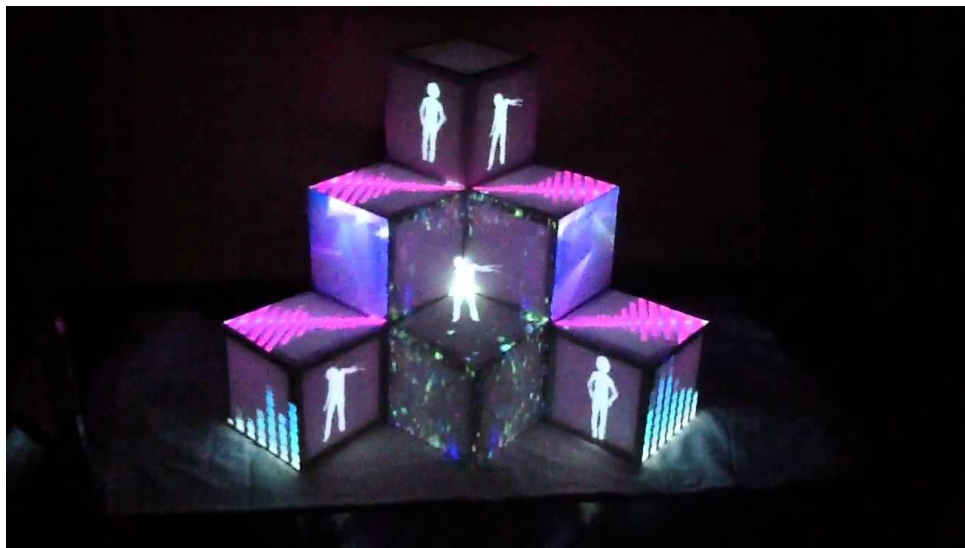


MINI-PROJETO 1: MAPEAMENTO PROJETIVO

ES235 – Aula 11
João Marcelo Teixeira
Williams Costa

OBJETIVO

- Através de mapeamento projetivo, projetar três conteúdos dinâmicos distintos (três vídeos) em três planos não-coplanares no espaço
- Link: <https://goo.gl/KMRuPA>

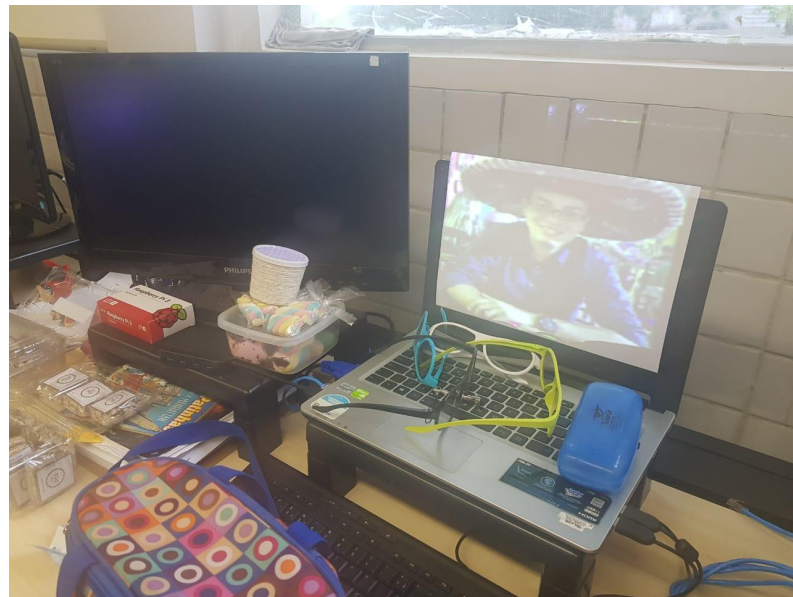
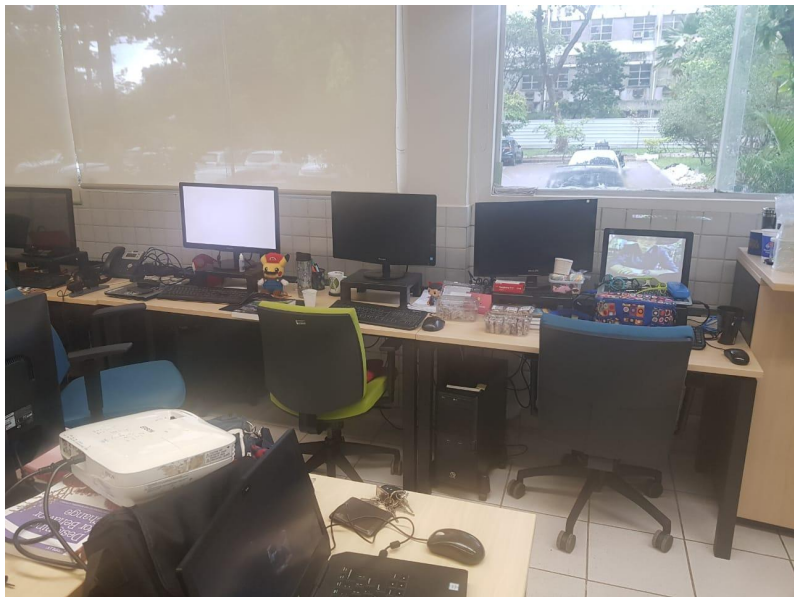


INSTRUÇÕES

- Data de apresentação/entrega: 02/04/2019 (terça-feira)
- Vale 35% da primeira nota

INSTRUÇÕES

- Modifique o código fornecido para:
 - Suportar pelo menos 3 projeções simultâneas
 - Funcionar com arquivos de vídeo (conteúdo dinâmico)



CÓDIGO DE EJEMPLO

```
import math
import numpy as np, cv2

width = 640
height = 480

referencePoints = np.float32([[width/4,height/4],[3*width/4,height/4],[3*width/4,3*height/4],[width/4,3*height/4]])

currentPoint = -1
calibrating = True
fullScreen = False

inputimage1 = cv2.imread("pp.jpg")
rows1, cols1 = inputimage1.shape[:2]
pts1 = np.float32([[0,0],[cols1,0],[cols1,rows1],[0,rows1]])

image = np.zeros((height, width, 3), np.uint8)

def pointColor(n):
    if n == 0:
        return (0,0,255)
    elif n == 1:
        return (0,255,255)
    elif n == 2:
        return (255,255,0)
    else:
        return (0,255,0)
```

CÓDIGO DE EXEMPLO

```
def mouse(event, x, y, flags, param):  
    global currentPoint  
  
    if event == cv2.EVENT_LBUTTONDOWN:  
        cp = 0  
        for point in referencePoints:  
            dist = math.sqrt((x-point[0])*(x-point[0])+(y-point[1])*(y-point[1]))  
            if dist < 4:  
                currentPoint = cp  
                break  
            else:  
                cp = cp + 1  
  
    if event == cv2.EVENT_LBUTTONUP:  
        currentPoint = -1  
  
    if currentPoint != -1:  
        referencePoints[currentPoint] = [x,y]  
  
cv2.namedWindow("test", cv2.WINDOW_NORMAL)  
cv2.setMouseCallback("test", mouse)
```

CÓDIGO DE EXEMPLO

```
while True:
```

```
    image[:] = (0,0,0)
```

```
    if calibrating:
```

```
        color = 0
```

```
        for point in referencePoints:
```

```
            cv2.circle(image, (int(point[0]), int(point[1])), 5, pointColor(color), -1)
```

```
            color = color + 1
```

```
M = cv2.getPerspectiveTransform(pts1, referencePoints)
```

```
cv2.warpPerspective(inputimage1, M, (width,height), image, borderMode=cv2.BORDER_TRANSPARENT)
```

```
cv2.imshow("test", image)
```

```
key = cv2.waitKey(1) & 0xFF
```

```
if key == ord("c"):
```

```
    calibrating = not calibrating
```

```
if key == ord("f"):
```

```
    if fullScreen == False:
```

```
        cv2.setWindowProperty("test", cv2.WND_PROP_FULLSCREEN, cv2.WINDOW_FULLSCREEN)
```

```
    else:
```

```
        cv2.setWindowProperty("test", cv2.WND_PROP_FULLSCREEN, cv2.WINDOW_NORMAL)
```

```
    fullScreen = not fullScreen
```

```
if key == ord("q"):
```

```
    break
```

```
cv2.destroyAllWindows()
```