Catch-up

Report

20190365 김찬주

- Reference: 수업 & https://blog.naver.com/pcy4202/222779569961 (2. Clock에 참고)
- https://github.com/chanjooo/catchup_report_20190365.git

1. twinklingstar

1_twinklingstar_20190365.py

과제 > 1_twinklingstar_20190365.py > ...

```python
#20190365 김찬주
import numpy as np
import cv2

def pos_neg(p0, p1):
    if p1 > p0: arrow = 1
    else: arrow = -1
    return arrow

def getline(x0, y0, x1, y1):
    points = []

    if np.absolute(y1-y0) < np.absolute(x1-x0):
        if x0 == x1:
            for y in range(y0, y1, pos_neg(y0, y1)):
                points.append((x0, y))
        else:
            for x in range(x0, x1, pos_neg(x0, x1)):
                y = (x - x0) * (y1 - y0) / (x1 - x0) + y0
                yint = int(y)
                points.append((x, yint))
```

- range 함수에서 +1씩 이동할지 결정하기 위한 함수.
  - p0 < p1일 때
  - −1씩 이동할지
  - p0 > p1일 때
- getline 함수
  1. |slope| < 1
     (1) $x = \square$ 그래프
     (2) $y = ax + b$ $(-1 < a < 1)$
     - i) a=음수, range(x0, x1, −1)
     - ii) a=양수, range(x0, x1, 1)

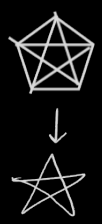
```python
    else:
        if y0 == y1:
            for x in range(x0, x1, pos_neg(x0, x1)):
                points.append((x, y0))
        else:
            for y in range(y0, y1, pos_neg(y0, y1)):
                x = (y - y0) * (x1 - x0) / (y1 - y0) + x0
                xint = int(x)
                points.append((xint, y))

    points.append((x1, y1))
    return points

> def drawLine(canvas, x0, y0, x1, y1, color=(255, 255, 255)): ...

> def deg2rad(deg): ...

> def getRegularNGon(ngon): ...

> def drawLinePQ(canvas, p, q, color): ...

> def drawPolygon(canvas, pts, color, axis=False): ...

> def makeRmat(deg): ...

> def makeTmat(a, b): ...
```

← append(x1, y1)

  2. |slope| > 1
     (1) $y = \square$ 그래프
     (2) $x = cx + d$ $(c > 1$ or $c < -1)$
     - i) c=음수, range(y0, y1, −1)
     - ii) c=양수, range(y0, y1, 1)

```python
def drawStar(canvas, points, color):
    for i in range(points.shape[0]-1):
        r = i+1
        while True:
            drawLine(canvas, points[i][0], points[i][1],
                     points[r][0], points[r][1], color)
            r += 1
            if r == points.shape[0]: break
    drawPolygon(canvas, points, (0,0,0))
    return

def makingStar(canvas, ngon, size, T, R, color):
        #star
        points = getRegularNGon(ngon)
        points *= size
        points[:, 2] = 1
        T1 = makeTmat(T[0], T[1])
        R1 = makeRmat(R)

        pT = T1 @ R1 @ points.T
        points = pT.T
        points = points.astype('int')
        drawStar(canvas, points, color)
```

- star를 그리는 함수
  - 예) 5각형일 때
    - p0 → p1, 2, 3, 4
    - p1 → p2, 3, 4
    - ⋮
    - p3 → p4
  - 테두리 지우기



- ngon, size, Translation, Rotation, color에 맞는 star를 canvas에 그리는 함수

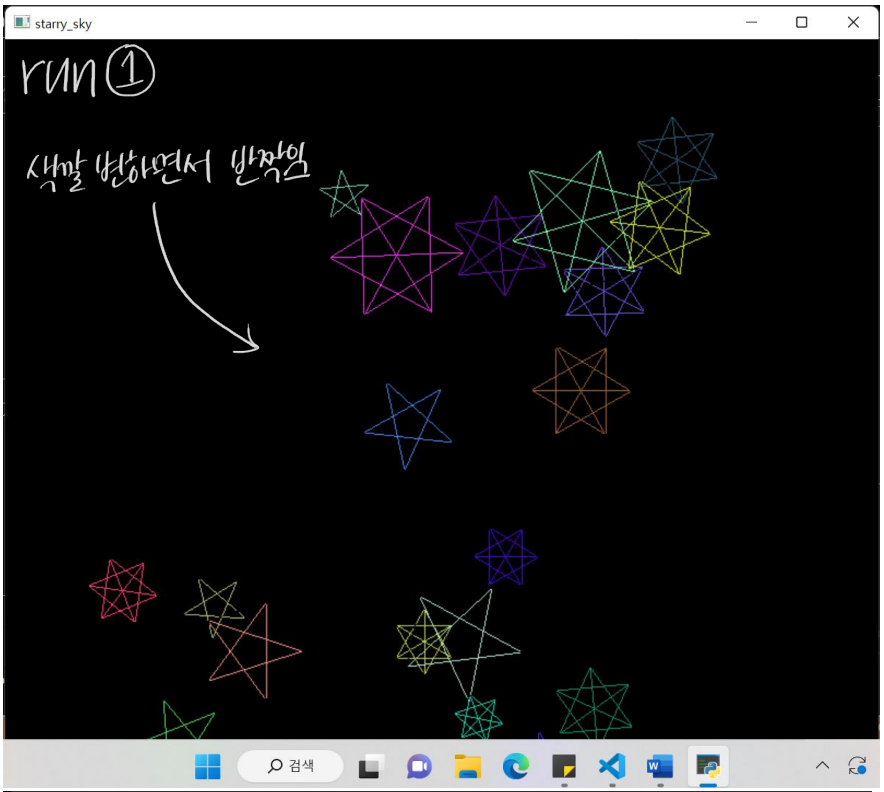
```python
def main():
    width, height = 800, 800
    canvas = np.zeros((height, width, 3), dtype='uint8')

    ngonList = []
    sizeList = []
    locationList = []
    angleList = []

    for i in range(20):
        ngonList.append(np.random.randint(5, 7))
        sizeList.append(np.random.randint(20, 70))
        locationList.append(np.random.randint(100, 700, size=2))
        angleList.append(np.random.randint(0, 360))

    while True:
        colorList = []
        for i in range(20):
            colorList.append(np.random.randint(0, 256, size=3))

        for i in range(20):
            makingStar(canvas, ngonList[i], sizeList[i],
                       locationList[i], angleList[i], colorList[i])

        cv2.imshow("starry_sky", canvas)
        if cv2.waitKey(500) == 27: break
```

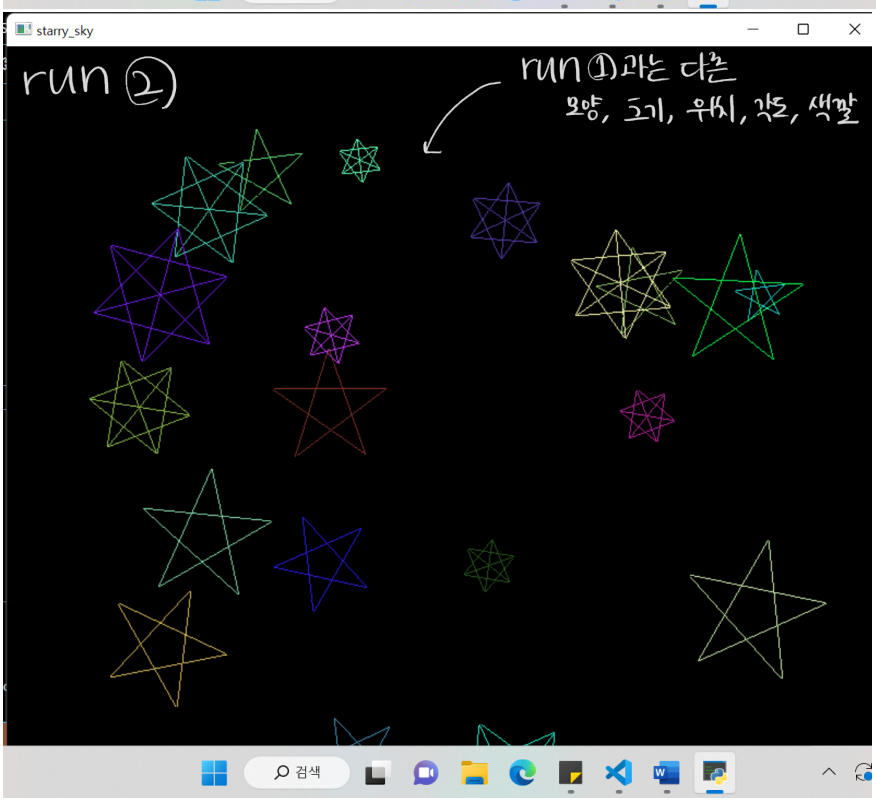
무한 loop

천천히 깜박이도록

- main함수 : 5~7각형 20개를 random하게 그림
  - 빈 리스트 4개 생성
  - 리스트에 20개의 상수 append하기
    - ngon: 5~7각형 생성 위해서 5, 6, 7中 append
    - size: star의 크기 정의 ← 20, 21, ···· 70
    - location: (x좌표, y좌표) ← 100, 101, ··· 700
    - angle: 회전시킬 각도 정의 ← 0, 1, ··· 360
  - colorList 생성 후 BGR 정의
    - 색깔이 변화하면서 반짝이도록 무한 loop에 넣기
  - 리스트의 random 상수 활용해서 star 20개 그리기

```python


if __name__ == "__main__":
    main()
```





2. clock

파일(F) 편집(E) 선택 영역(S) 보기(V) 이동(G) 실행(R) 터미널(T) 도움말(H) 2_clock_20190365.py - MAS2011 - Visual Studio Code

2_clock_20190365.py

과제 > 2_clock_20190365.py > main

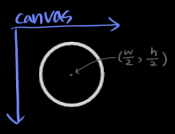
```python
#20190365 김건우
import numpy as np
import cv2
import time

> def pos_neg(p0, p1):...

> def getline(x0, y0, x1, y1):...

> def drawLine(canvas, x0, y0, x1, y1, color=(255, 255, 255)):...

> def deg2rad(deg):...

> def getRegularNGon(ngon):...

> def drawLinePQ(canvas, p, q, color):...

> def drawPolygon(canvas, pts, color, axis=False):...

> def makeRmat(deg):...

> def makeTmat(a, b):...
```

· time 모듈 import

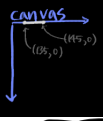
```python
def main():
    width, height = 700, 700
    canvas = np.zeros((height, width, 3), dtype='uint8')

    hour = 0
    hour_angle = 0
    minute_angle = 0

    while True:
        canvas = np.zeros((height, width, 3), dtype='uint8')

        #frame
        frame = getRegularNGon(100)
        frame *= 150
        frame[:,2] = 1
        centerT1 = makeTmat(height/2, width/2)  #center

        fT = centerT1 @ frame.T
        frame = fT.T
        frame = frame.astype('int')
        drawPolygon(canvas, frame, (255,255,255))

        for i in range(0,12):
            gap = np.array([(135,0,1), (145,0,1)])
            startR1 = makeRmat(-90)    #make polygon stand
            gapR2 = makeRmat(i*30)

            gT = centerT1 @ startR1 @ gapR2 @ gap.T
            gap = gT.T
            gap = gap.astype('int')
            drawLinePQ(canvas, gap[0], gap[1], (255,255,255))

        #a hour hand
        hourhand = np.array([(0,-4,1), (60,-4,1), (65,0,1), (60,4,1), (0,4,1)])
        hhR2 = makeRmat(hour_angle)

        hT = centerT1 @ startR1 @ hhR2 @ hourhand.T
        hourhand = hT.T
        hourhand = hourhand.astype('int')
        drawPolygon(canvas, hourhand, (255,80,80))

        #a minute hand
        minutehand = np.array([(0,-5,1), (100,-5,1), (105,0,1), (100,5,1), (0,5,1)])
        mhR2 = makeRmat(minute_angle)

        mT = centerT1 @ startR1 @ mhR2 @ minutehand.T
        minutehand = mT.T
        minutehand = minutehand.astype('int')
        drawPolygon(canvas, minutehand, (255,128,128))

        #time
        now = time.localtime(time.time())

        minute_angle = now.tm_min * 6
        if now.tm_hour > 12: hour = now.tm_hour - 12
        hour_angle = hour*30 + (now.tm_min / 60 * 30)

        #show
        cv2.imshow("window", canvas)
        if cv2.waitKey(10) == 27: break

#main
if __name__ == "__main__":
    main()
```

main함수

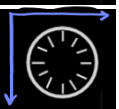
- 변수 정의
- canvas 정의
- frame (시계 틀) 정의
  ↳ 원 모양 (반지름 150)
  ↳ canvas 중앙에 모두 translate
  ↳ 흰색으로 canvas에 그리기



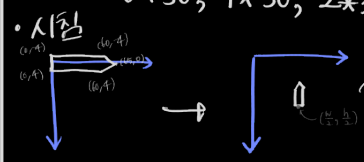
• 시계의 1~12를 직선으로 표시
  - 길이 10의 직선



  - -90° 돌려서 세로로 세우기
  - 30°씩 돌려가며 12개 그리기
    ↳ 0*30, 1*30, 2*30, … 11*30



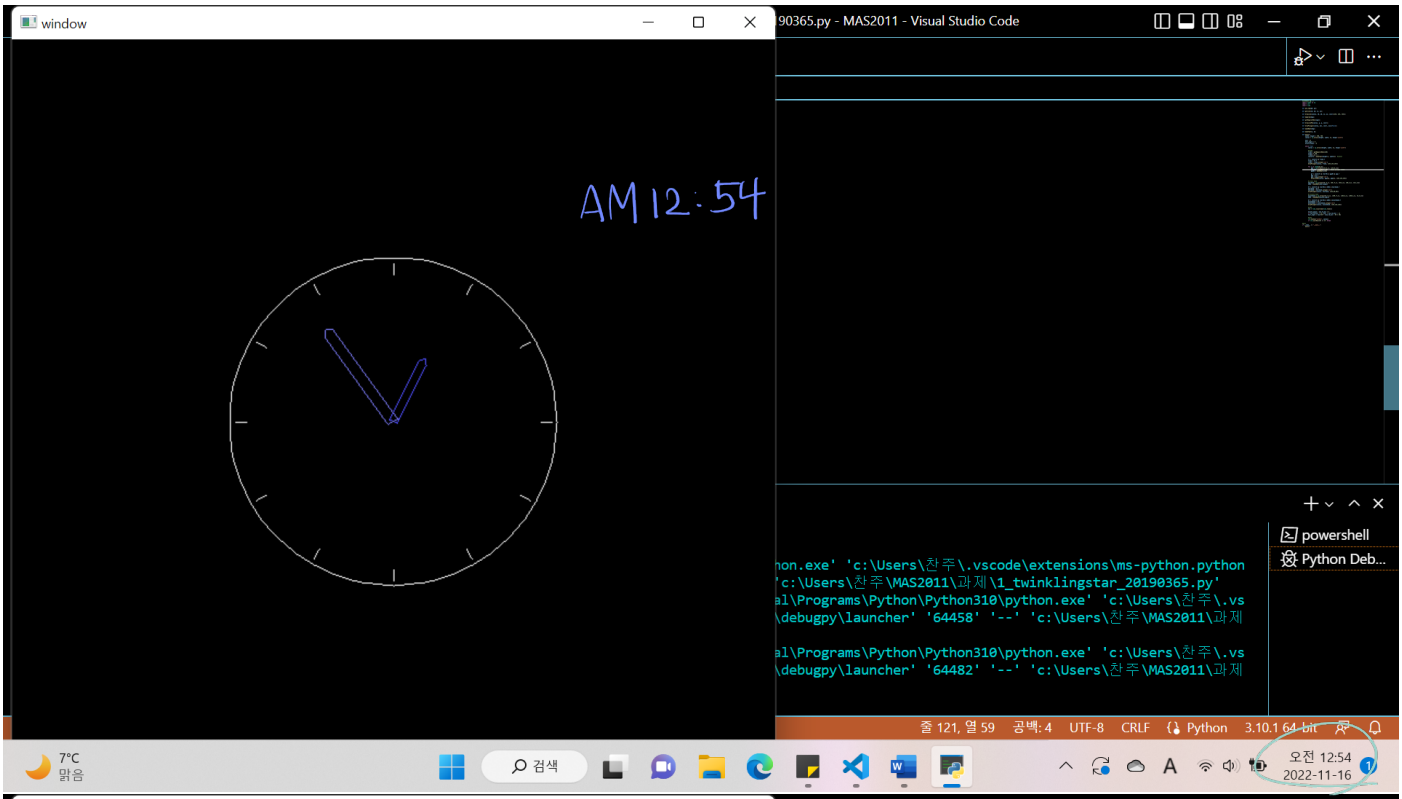
• 시침

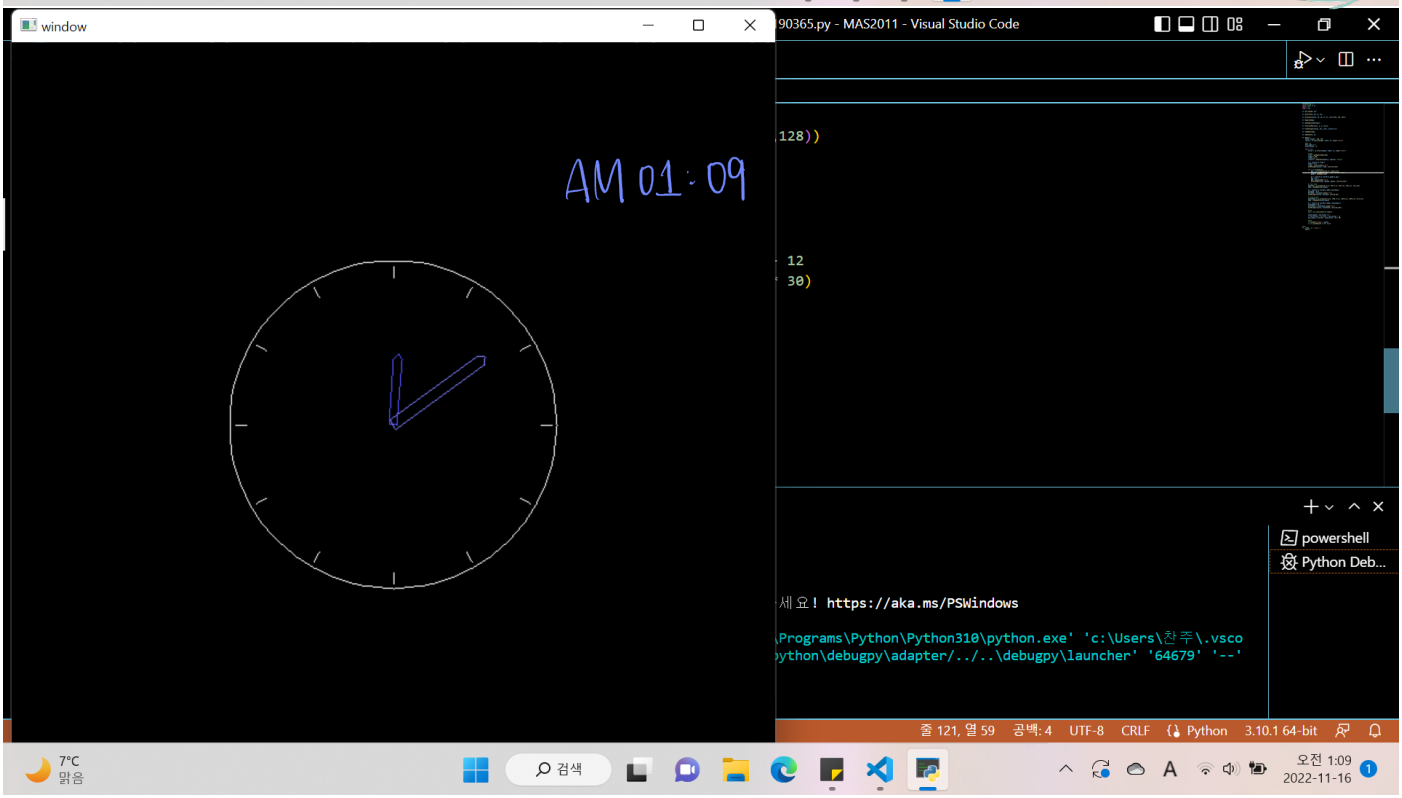


중앙으로 옮기고
-90° 돌림
(시계의 12부터 시작)

• 분침
  - 시침보다 길게 그리기
  - 같은 방식으로 T & R

- 분침의 각도: 60분을 360°로 표현
  ↳ ∴ now.tm_min * 6
- 시침의 각도
  ↳ 13~24시는 1~12로 바꾸기 (-12)
  ↳ 12시를 360°로 표현 ∴ hour * 30 (30°씩)
  ↳ 분이 흘러가는 만큼 시침도 이동 (1분당 0.5°씩 이동)


window
AM 12:54
90365.py - MAS2011 - Visual Studio Code
hon.exe' 'c:\Users\찬주\.vscode\extensions\ms-python.python
'c:\Users\찬주\MAS2011\과제\1_twinklingstar_20190365.py'
al\Programs\Python\Python310\python.exe' 'c:\Users\찬주\.vs
\debugpy\launcher' '64458' '--' 'c:\Users\찬주\MAS2011\과제
al\Programs\Python\Python310\python.exe' 'c:\Users\찬주\.vs
\debugpy\launcher' '64482' '--' 'c:\Users\찬주\MAS2011\과제
powershell
Python Deb...
줄 121, 열 59 공백: 4 UTF-8 CRLF Python 3.10.1 64-bit
7°C
맑음
검색
오전 12:54
2022-11-16


window
AM 01:09
90365.py - MAS2011 - Visual Studio Code
128))
12
30)
세요! https://aka.ms/PSWindows
Programs\Python\Python310\python.exe' 'c:\Users\찬주\.vsco
ython\debugpy\adapter/../..\debugpy\launcher' '64679' '--'
powershell
Python Deb...
줄 121, 열 59 공백: 4 UTF-8 CRLF Python 3.10.1 64-bit
7°C
맑음
검색
오전 1:09
2022-11-16

3. windmill

파일(F) 편집(E) 선택 영역(S) 보기(V) 이동(G) 실행(R) 터미널(T) 도움말(H) 3_windmill_20190365.py - MAS2011 - Visual Studio Code

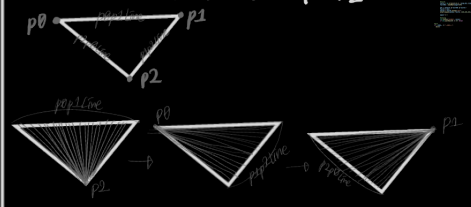
3_windmill_20190365.py

과제 > 3_windmill_20190365.py > ...

```python
#20190365 김찬주
import numpy as np
import cv2

> def pos_neg(p0, p1):...

> def getline(x0, y0, x1, y1):...

> def drawLine(canvas, x0, y0, x1, y1, color=(255, 255, 255)):...

> def deg2rad(deg):...

> def getRegularNGon(ngon):...

> def drawLinePQ(canvas, p, q, color):...

> def drawPolygon(canvas, pts, color, axis=False):...

> def makeRmat(deg):...

> def makeTmat(a, b):...
```

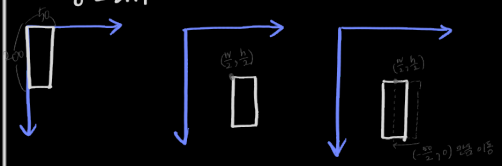
```python
def drawTriangle(canvas, pts, color, fill=False):
    for k in range(pts.shape[0]-1):
        drawLine(canvas, pts[k,0], pts[k,1],
                 pts[k+1,0], pts[k+1,1], color)
    drawLinePQ(canvas, pts[-1], pts[0], color)

    if fill == True:
        p0p1Line = getline(pts[0][0], pts[0][1], pts[1][0], pts[1][1])
        p1p2Line = getline(pts[1][0], pts[1][1], pts[2][0], pts[2][1])
        p2p0Line = getline(pts[2][0], pts[2][1], pts[0][0], pts[0][1])

        for i in p0p1Line:   #p2 ~ line
            drawLine(canvas, pts[2][0], pts[2][1], i[0], i[1], color)
        for i in p1p2Line:   #p0 ~ line
            drawLine(canvas, pts[0][0], pts[0][1], i[0], i[1], color)
        for i in p2p0Line:   #p1 ~ line
            drawLine(canvas, pts[1][0], pts[1][1], i[0], i[1], color)

    return

def main():
    width, height = 800, 800
    canvas = np.zeros((height, width, 3), dtype='uint8')

    angle = 0
```

- 삼각형 그리는 함수 (fill==True면 색칠)
  - 삼각형 테두리 그리기
  - fill==True면 삼각형 내부 색칠하기



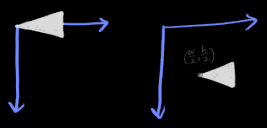

```python
    while True:
        canvas = np.zeros((height, width, 3), dtype='uint8')
        #color = np.random.randint(0, 256, size=3)

        rec_wid = 50
        rec_hei = 200

        wing_wid = 80
        wing_hei = 120

        rec = np.array([(0,0,1), (rec_wid,0,1), (rec_wid,rec_hei,1), (0,rec_hei,1)])
        centerT1 = makeTmat(400,400)
        recT2 = makeTmat(-rec_wid/2,0)

        R = centerT1 @ recT2
        rT = R @ rec.T
        rec = rT.T
        rec = rec.astype('int')
        drawPolygon(canvas, rec, (255,0,0))
```

main 함수
- canvas 정의
- 기둥과 날개의 너비 & 높이 정의
- 기둥 그리기




```python
        #first
        first = np.array([(0,0,1), (wing_hei,-wing_wid/2,1), (wing_hei,wing_wid/2,1)])
        firstR1 = makeRmat(angle)

        pT1 = centerT1 @ firstR1 @ first.T
        first = pT1.T
        first = first.astype('int')
        drawTriangle(canvas, first, (255,255,255), fill=True)

        #second
        second = np.array([(0,0,1), (wing_hei,-wing_wid/2,1), (wing_hei,wing_wid/2,1)])
        secondR1 = makeRmat(angle+90)

        pT2 = centerT1 @ secondR1 @ second.T
        second = pT2.T
        second = second.astype('int')
        drawTriangle(canvas, second, (255,255,255), fill=True)

        #third
        third = np.array([(0,0,1), (wing_hei,-wing_wid/2,1), (wing_hei,wing_wid/2,1)])
        thirdR1 = makeRmat(angle+180)

        pT3 = centerT1 @ thirdR1 @ third.T
        third = pT3.T
        third = third.astype('int')
        drawTriangle(canvas, third, (255,255,255), fill=True)
```

- 첫 번째 날개 그리기



- 두 번째 날개 그리기
  ↳ first 보다 90° 더 돌리기

- 세 번째 날개 그리기
  ↳ first 보다 180° 더 돌리기

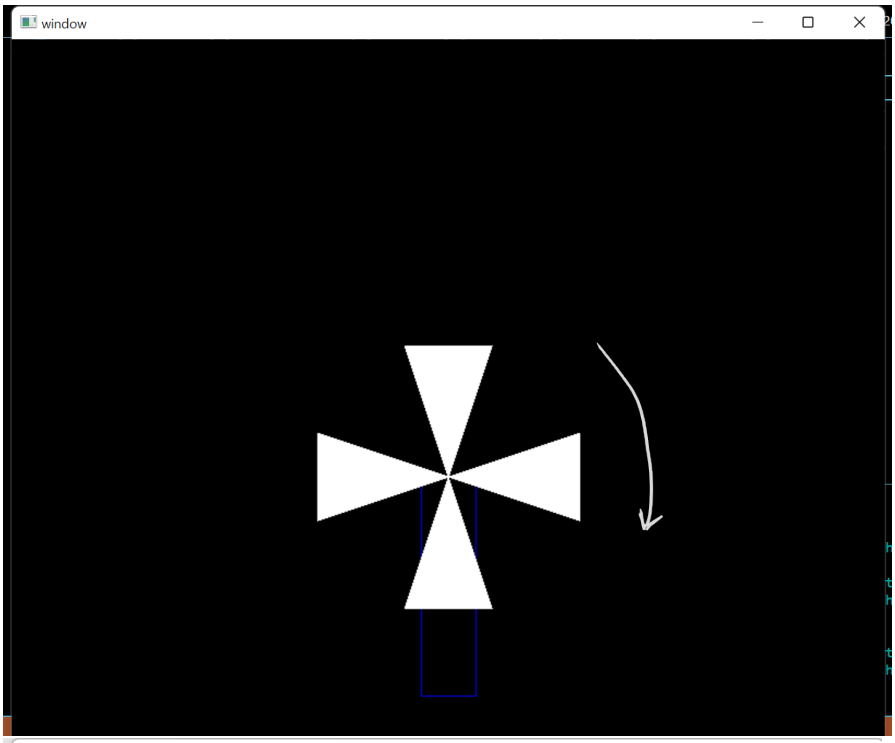
```python
        #fourth
        fourth = np.array([(0,0,1), (wing_hei,-wing_wid/2,1), (wing_hei,wing_wid/2,1)])
        fourthR1 = makeRmat(angle+270)

        pT4 = centerT1 @ fourthR1 @ fourth.T
        fourth = pT4.T
        fourth = fourth.astype('int')
        drawTriangle(canvas, fourth, (255,255,255), fill=True)

        angle += 5
```

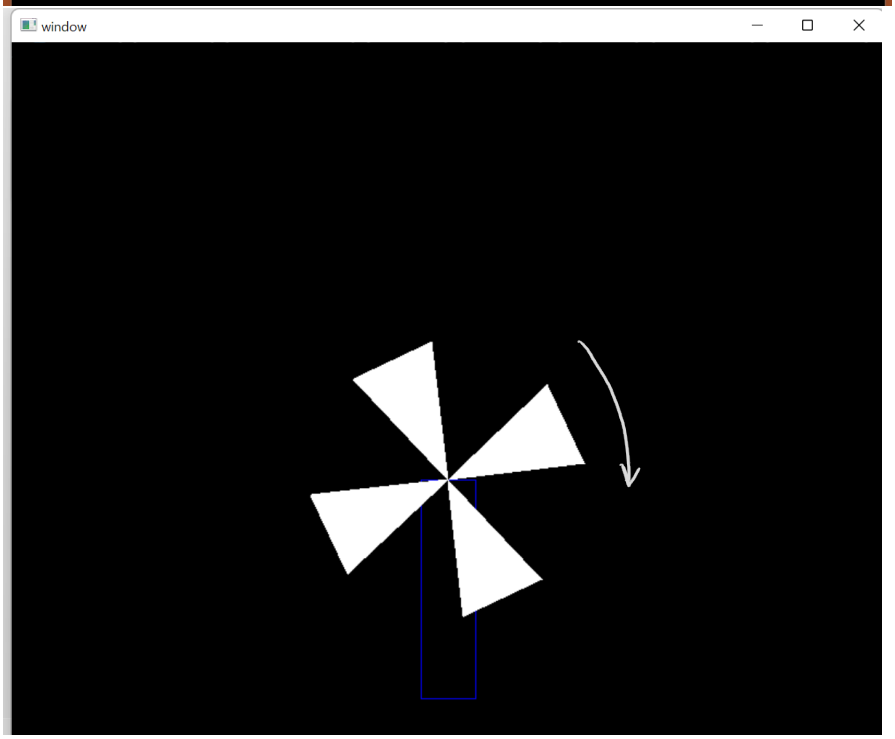
← 5°씩 날개 돌아가도록 angle 정의

```python
        #showing
        cv2.imshow("window", canvas)
        if cv2.waitKey(10) == 27: break

#main
if __name__ == "__main__":
    main()
```

— 네 번째 날개 그리기

↳ first 보다 270° 더 돌리기





4. solar system

파일(F) 편집(E) 선택 영역(S) 보기(V) 이동(G) 실행(R) 터미널(T) 도움말(H) 4_solar_system_20190365.py - MAS2011 - Visual Studio Code

4_solar_system_20190365.py

과제 > 4_solar_system_20190365.py > ...

```python
#20190365 김찬우
import numpy as np
import cv2

> def pos_neg(p0, p1):...

> def getline(x0, y0, x1, y1):...

> def drawLine(canvas, x0, y0, x1, y1, color=(255, 255, 255)):...

> def deg2rad(deg):...

> def getRegularNGon(ngon):...

> def drawLinePQ(canvas, p, q, color):...

> def drawPolygon(canvas, pts, color, axis=False):...

> def makeRmat(deg):...

> def makeTmat(a, b):...
```

```python
def main():
    width, height = 900, 900
    canvas = np.zeros((height, width, 3), dtype='uint8')

    v_revolution = 0
    v_rotation = 0
    e_revolution = 0
    e_rotation = 0
    m_revolution = 0
    m_rotation = 0
    r_revolution = 0

    while True:
        canvas = np.zeros((height, width, 3), dtype='uint8')

        #sun
        sun = getRegularNGon(50)
        sun *= 50
        sun[:, 2] = 1

        sunT1 = makeTmat(height/2, width/2)  #center

        S = sunT1
        sun = (S @ sun.T).T
        sun = sun.astype('int')
        drawPolygon(canvas, sun, (0,0,255))
```

main함수

- canvas 정의
- venus, earth, moon, rocket의 공전 및 자전 각도 변수 생성

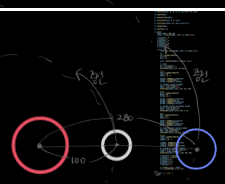
• sun

- 원 모양, 반지름 50
- 캔버스 중앙에 위치

```python
        #venus
        venus = getRegularNGon(20)
        venus *= 20
        venus[:, 2] = 1

        venusR1 = makeRmat(v_revolution)
        venusT1 = makeTmat(100, 0)
        venusR2 = makeRmat(-v_revolution)
        venusR3 = makeRmat(v_rotation)

        V = venusR1 @ venusT1 @ venusR2 @ venusR3
        venus = (S @ V @ venus.T).T
        venus = venus.astype('int')
        drawPolygon(canvas, venus, (255, 255, 255))

        #earth
        earth = getRegularNGon(20)
        earth *= 30
        earth[:, 2] = 1

        earthR1 = makeRmat(e_revolution)
        earthT1 = makeTmat(280, 0)
        earthR2 = makeRmat(-e_revolution)
        earthR3 = makeRmat(e_rotation)

        E = earthR1 @ earthT1 @ earthR2 @ earthR3
        earth = (S @ E @ earth.T).T
        earth = earth.astype('int')
        drawPolygon(canvas, earth, (255, 0, 0))
```

• venus

- 원 모양, 반지름 20
- sun과의 거리 : 100
- sun 주위를 공전하면서 자전
  - ↳ S 적용한 후 V 적용
  - ↳ 공전: R 후에 T / 자전: 공전 각도 뺀 후 자전 각도 R



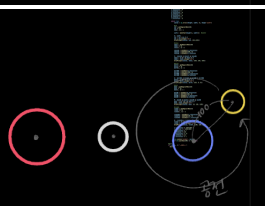
• earth

- 원 모양, 반지름 30
- sun과의 거리 : 280
- sun 주위를 공전하면서 자전
  - ↳ S 적용한 후 E 적용
  - ↳ 공전: R 후에 T / 자전: 공전 각도 뺀 후 자전 각도 R

```python
        #moon
        moon = getRegularNGon(20)
        moon *= 20
        moon[:, 2] = 1

        moonR1 = makeRmat(m_revolution)
        moonT1 = makeTmat(100, 0)
        moonR2 = makeRmat(-m_revolution)
        moonR3 = makeRmat(m_rotation)

        M = moonR1 @ moonT1 @ moonR2 @ moonR3
        moon = (S @ E @ M @ moon.T).T
        moon = moon.astype('int')
        drawPolygon(canvas, moon, (0, 120, 120))
```

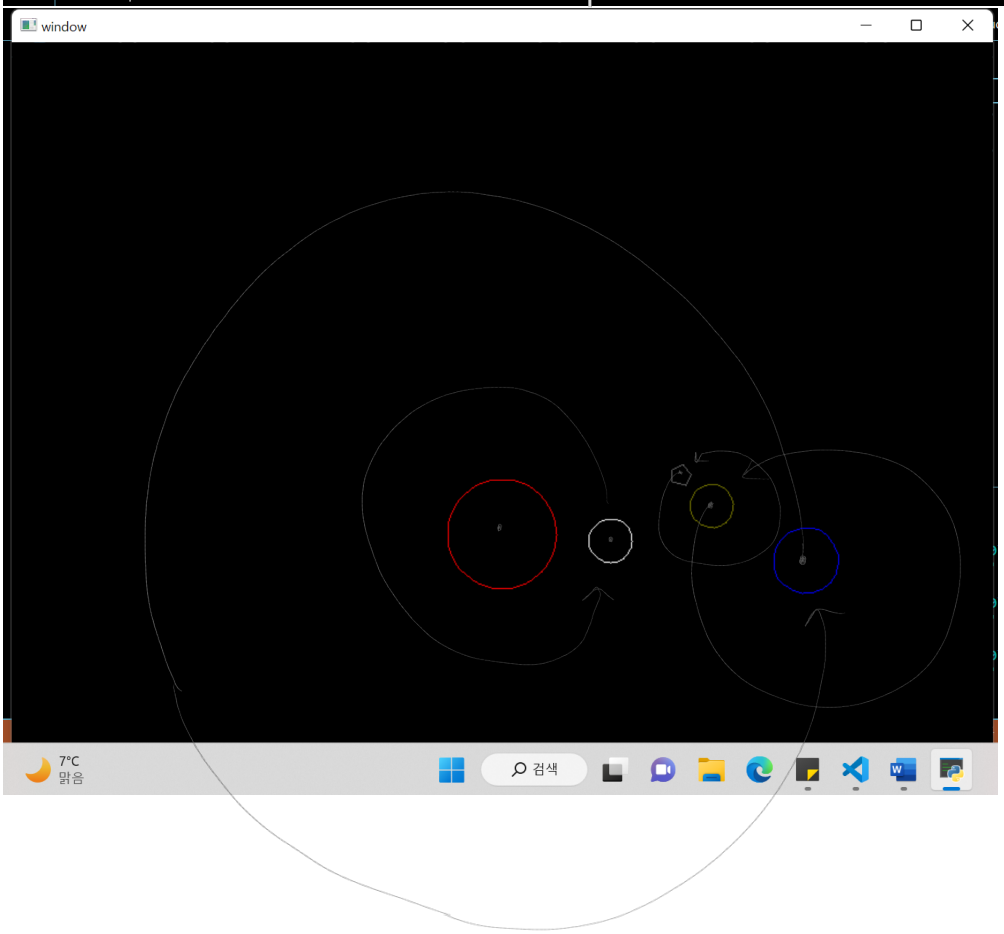
• moon

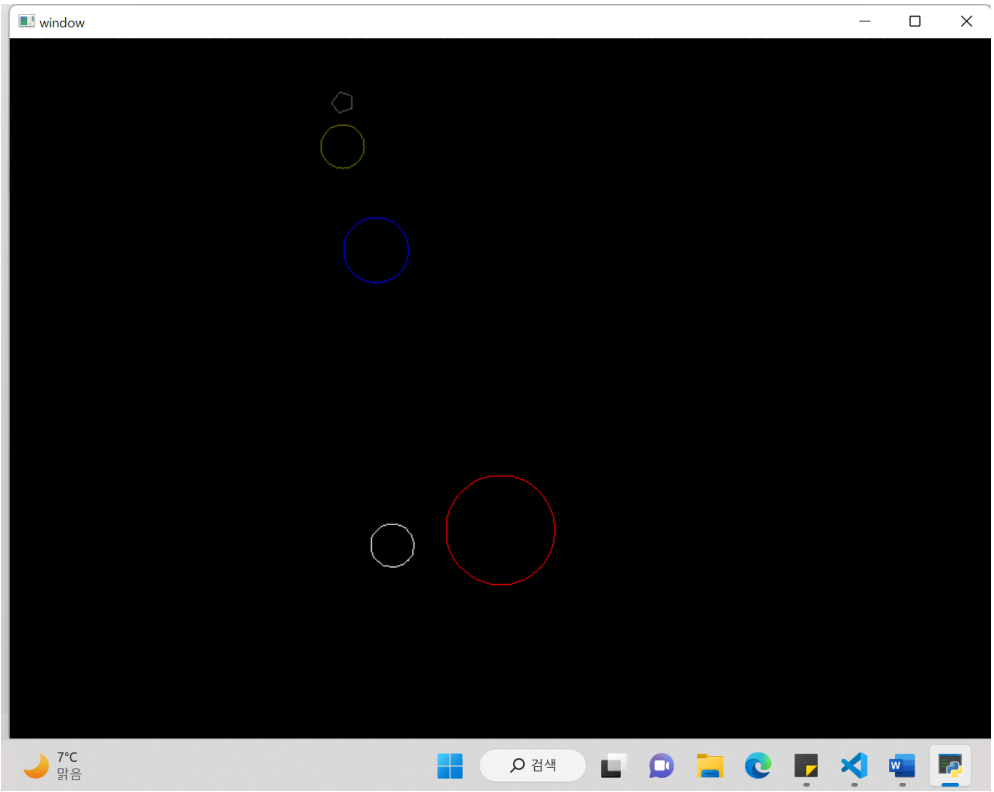
- 원 모양, 반지름 20
- earth와의 거리 : 100
- earth 주위를 공전하면서 자전
  - ↳ S와 E 적용한 후 M 적용
  - ↳ 공전: R 후에 T / 자전: 공전 각도 뺀 후 자전 각도 R

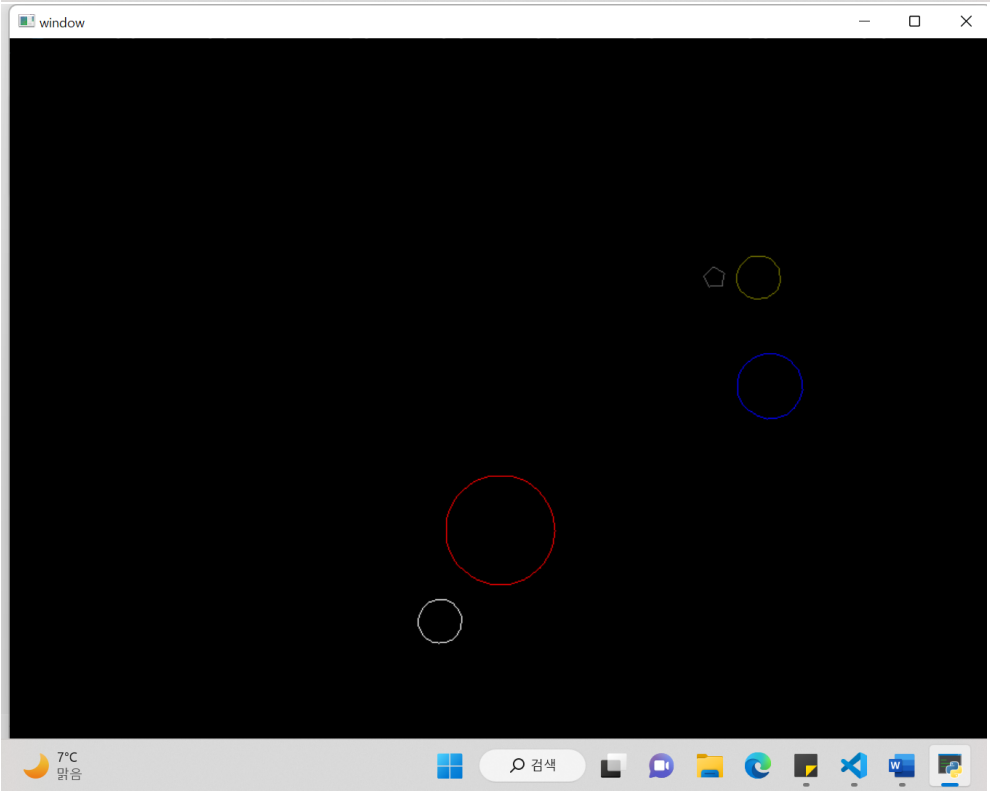


```python
        #rocket
        rocket = getRegularNGon(5)
        rocket *= 10
        rocket[:, 2] = 1

        rocketR1 = makeRmat(r_revolution)
        rocketT1 = makeTmat(40, 0)
        rocketR2 = makeRmat(-r_revolution)

        R = rocketR1 @ rocketT1 @ rocketR2
        rocket = (S @ E @ M @ R @ rocket.T).T
        rocket = rocket.astype('int')
        drawPolygon(canvas, rocket, (100, 100, 100))
    <최전 속도>
        v_revolution += 0.5
        v_rotation += - 0.5   ←금성은 공전 방향과 자전 방향 다름
        e_revolution += 1
        e_rotation += 30
        m_revolution += 12
        m_rotation += 12
        r_revolution += 3

        cv2.imshow("window", canvas)
        if cv2.waitKey(100) == 27: break

#main
if __name__ == "__main__":
    main()
```

- rocket
  - 오각형, 반지름 10
  - moon과의 거리: 40
  - moon 주위 공전 (자전 X)
    - ↳ S와 E와 M 적용한 후 R적용
    - ↳ 공전: R 빼 T / 자전X: 공전 각도 뺀 만큼 R




window
7°C
맑음
검색


window
7°C
맑음
검색

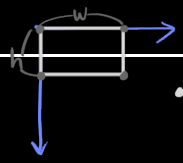
5. robot arm system

파일(F) 편집(E) 선택 영역(S) 보기(V) 이동(G) 실행(R) 터미널(T) 도움말(H) 5_robot_arm_system_20190365.py - MAS2011 - Visual Studio Code

5_robot_arm_system_20190365.py

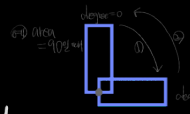
과제 > 5_robot_arm_system_20190365.py > main

```python
#20190365 김찬주
import numpy as np
import cv2

> def pos_neg(p0, p1):...

> def getline(x0, y0, x1, y1):...

> def drawLine(canvas, x0, y0, x1, y1, color=(255, 255, 255)):...

> def deg2rad(deg):...

> def getRegularNGon(ngon):...

> def drawLinePQ(canvas, p, q, color):...

> def drawPolygon(canvas, pts, color, axis=False):...

> def makeRmat(deg):...

> def makeTmat(a, b):...
```



• 직사각형의 꼭짓점을 얻는 함수

```python
def getRectangle(width, height):
    points = [(0,0,1), (width,0,1), (width, height,1), (0,height,1)]
    points = np.array(points)
    return points
```



계속 증가하는 변수

```python
def wiper_moving(angle, area):
    ② if (angle // area) % 2 == 0:
        degree = area-(angle % area)
    ① else:
        degree = angle % area
    return degree
```

• area 안에서 왔다갔다 움직이는 함수

- angle ÷ area = 몫 … 나머지
  - ① 몫이 홀수이면 degree = 나머지 ~ 0, 1, 2 .... area
  - ② 몫이 짝수면 degree = area − 나머지 ~ area, area-1, … 0

```python
def main():
    width, height = 1000, 600
    canvas = np.zeros((height, width, 3), dtype='uint8')
    degree_2 = 0
    degree_3 = 0
    degree_4 = 0
    degree_5 = 0

    alpha = 0
    beta = 0
    gamma = 0
    delta = 0
```

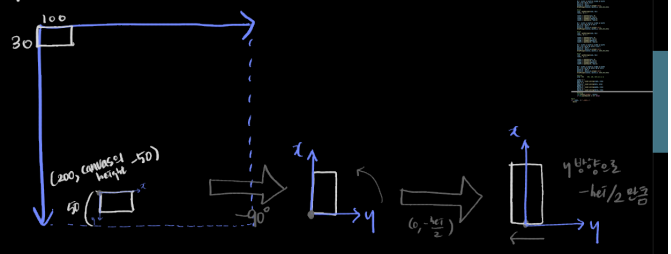
main 함수

- canvas 정의
- 사용할 변수 정의

```python
    wid = 100
    hei = 30

    while True:
        canvas = np.zeros((height, width, 3), dtype='uint8')
        #color = np.random.randint(0, 256, size=3)

        #Rec1
        rec1 = getRectangle(wid, hei)
        rec1[:, 2] = 1

        rec1T1 = makeTmat(200, height-50)
        rec1R1 = makeRmat(-90)
        rec1T2 = makeTmat(0, -hei/2)

        Q1 = rec1T1 @ rec1R1 @ rec1T2
        qT1 = Q1 @ rec1.T
        points_1 = qT1.T
        points_1 = points_1.astype('int')
        drawPolygon(canvas, points_1, (255,155,155))

        #rec2
        rec2 = getRectangle(wid, hei)
        rec2[:, 2] = 1
```

- 직사각형의 width와 height

• Rec 1



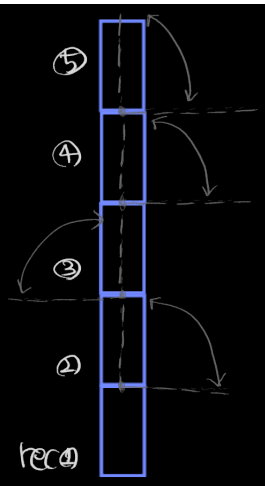
• Rec 2

```python
        rec2T1 = makeTmat(wid, 0)
        rec2T2 = makeTmat(0, hei/2)
        rec2R1 = makeRmat(degree_2)
        rec2T3 = makeTmat(0, -hei/2)

        Q2 = rec2T1 @ rec2T2 @ rec2R1 @ rec2T3
        qT2 = Q1 @ Q2 @ rec2.T
        points_2 = qT2.T
        points_2 = points_2.astype('int')
        drawPolygon(canvas, points_2, (255,155,155))

        #rec3
        rec3 = getRectangle(wid, hei)
        rec3[:, 2] = 1

        rec3T1 = makeTmat(wid, 0)
        rec3T2 = makeTmat(0, hei/2)
        rec3R1 = makeRmat(-degree_3)  ← 반대 방향
        rec3T3 = makeTmat(0, -hei/2)
```

- Rec 1 기준 x방향으로 wid 만큼 move
  hei/2 만큼 move
  degree_2 만큼 rotate
  - hei/2 만큼 move back
- Rec 1에서 시작해서
  추가로 T, R

⇓ 반복

```python
        Q3 = rec3T1 @ rec3T2 @ rec3R1 @ rec3T3
        qT3 = Q1 @ Q2 @ Q3 @ rec3.T
        points_3 = qT3.T
        points_3 = points_3.astype('int')
        drawPolygon(canvas, points_3, (255,155,155))

        #rec4
        rec4 = getRectagle(wid, hei)
        rec4[:, 2] = 1

        rec4T1 = makeTmat(wid, 0)
        rec4T2 = makeTmat(0, hei/2)
        rec4R1 = makeRmat(degree_4)
        rec4T3 = makeTmat(0, -hei/2)

        Q4 = rec4T1 @ rec4T2 @ rec4R1 @ rec4T3
        qT4 = Q1 @ Q2 @ Q3 @ Q4 @ rec4.T
        points_4 = qT4.T
        points_4 = points_4.astype('int')
        drawPolygon(canvas, points_4, (255,155,155))

        #rec5
        rec5 = getRectagle(wid, hei)
        rec5[:, 2] = 1
```




```python
        rec5T1 = makeTmat(wid, 0)
        rec5T2 = makeTmat(0, hei/2)
        rec5R1 = makeRmat(degree_5)
        rec5T3 = makeTmat(0, -hei/2)

        Q5 = rec5T1 @ rec5T2 @ rec5R1 @ rec5T3
        qT5 = Q1 @ Q2 @ Q3 @ Q4 @ Q5 @ rec5.T
        points_5 = qT5.T
        points_5 = points_5.astype('int')
        drawPolygon(canvas, points_5, (255,155,155))

        #moving
        area = 90     #90, 180, 360 and so on
```

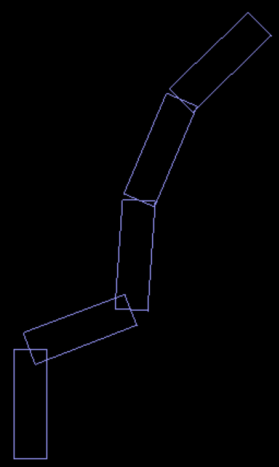
← 90° 안에서 움직이도록

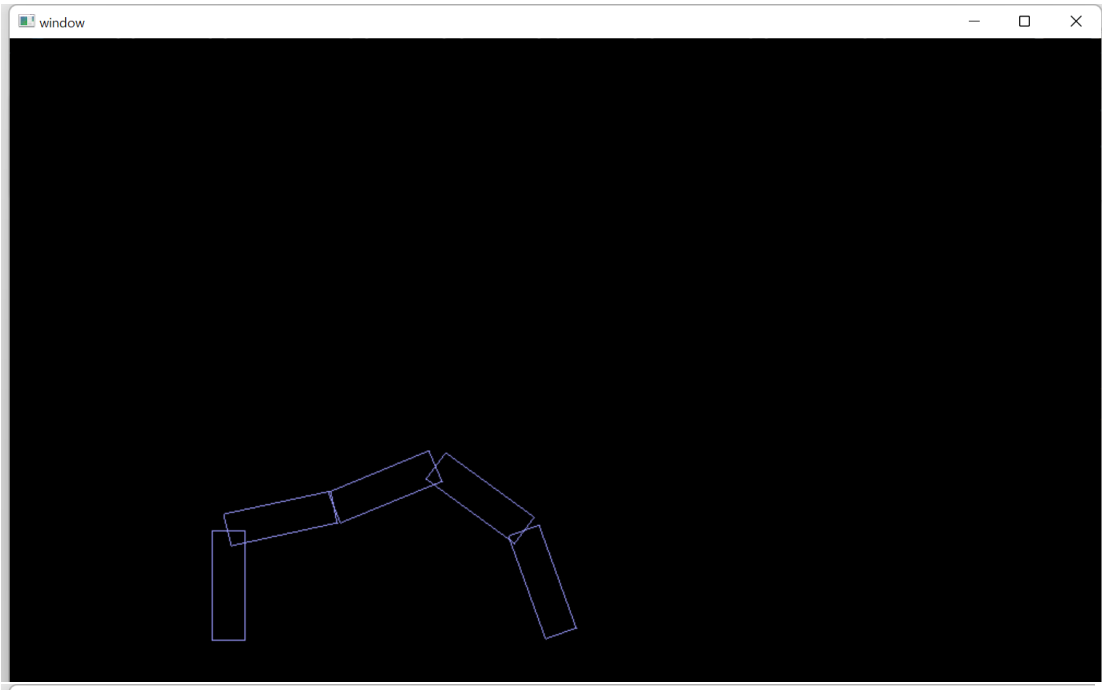
```python
        alpha += 3
        degree_2 = wiper_moving(alpha, area)
        beta += 5
        degree_3 = wiper_moving(beta, area)
        gamma += 7
        degree_4 = wiper_moving(gamma, area)
        delta += 1
        degree_5 = wiper_moving(delta, area)
```

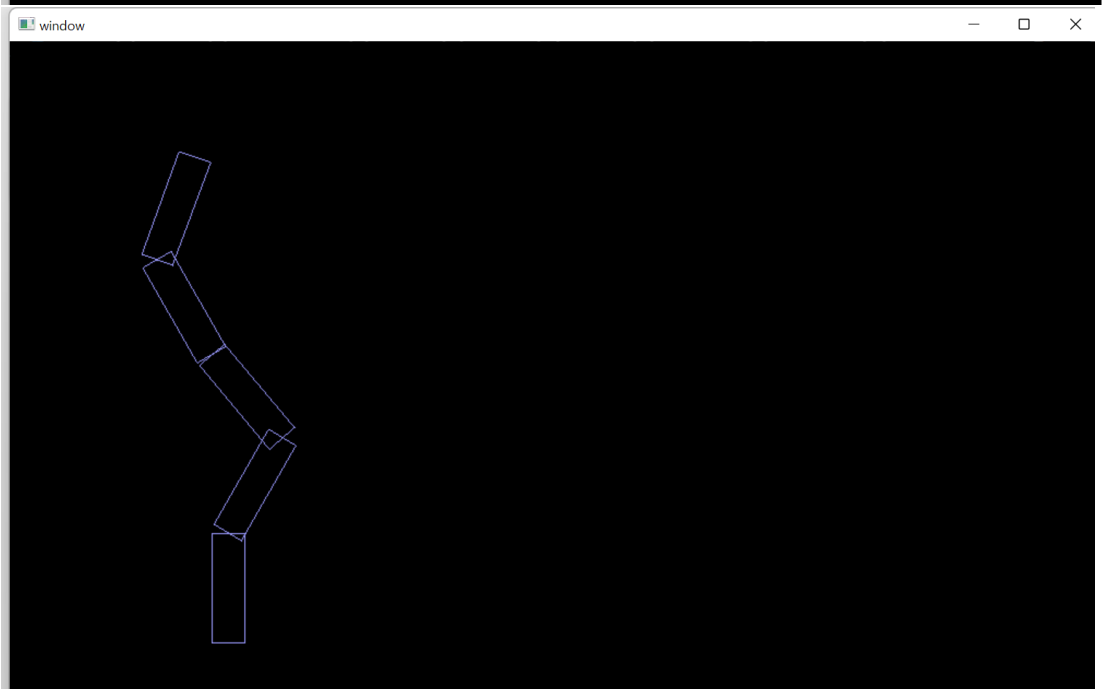
정도 차이 有

```python
        #showing
        cv2.imshow("window", canvas)
        if cv2.waitKey(10) == 27: break

#main
if __name__ == "__main__":
    main()
```

window




window


window