Writing a Video Using OpenCV

In [1]:
```
#Import Libraries
import cv2
import matplotlib.pyplot as plt
%matplotlib inline
```

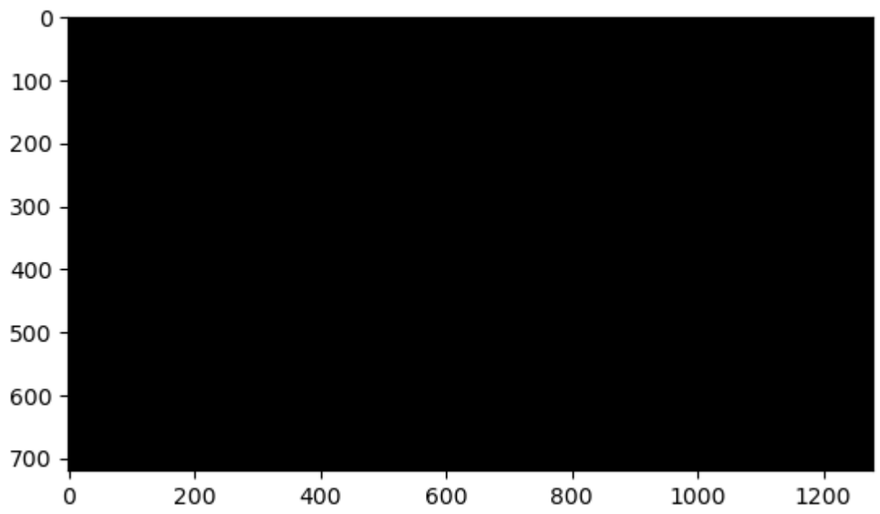
In [2]:
```
#Read a Video from Source
src = "RealRace.mp4" #Source = 0 for Webcam

cap = cv2.VideoCapture(src)

if cap.isOpened() == False:
    print("Error Opening Video Stream of File")

#Read and Display One Frame
ret, frame = cap.read()
plt.imshow(frame[..., ::-1])
```

Out[2]: <matplotlib.image.AxesImage at 0x1f1d53da100>



In [3]:
```
#Display the Video from File
from IPython.display import HTML
HTML(""" <video width = 1024 controls>
<source src = "RealRace.mp4" type = "video/mp4">
</video>""")
```

Out[3]:

0:00

Write Video Using OpenCV

For Writing The Video, We need to Create a Video Writer Object with Right Parameters

In [4]:
```
#Function Syntax --> VideoWriter object  = cv.VideoWriter(filename, fourcc, fps, framesize)
```

This Function 4 Required Parameters

In [5]:
```
#1. filename: Name of The Output Video File
#2. fourcc: 4-Character of codec used to compress the frames.
#Example, VideoWriter::fourcc("P","I","M","1") is MPEG-1 Codec
#VideoWriter::fourcc("P","I","M","1") is motion-jpeg codec. List od codes can be obtained at Video Codecs by FOURCC page.
#FFMPEG backend with MP4 container natively uses other values as fourcc code:
#see ObjectType, so we may recieve a warning message from OpenCVabout fourcc code conversion.

#3. fps: Frame rate of the Created Video Stream
#4. framesize: Size of the Video Frames
```

In [6]:
```
#Default Resolutions of the Frame are Obtained
#Convert the Resolutions from float to Integer
frame_width = int(cap.get(3))
frame_height = int(cap.get(4))

#Define the Codec and Create VideoWriter Object
Out_AVI = cv2.VideoWriter("RealRace.avi", cv2.VideoWriter_fourcc("M","J","P","G"), 10, (frame_width, frame_height))
Out_MP4 = cv2.VideoWriter("RealRace.mp4", cv2.VideoWriter_fourcc(* "XVID"), 10, (frame_width, frame_height))
```

Read Frames and Write to File

In [7]:
```
#Read Untill Video is Completed
while(cap.isOpened()):
    ret, frame = cap.read() #Capture Frame by Frame

    if ret == True:
        #Write The Frames to the Output Files
        Out_AVI.write(frame)
        Out_MP4.write(frame)

    else:
        break

cap.release()
Out_AVI.release()
Out_MP4.release()
```