project_3 (joint)

20171300 지용환

GIT : https://github.com/YONGYONGA/guaZ/tree/three 의 project3폴더

Due to time constraints, I simply added the mod in my homework. Mode 1 is automatic, and mode 2 is the default mode 2 when starting manually.

```python
52    poly = np.array( [[0, 0, 1], [100, 0, 1], [100, 20, 1], [0, 20, 1]])
53    poly = poly.T # 3x4 matrix
54
55    cor = np.array([10, 10, 1])
56    joint1=np.array([90,10,1])
57
58    hand = np.array( [[0, 0, 1], [50, 0, 1], [50, 30, 1], [0, 30, 1]])
59    finger=np.array( [[0, 0, 1], [7, 0, 1], [7, 30, 1], [0, 30, 1]])
60    #hand = np.array( [[0, 0, 1], [100, 0, 1], [100, 20, 1], [0, 20, 1]])
61    hand = hand.T # 3x4 matrix
62    finger=finger.T
63    fin1=np.array([17, 20, 1])
64    fin2=np.array([34, 20, 1])
65
66    degree = 10
67    degree2=10
68    degree3=10
69    degree4=-80
70    mode=2
```

First ploy is shpae of arm. and hand and finger is shpae o finger and hand. Since 3 arms and hand move independently, 4 degrees needed. Finally base mode is 2.

```python
74        for event in pygame.event.get():
75            if event.type == pygame.QUIT:
76                done = True
77            if event.type==pygame.KEYDOWN:
78                if event.key==pygame.K_1:
79                    mode=1
80                elif event.key==pygame.K_2:
81                    mode=2
```

Using keyDown, if push 1, then mode becomes 1 and 2 becomes 2.

```
82 ∨        if(mode==2):
83              keystate = pygame.key.get_pressed()
84 ∨            if keystate[pygame.K_a]:
85                  degree += 1
86 ∨            elif keystate[pygame.K_s]:
87                  degree2+=1
88 ∨            elif keystate[pygame.K_d]:
89                  degree3+=1
90 ∨            elif keystate[pygame.K_f]:
91                  degree4+=1
92 ∨            elif keystate[pygame.K_q]:
93                  degree-=1
94 ∨            elif keystate[pygame.K_w]:
95                  degree2-=1
96 ∨            elif keystate[pygame.K_e]:
97                  degree3-=1
98 ∨            elif keystate[pygame.K_r]:
99                  degree4-=1
```

In manual mode 2, each angle can be adjusted according to keyboard input.

```
100         elif(mode==1):
101             degree+=np.random.randint(-2,2)
102             degree2+=np.random.randint(-2,2)
103             degree3+=np.random.randint(-2,2)
104             degree4+=np.random.randint(-2,2)
105
```

In auto mode 2, each angle can be adjusted accroding to random number.

```
109         text = font.render("1:auto 2:control", True, BLACK)
110         screen.blit(text, [300, 50])
111         text = font.render("NOW MODE : {}".format(mode), True, BLUE)
112         screen.blit(text, [300, 100])
113         text = font1.render("q,w,e,r for -degree", True, BLACK)
114         screen.blit(text, [400, 170])
115         text = font1.render("a,s,d,f for +degree", True, BLACK)
116         screen.blit(text, [400, 200])
```

First blit game information.

```
117         pygame.draw.rect(screen, BLACK, [0,100,80,170], 4)
118         pygame.draw.circle(screen, BLACK,[50,160] , 4)
119         pygame.draw.rect(screen, BLACK, [0,220,40,20], 4)
120         pygame.draw.rect(screen, BLACK, [0,270,170,500], 4)
121         pygame.draw.polygon(screen,BLACK,[[100,690],[130,690],[150,900]
```

And draw robot. Now I have to draw arms, which can move(roate.).

```
127        H = Tmat(100, 400) @ Tmat(10, 10) @ Rmat(degree)
128        pp = H @ poly
129        corp = H @ cor
130        joint11=H@joint1
131        # print(pp.shape, pp, pp.T )
132
133        q = pp[0:2, :].T # N x 2 matrix
134        pygame.draw.polygon(screen, RED, q, 4)
135        pygame.draw.circle(screen, (255, 128, 128), corp[:2], 3)
136        pygame.draw.circle(screen, (255, 128, 128), joint11[:2], 3)
```

Using, Tmat and Rmat, I can draw rotate arm. And since I have to connect second arm to first arm joint11 is needed.

```
140        H = Tmat(joint11[0],joint11[1]) @ Rmat(degree2)@ Tmat(-10, -10)
141        pp = H @ poly
142        corp = H @ cor
143        joint22=H@joint1
144        q = pp[0:2, :].T
145        pygame.draw.polygon(screen, GREEN, q, 4)
146        pygame.draw.circle(screen, (255, 128, 128), corp[:2], 3)
147        pygame.draw.circle(screen, (255, 128, 128), joint22[:2], 3)
```

Using, joinjt11, I can connect second and first arm. Simiraly, use joint 22 I can connect thrid arm to second arm. Moreover I can connect hand, finger, in same way. This can be implemented very easily by just copying 4 sides if you know the conversion matrix, and since there is really no difference, detailed explanation will be omitted.