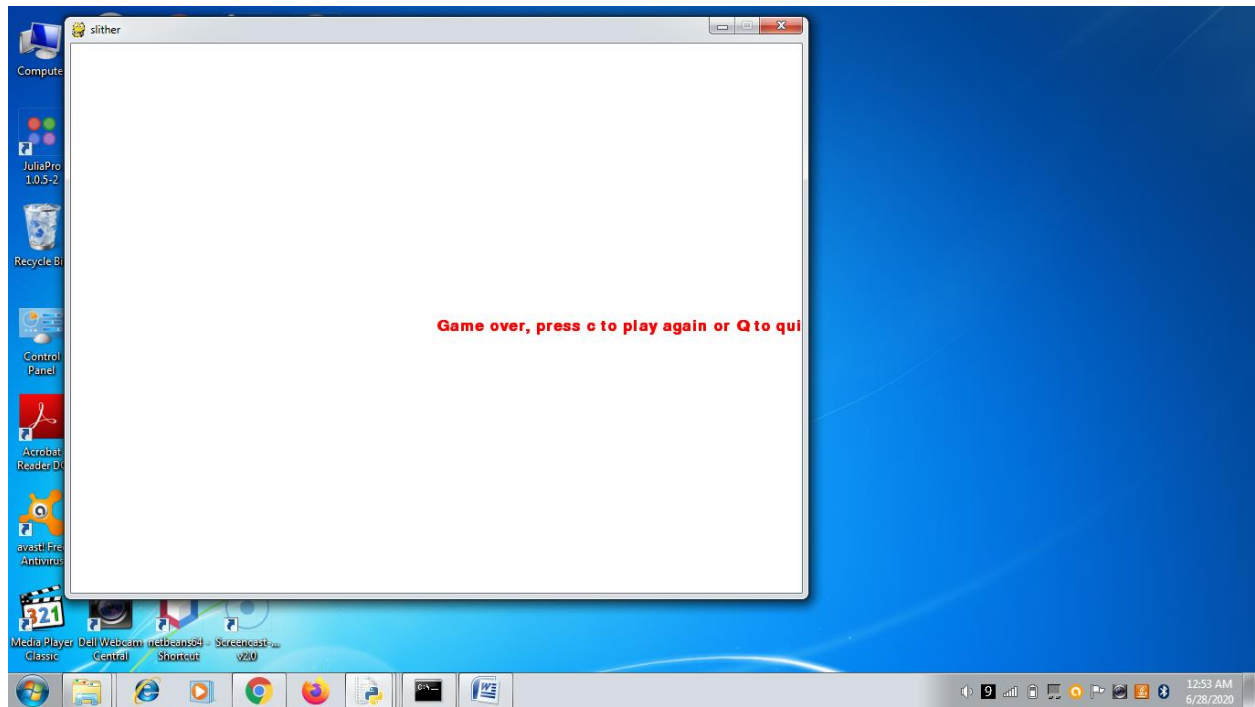
```
pygame.quit()
```

```
quit()
```

```
gameLoop()
```

OUTPUT





Create a VideoCapture object and read from input file

If the input is the camera, pass 0 instead of the video file name

Program

```
import cv2
```

```
# Create a VideoCapture object and read from input file
```

```
# If the input is the camera, pass 0 instead of the video file name
```

```
cap = cv2.VideoCapture("STTP on Python Programming (2020-06-24 at 03_22 GMT-7).mp4")
```

```
# Check if camera opened successfully
```

```
if (cap.isOpened() == False):
```

```
    print("Error opening video stream or file")
```

```
# Read until video is completed
```

```
while(cap.isOpened()):
```


Ms.M.Kamala Malar

AP/IT

```
# Capture frame-by-frame
```

```
ret, frame = cap.read()
```

```
if ret == True:
```

```
    # Display the resulting frame
```

```
    cv2.imshow('Frame',frame)
```

```
    # Press Q on keyboard to exit
```

```
    if cv2.waitKey(25) & 0xFF == ord('q'):
```

```
        break
```

```
# Break the loop
```

```
else:
```

```
    break
```

```
# When everything done, release the video capture object
```

```
cap.release()
```

```
# Closes all the frames
```

```
cv2.destroyAllWindows()
```

OUTPUT

Ms.M.Kamala Malar
AP/IT

