

MODISCAN E1

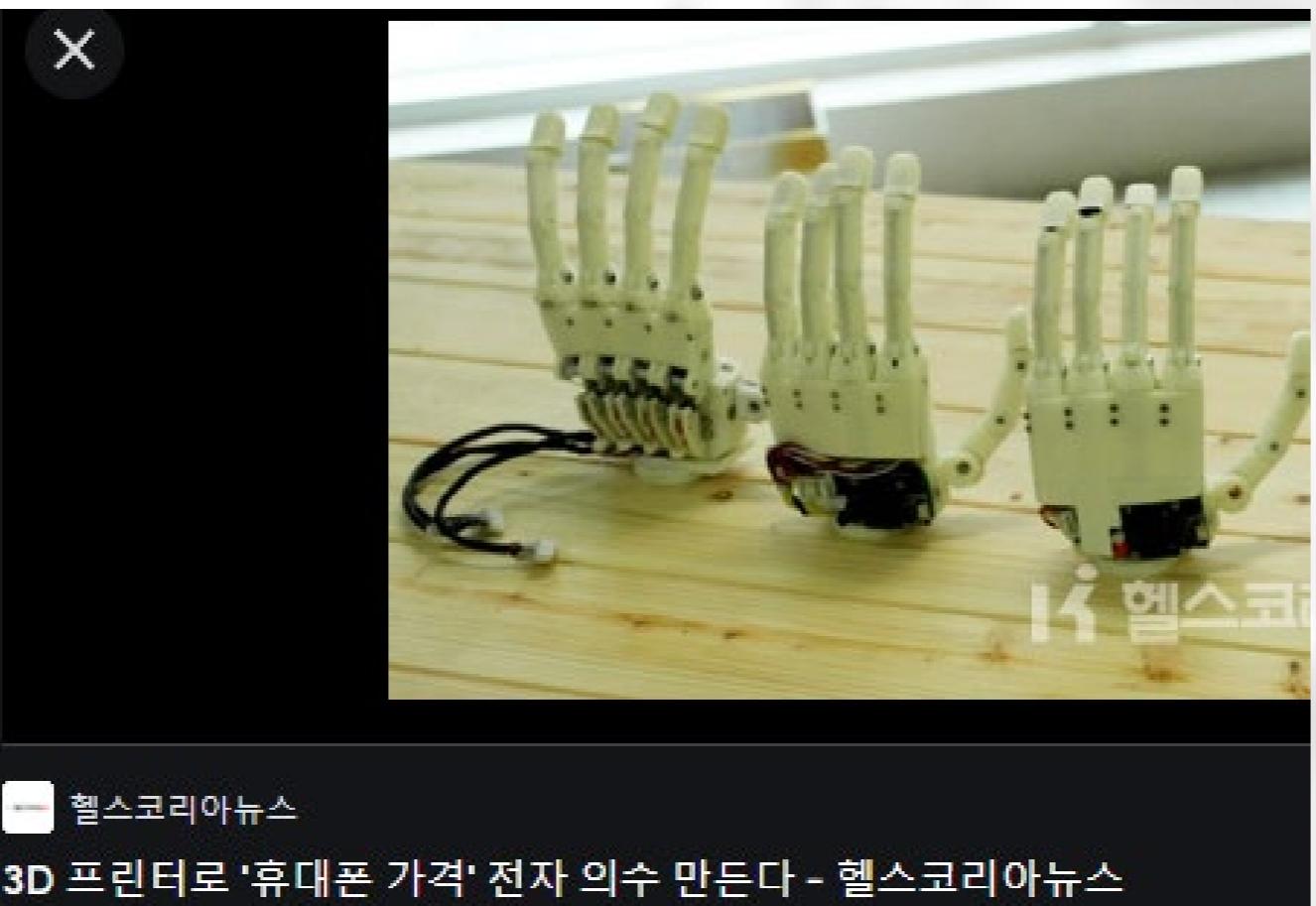
보급형 (교육용) 접촉식 3D 스캐너



프로젝트 기획배경

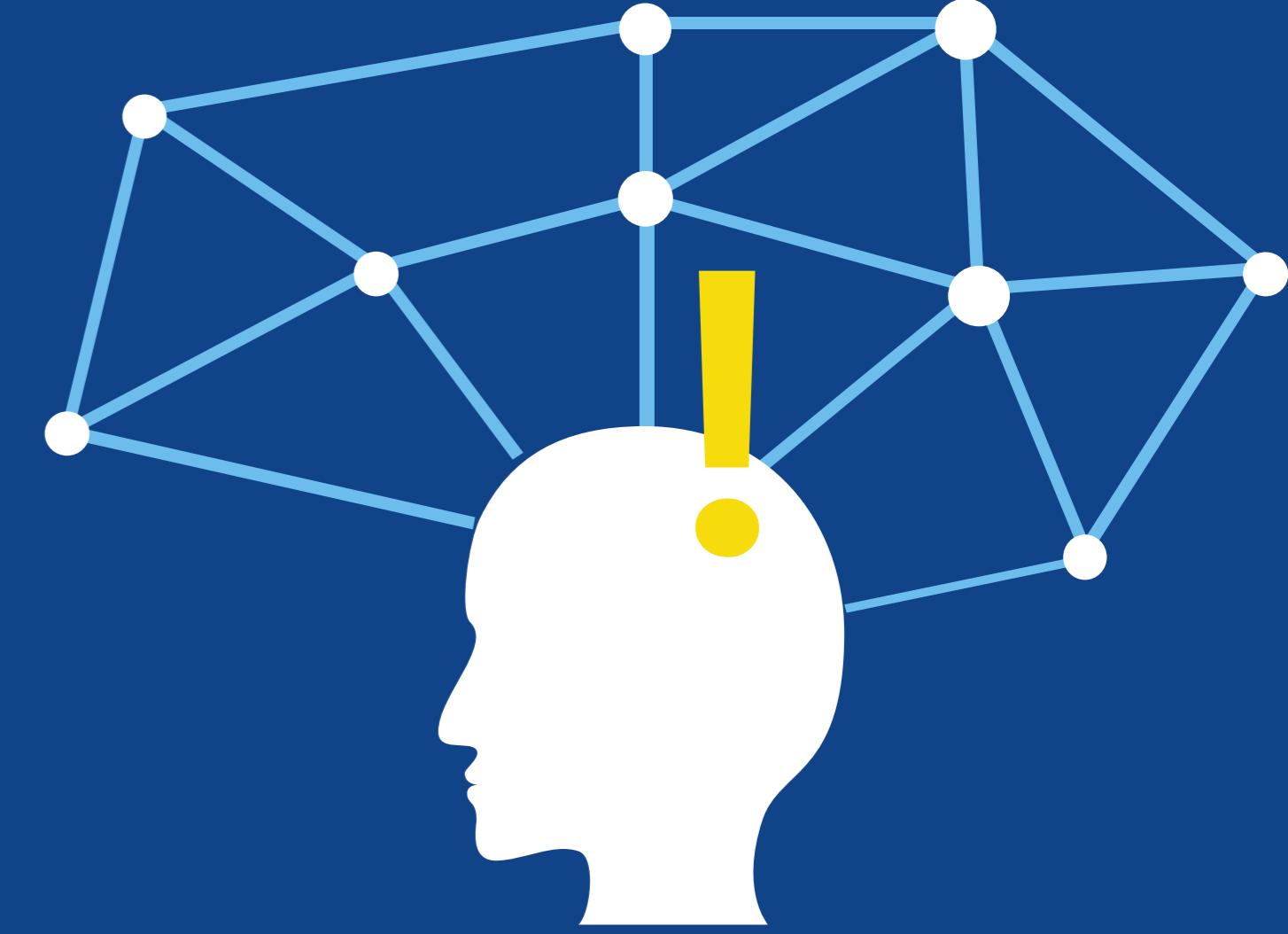
1

3D 프린팅



Core77

The Nuo 3D Mask: 3D-Printed, Custom-Fit to Your Face, Washable, Reusable, N95-Compliant -



3D 프린팅 3D 스캐닝

합리적인 가격대



손도리닷컴 교육용 DIY 3D프린터 피코 PICO

99,000원

디지털/가전 > 주변기기 > 프린터 > 3D프린터

노즐 : 0.4mm | 소모품 : PLA | 층간간격 : 0.05~0.2mm | 지원파일 : Gcode, STL | 최대출력물크기 : 10 x 10 x 10cm | 단자 : USB | 지원OS : 윈도, Mac | 인쇄속도 : 10 ~ 40mm/s |

리뷰 407 · 구매건수 80 · 등록일 2021.01. · ❤️ 찜하기 125 · 💬 신고하기



3D 스캐너

2,200,000원

디지털/가전 > 주변기기 > 스캐너 > 3D 스캐너

리뷰 2 · 구매건수 1 · 등록일 2017.12. · ❤️ 찜하기 9 · 💬 신고하기



해외 델타 리니어 플러스 3d프린터 자작 부품

165,000원

디지털/가전 > 주변기기 > 프린터 > 3D프린터

노즐 : 0.4~0.8mm | 소모품 : PLA, ABS, 레이우드, 나일론, HIPS, 우드, 스톤 | 층간간격 : 0.02mm | 내가 써본것만 팝니다 그래서 내써봣!! 안써본건 올리지도 않아용~~

리뷰 88 · 구매건수 5 · 등록일 2020.03. · ❤️ 찜하기 36 · 💬 신고하기



3D스캐너 아인스캔 EinScan H

7,700,000원

디지털/가전 > 주변기기 > 스캐너 > 3D 스캐너

단자 : USB | 스캔센서 : CMOS

리뷰 1 · 구매건수 1 · 등록일 2021.04. · ❤️ 찜하기 1 · 💬 신고하기



가정용·교육용 SMART3D MINI 3D프린터

180,000원

디지털/가전 > 주변기기 > 프린터 > 3D프린터

리뷰 70 · 구매건수 9 · 등록일 2018.11. · ❤️ 찜하기 138 · 💬 신고하기



해외 메타포트 프로2 Matterport Pro2 - 3D 캡쳐 카메라 - 추가금 X

3,550,000원

디지털/가전 > 주변기기 > 스캐너 > 3D 스캐너

리뷰 3 · 구매건수 1 · 등록일 2020.05. · ❤️ 찜하기 2 · 💬 신고하기



미니 완제품 3D프린터 오토레벨링 교육용 개인용 입문용 Makerpi M1

199,000원

디지털/가전 > 주변기기 > 프린터 > 3D프린터

노즐 : 0.4mm | 소모품 : PLA | 층간간격 : 0.05~0.3mm | 지원파일 : Gcode, STL, OBJ | 형태 : 오픈형 | 지원OS : 윈도, Mac | 부가기능 : 사용중소모품교체, 소모품인식, 일시정지,

리뷰 7 · 구매건수 12 · 등록일 2021.09. · ❤️ 찜하기 18 · 💬 신고하기

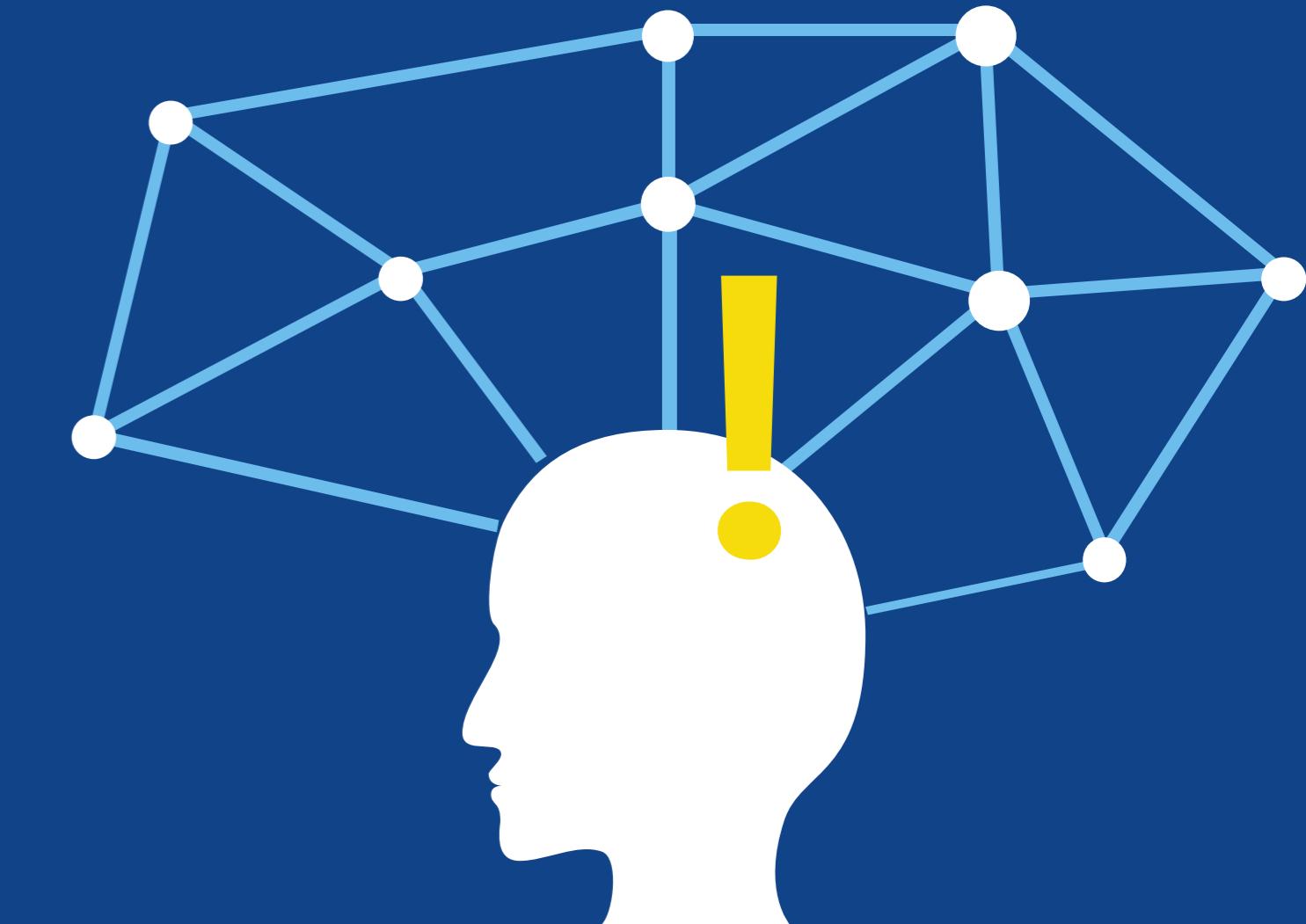


해외 메타포트 프로2 Matterport Pro2 with Tripod Kit - 추가금 X

3,800,000원

디지털/가전 > 주변기기 > 스캐너 > 3D 스캐너

리뷰 1 · 구매건수 1 · 등록일 2021.06. · ❤️ 찜하기 2 · 💬 신고하기

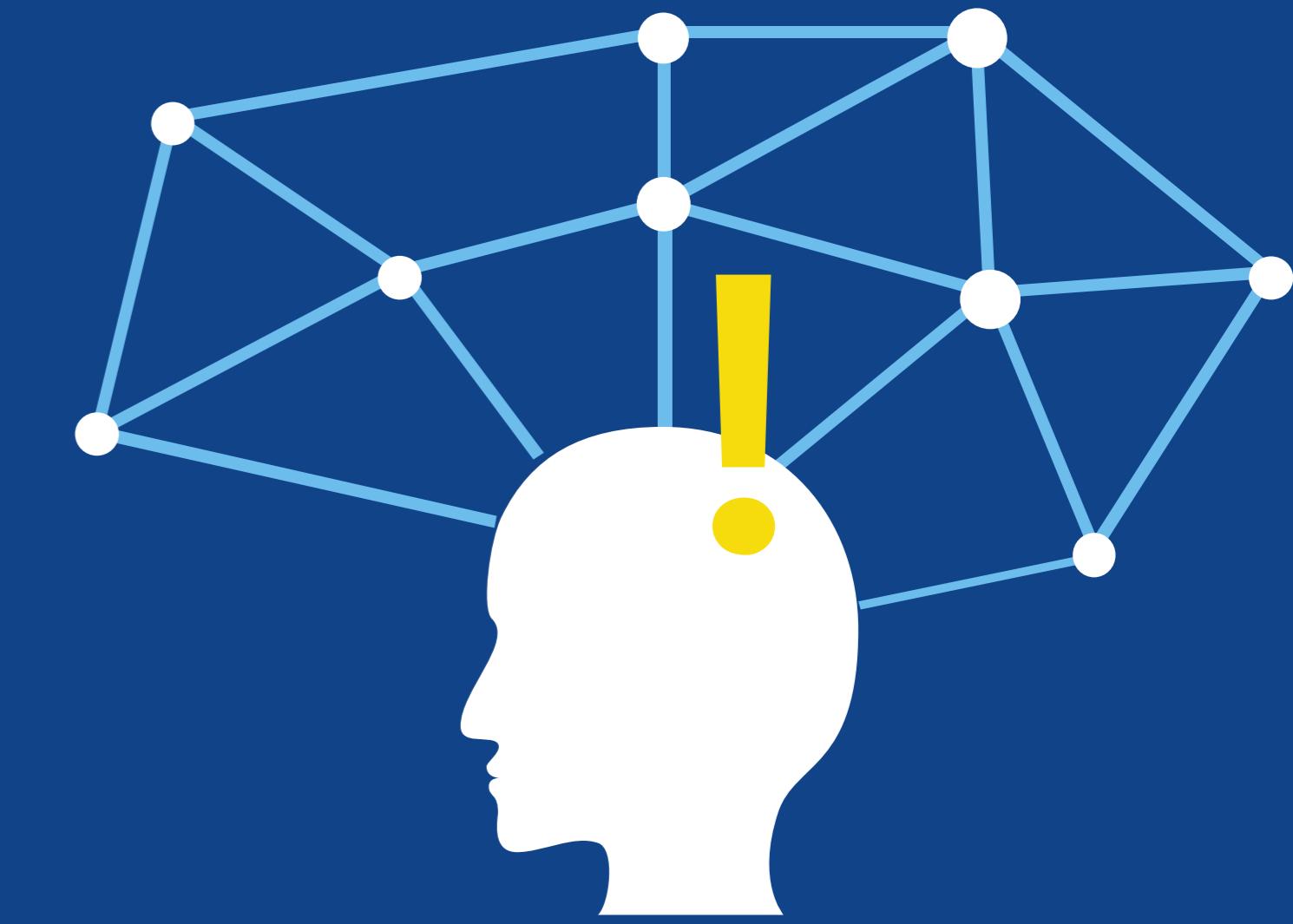


기존 3D 스캐닝 시장

광학적 스캐닝
세밀한 곡면 측정

새로운 시장 요구

접근성 있는 가격대
전반적 치수 측정



“**합리적인 가격의
접촉식 3D 스캐닝 키트를
만들어 보자!**”



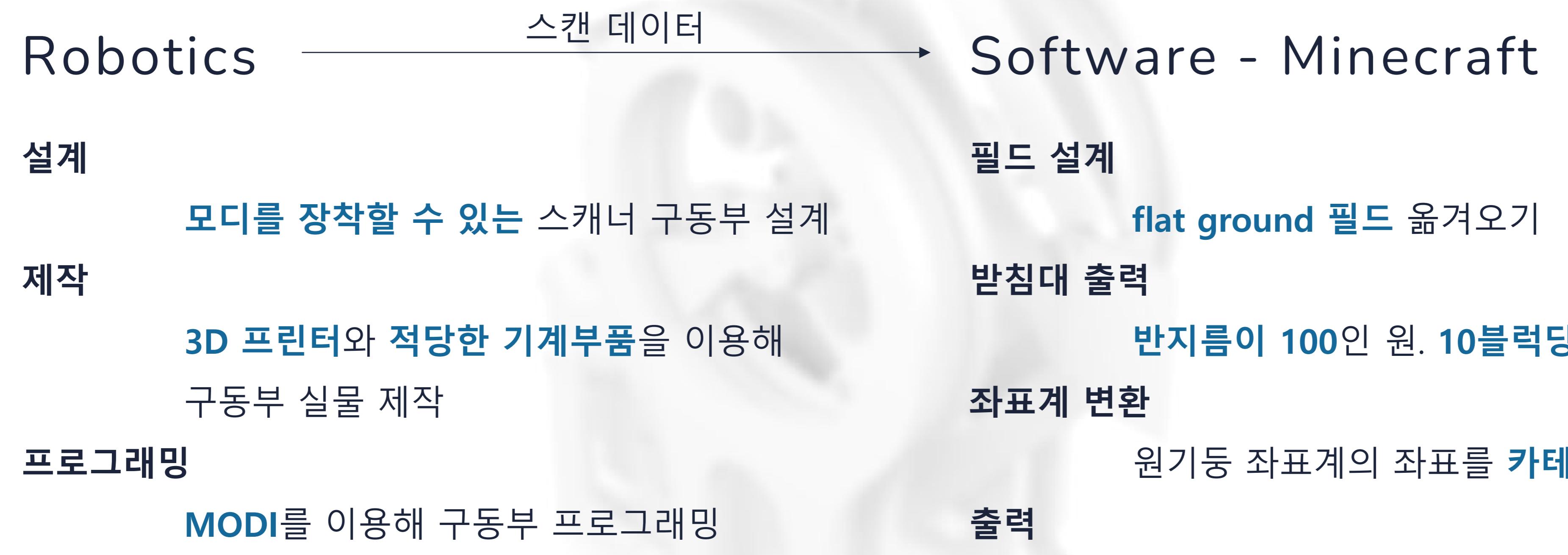


프로젝트 개요



2

프로젝트 개요



프로젝트 개요

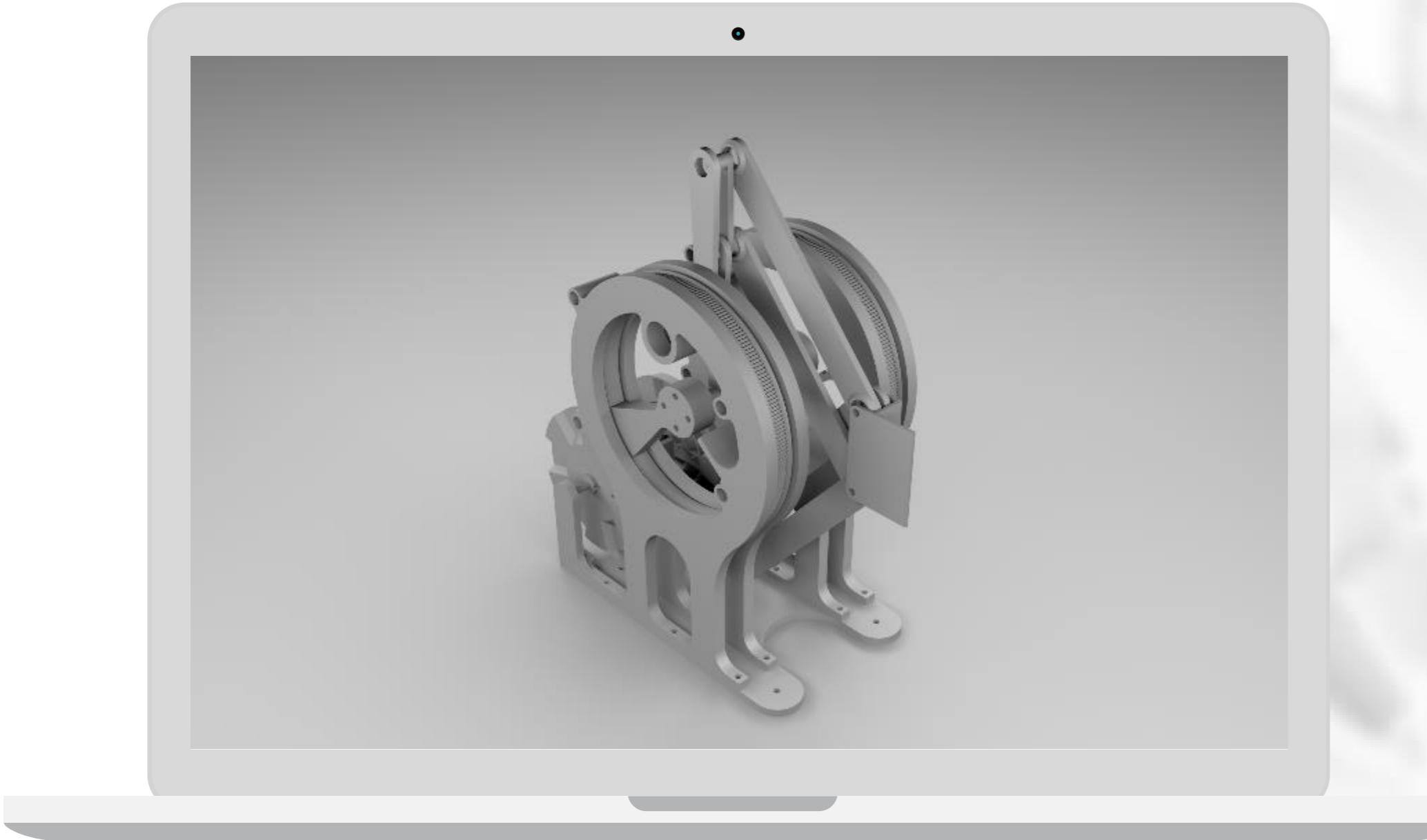




연구 과정

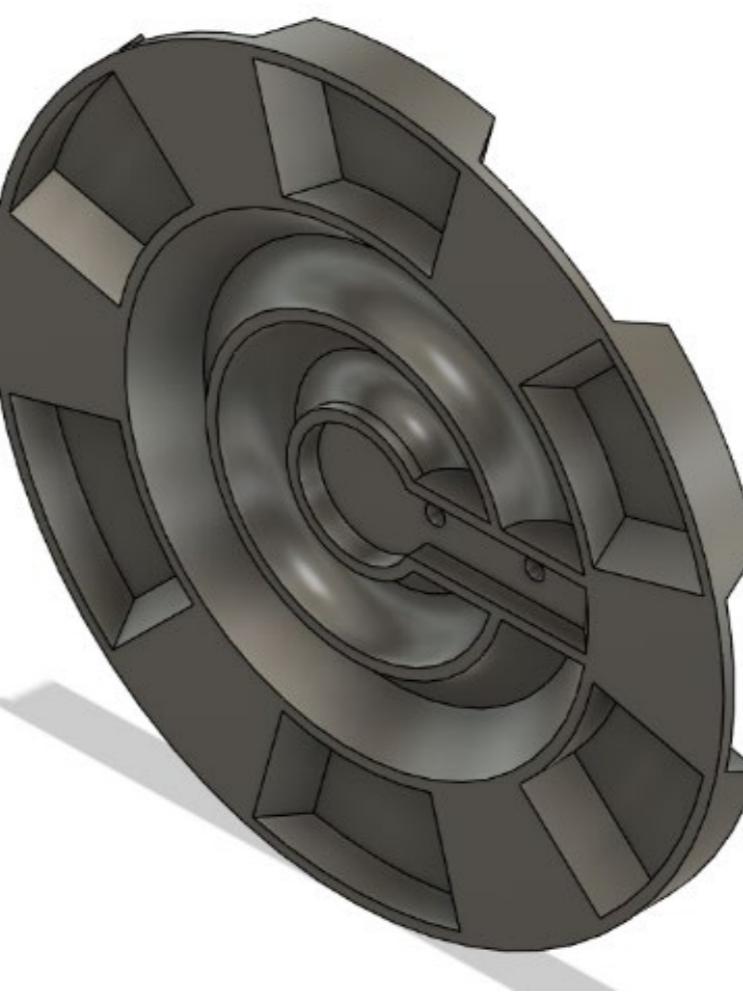
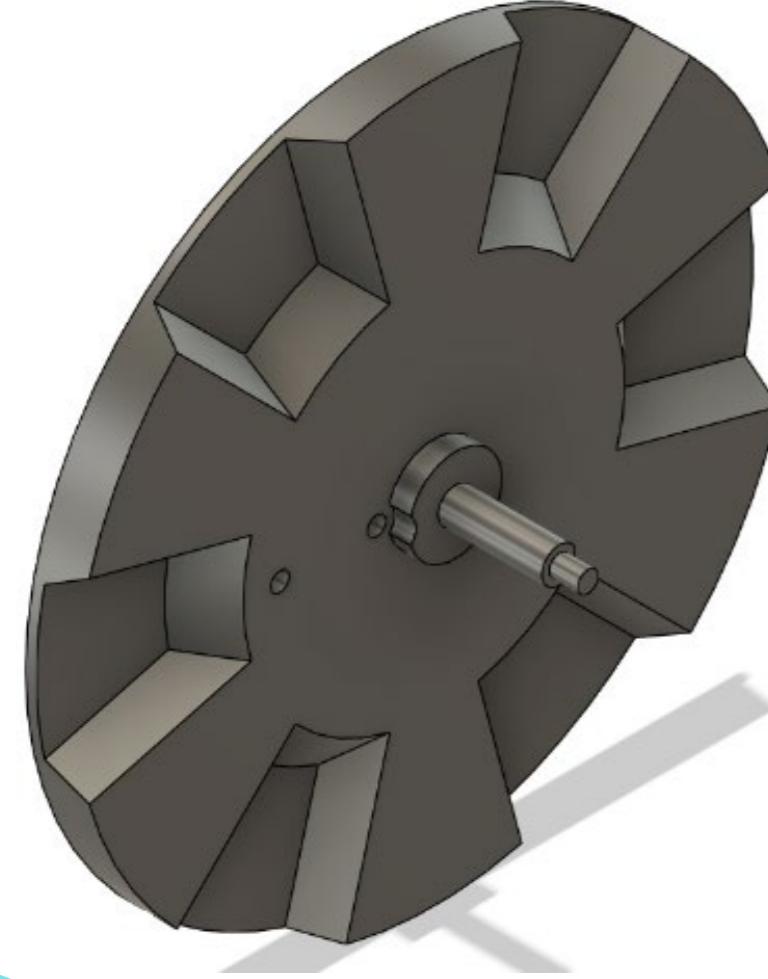
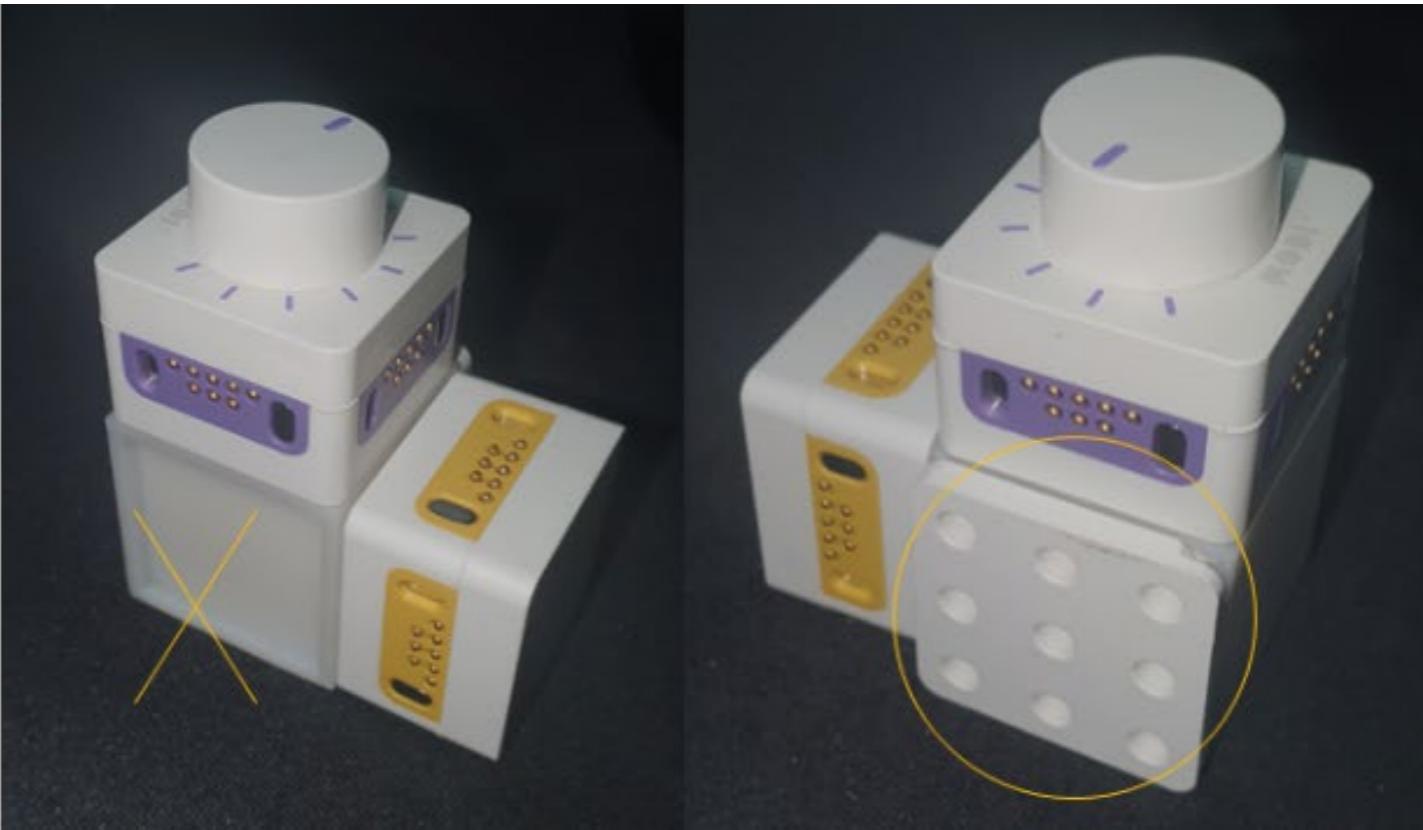
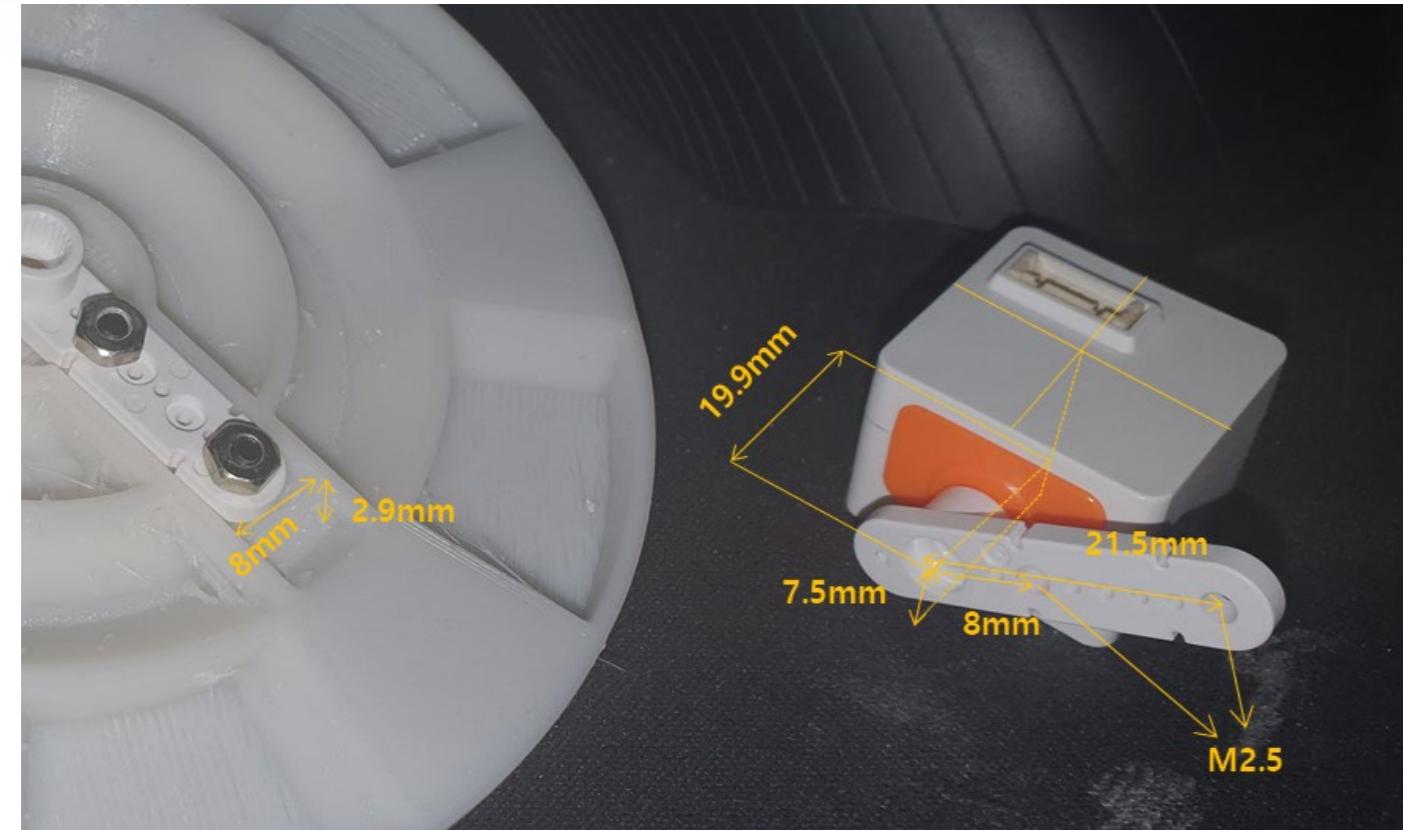


3

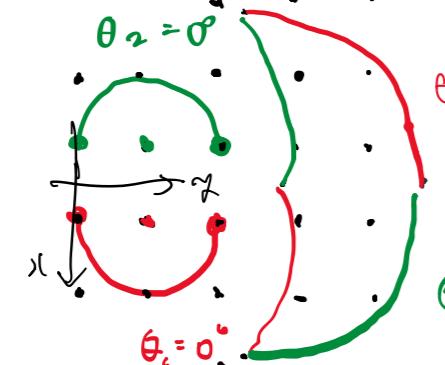
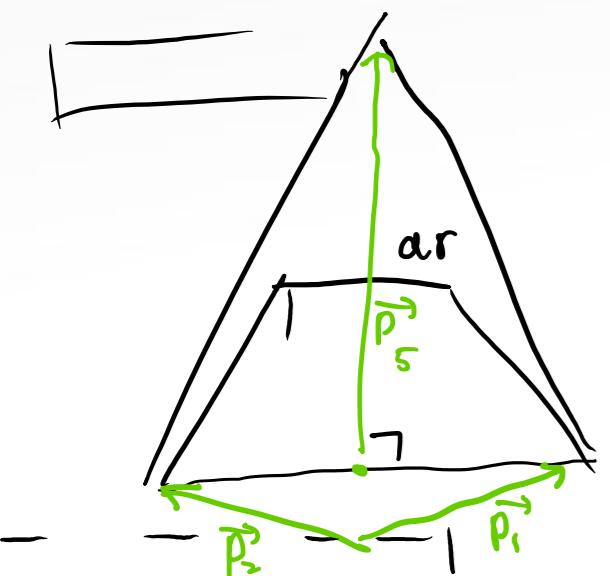


1. 스캐너 제작

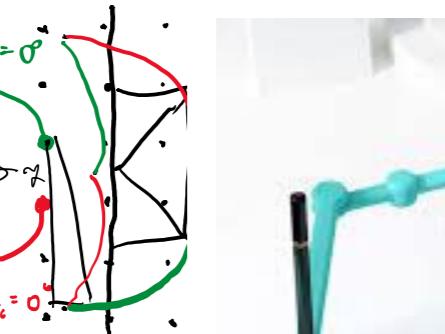
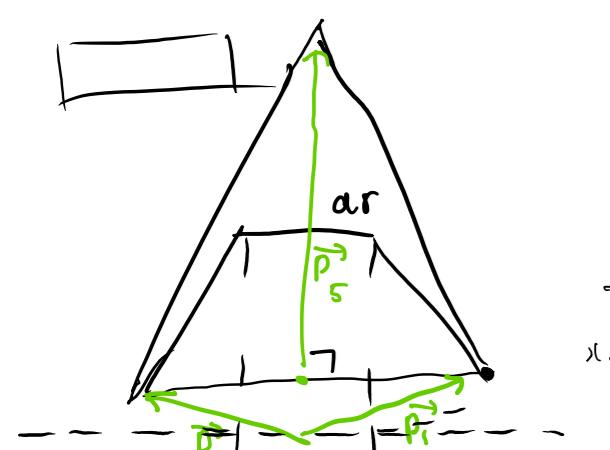
MODI 모듈 크기 측정 및 엔코더 설계



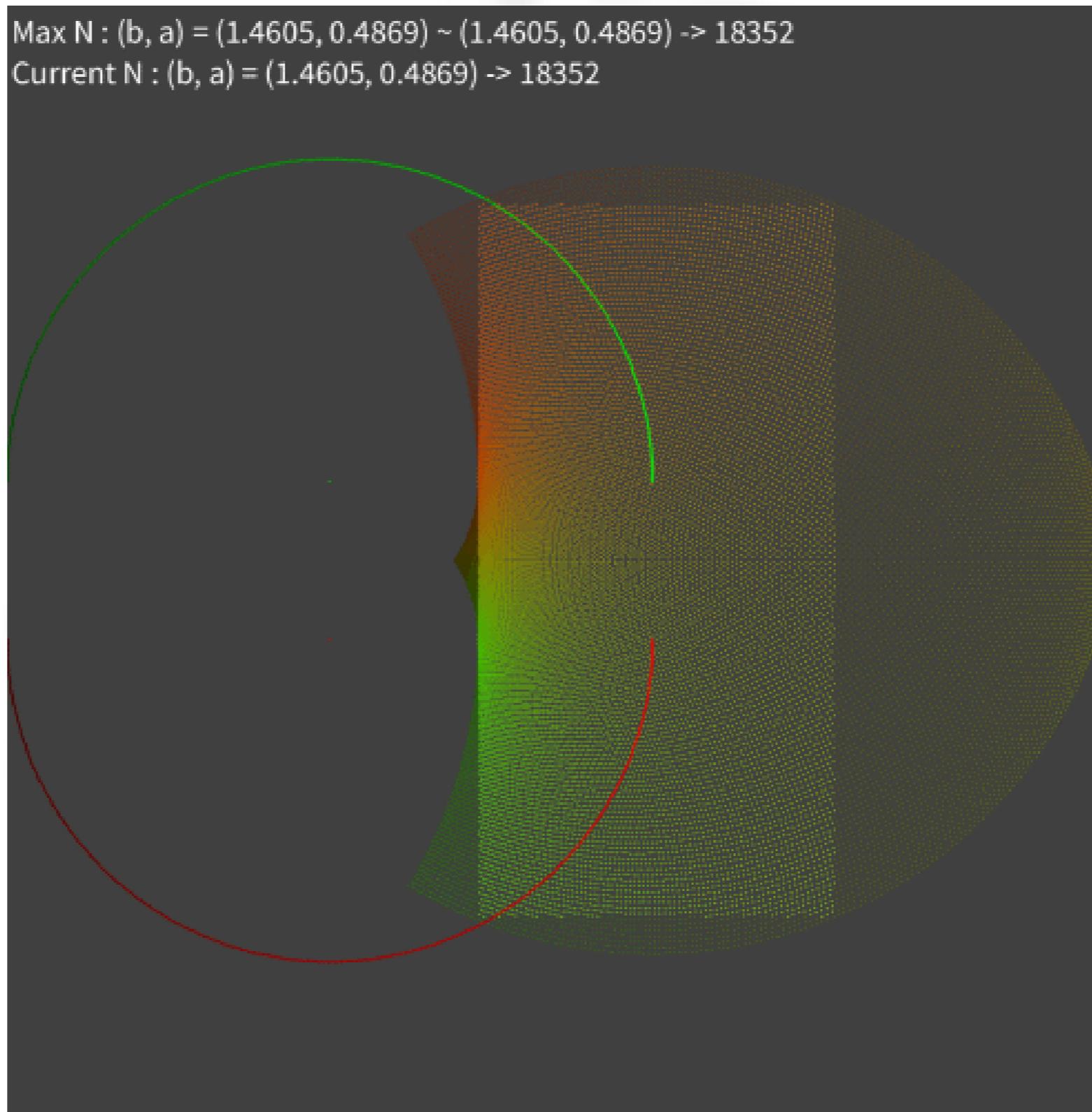
구동축 크기 및 간격 설정



$$\vec{p}_t = \vec{p}_s + \vec{p}_f \quad \left(\frac{|\vec{p}_f|}{2}\right)^2 + (|\vec{p}_s|)^2 = (b\gamma)^2$$



Max N : (b, a) = (1.4605, 0.4869) ~ (1.4605, 0.4869) -> 18352
Current N : (b, a) = (1.4605, 0.4869) -> 18352

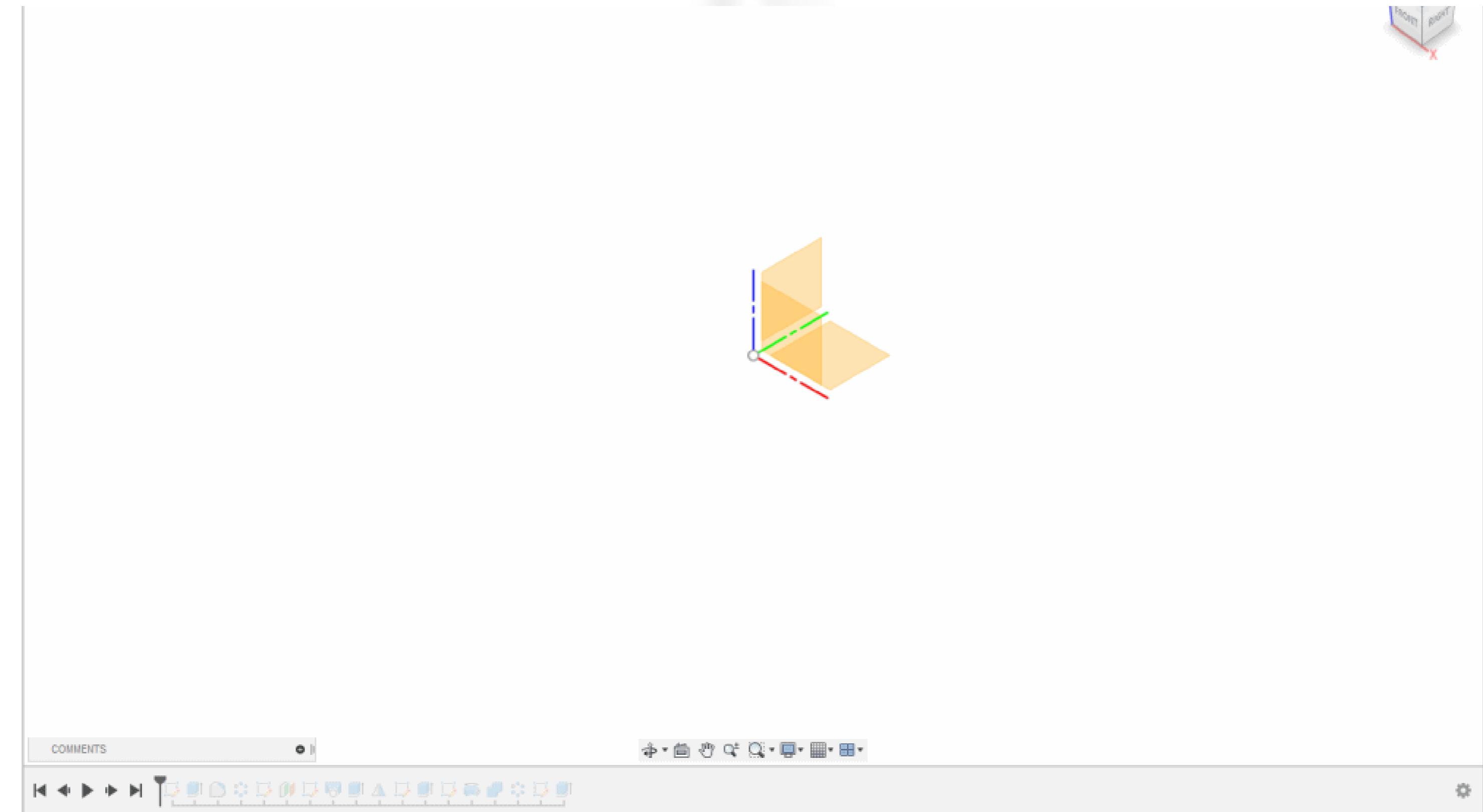


```

파일 편집 스케치 Debug 도구 도움말
sketch_211122_MOONSCAN
55 case 3 :
56     at = round((at + largeUnit) * scaler) / ((float)scaler);
57     if(at > bt - 2) return false;
58     break;
59 case 4 :
60     bt = round((bt + smallUnit) * scaler) / ((float)scaler);
61     if(bt < 1.2) bt = 1.2;
62     if(bt > bt - 2) at = floor((2 + bt - 2) * scaler) / ((float)scaler);
63     break;
64 case 5 :
65     if(bt == btax) return false;
66     bt = round((bt + smallUnit) * scaler) / ((float)scaler);
67     if(bt > btax) bt = btax;
68     if(at < bt / 3) at = ceil(bt / 3 * scaler) / ((float)scaler);
69     break;
70 case 6 :
71     at = round((at - smallUnit) * scaler) / ((float)scaler);
72     if(at < bt / 3) return false;
73     break;
74 case 7 :
75     at = round((at + smallUnit) * scaler) / ((float)scaler);
76     if(at > 2 * bt - 2) return false;
77     break;
78 case 8 :
79     bt = constrain(bt, 1.2, btax);
80     default :
81         at = constrain(at, ceil(bt / 3 * scaler) / ((float)scaler), floor((2 + bt - 2) * scaler) / ((float)scaler));
82     break;
83 }
84 if(at == a && bt == b && node < 8) return false;
85 rt = minHeight / sqrt(bt * bt + 4 - at * at - 4) / (at + at + 4)) / 2, height / (bt + 2 - at);
86 rt = min(width / 2 + sqrt(bt * bt + 4 - at * at) / 2, height / (2 + at), rt);
87 lt = (at > 2 * sqrt(bt - 1) ? (
88     2 + sqrt(bt * bt - (at / 2 + sqrt(bt * bt + 4 - at * at - 4) / (at + at + 4))) * (at / 2 + sqrt((bt * bt + 4 - at * at - 4) / (at + at + 4))) :
89     bt);
90 lt = sqrt(bt * bt - (bt - 2 - at / 2) * (bt - 2 - at / 2) / 2 - (bt - 2 + at / 2) / sqrt(2));
91 if(2 - lt > lt / sqrt(8)) {
92     lt = lt / sqrt(8);
93 }
94 }
```



본체 설계



본체 설계 결과

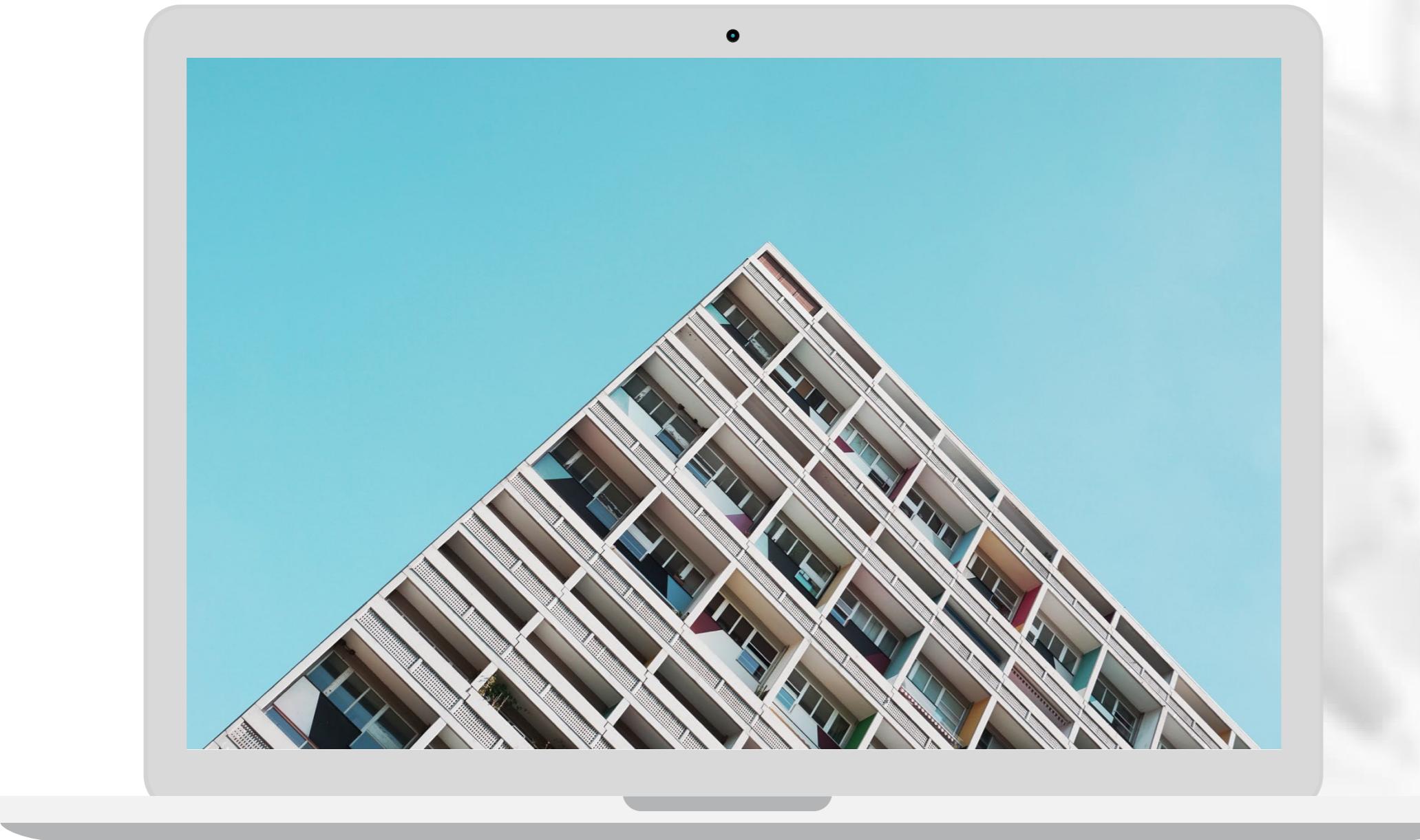


출력 및 조립

2번째 출력 모델

총 출력 시간 : 125시간
재질 : PLA+
레이어 두께 : 0.15mm





2. 파이썬 - 마인크래프트

Draw in Minecraft with Python

Step 1. Select Map



Problem

Minecraft
basic world is
too complex

Draw in Minecraft with Python

Step 1. Select Map



Solution

Use Superflat
world

Draw in Minecraft with Python

Step 2. Draw Map Structure



Problem

Flat map hard
to expect
distance

Draw in Minecraft with Python

Step 2. Draw Map Structure

set_floor.py

```
# 바닥에 원형으로 BLOCK을 까는 함수  
# 반지름이 커질수록 적용되는 각도를 줄여 더 많은 점을 찍도록 구현  
for i in range(1, 100):  
    for theta in np.arange(0, 2 * math.pi, 0.05 / i):  
        x = math.cos(theta) * i  
        y = math.sin(theta) * i  
        x = round(x)  
        y = round(y)  
        # Quartz Block 사용  
        mc.setBlock(x, 0, y, 155)
```



Draw in Minecraft with Python

Step 2. Draw Map Structure

set_floor.py

```
# 바닥에 반지름이 r인 선 긋는 함수
def round_line(r):
    for theta in np.arange(0, 2 * math.pi, 0.05 / r):
        x = math.cos(theta) * r
        y = math.sin(theta) * r
        x = round(x)
        y = round(y)
        # Chiseled Quartz Block 사용
        mc.setBlock(x, 0, y, 155, 1)

# 반지름 10 단위로 바닥에 선 긋기
for r in range(10, 100, 10):
    round_line(r)
```



Draw in Minecraft with Python

Step 2. Draw Map Structure

set_floor.py

```
# x,y,z 축 그리기  
# x 축 (RED)  
mc.setBlocks(1, 0, 0, 100, 0, 0, 251, 14)  
# 우리가 일반적으로 생각하는 Cartesian coordinate로 만들기 위해  
# minecraft상의 y축의 음의방향을 Cartesian coordinate의 y축으로 적용한다.  
# y 축 (GREEN)  
mc.setBlocks(0, 0, -100, 0, 0, -1, 251, 13)  
# z 축 (BLUE)  
mc.setBlocks(0, 1, 0, 0, 256, 0, 251, 11)
```



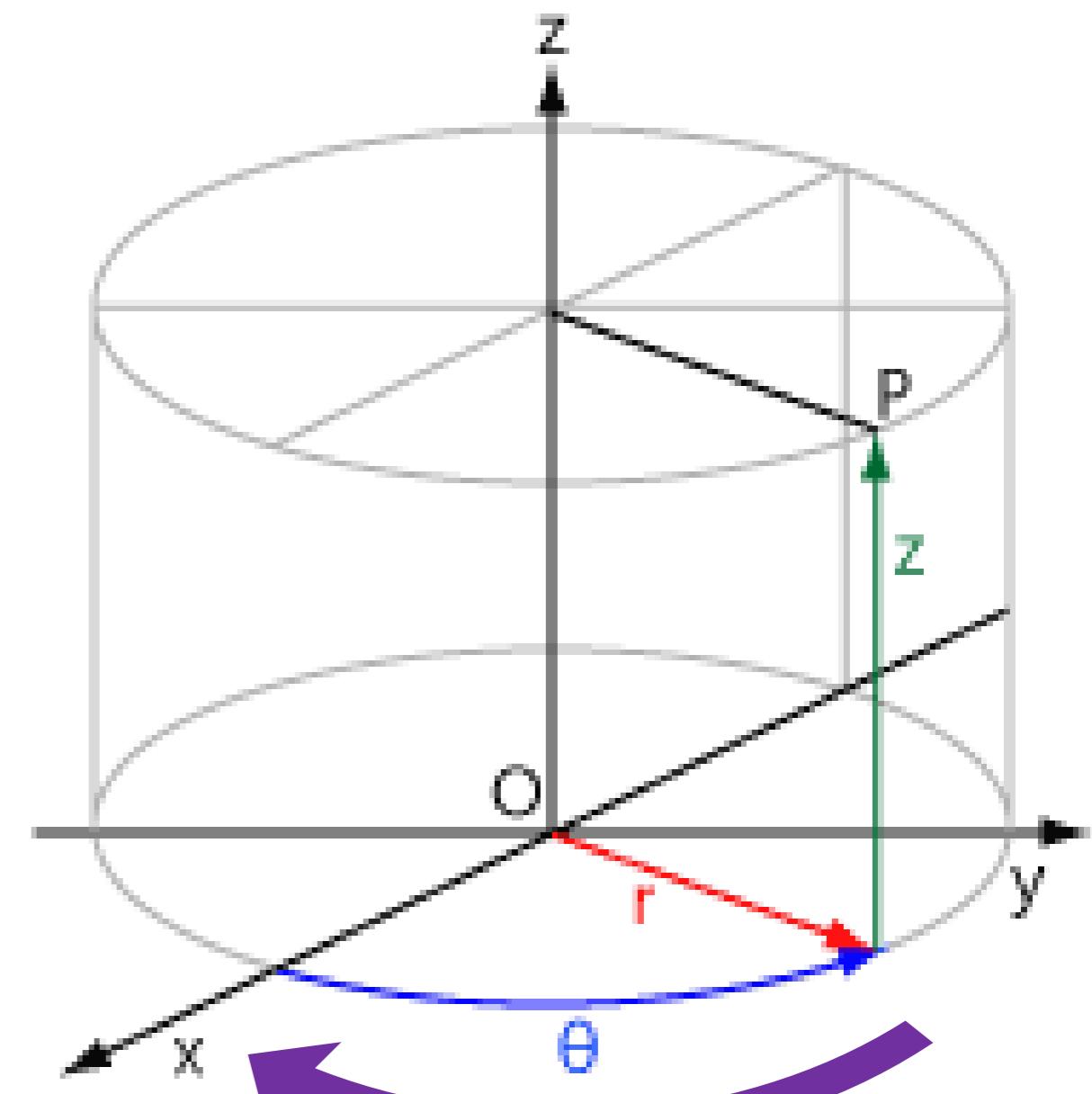
Draw in Minecraft with Python

Step 3. Define draw input to Minecraft function

3D_Draw.py

```
# Cylindrical coordinate의 좌표를 입력받아 마인크래프트에 그리는 함수 (theta의 단위는 라디안)
# 우리가 일반적으로 생각하는 Cartesian coordinate로 만들기 위해 theta는 음수로 적용한다.
def DRAW(r, theta, z):
    # Cylindrical to Cartesian
    x = r*(math.cos(-theta))
    y = r*(math.sin(-theta))
    z = z
    # 블럭의 Cartesian coordinate 좌표를 정수로 반올림
    x = round(x)
    y = round(y)
    z = round(z)
    # 마인크래프트 공간상에 블럭 생성 (YELLOW)
    mc.setBlock(x, z, y, 251, 4)
```

General Cylindrical

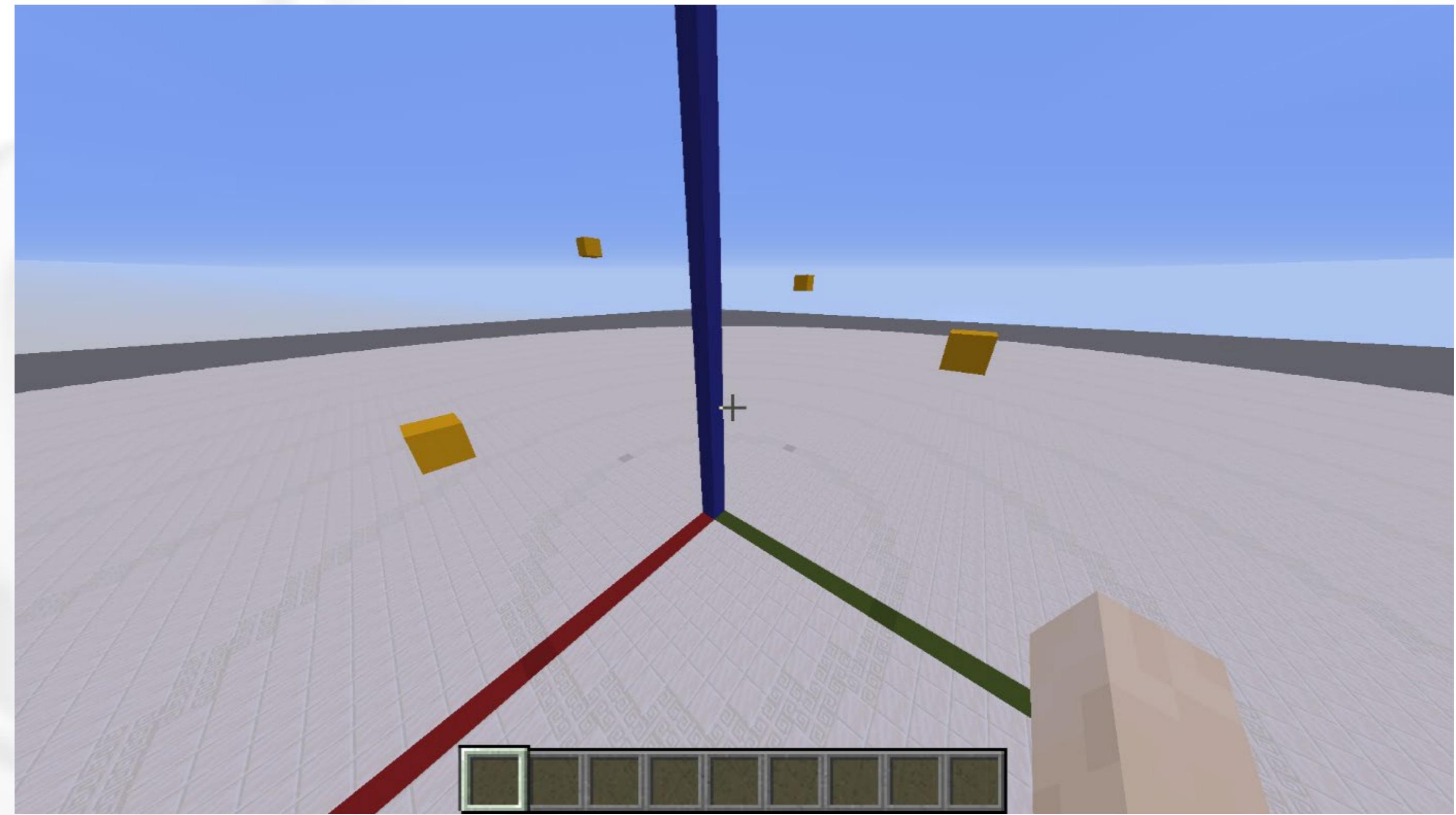


θ direction in Minecraft

Draw in Minecraft with Python

Step 4-1. test 1
3D_Draw.py

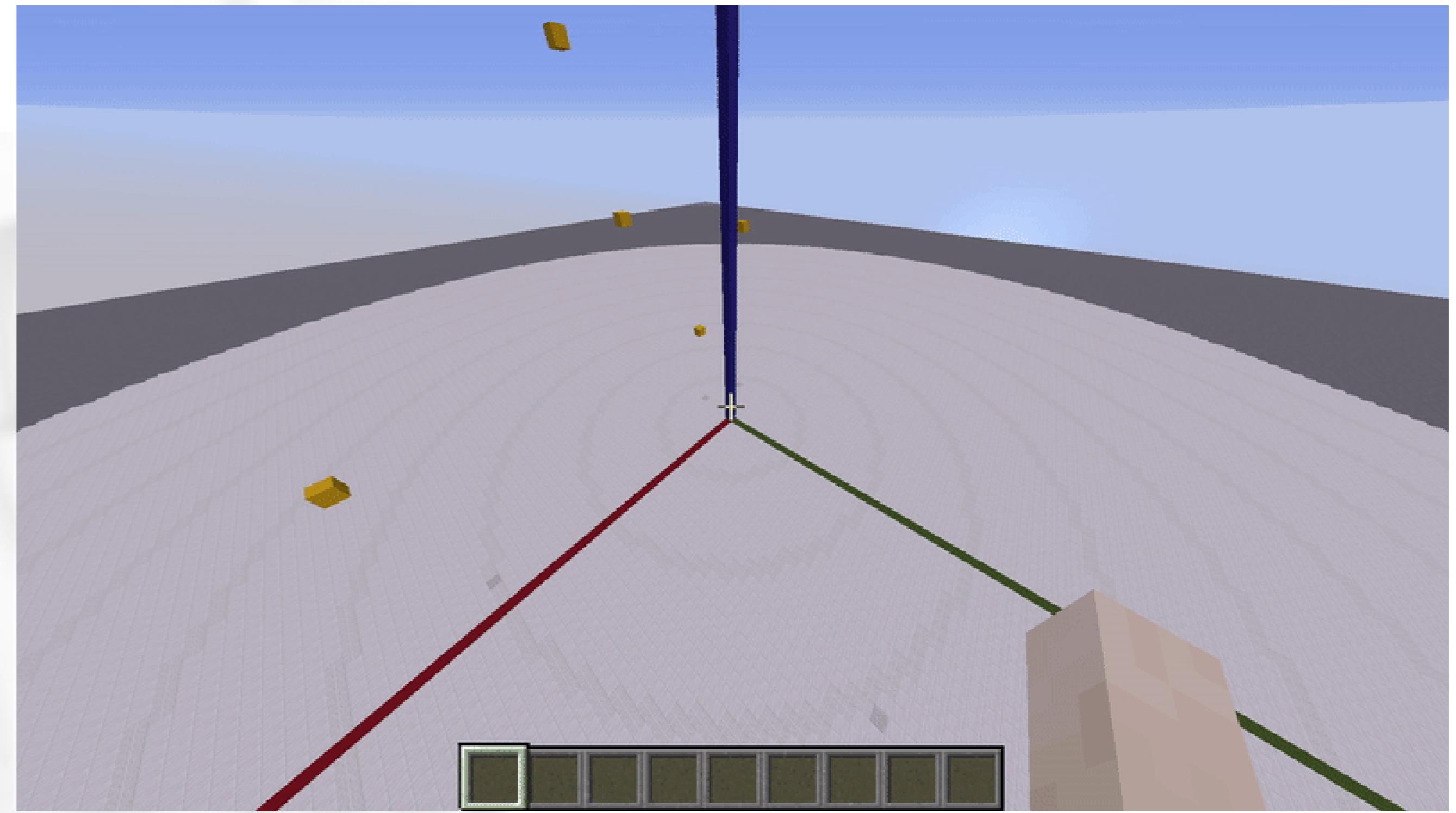
```
# test 1 for pi increase
t1_z = 10
DRAW(10, 0, t1_z)
DRAW(10, pi / 2, t1_z + 2)
DRAW(10, pi, t1_z + 4)
DRAW(10, pi * 3 / 2, t1_z + 6)
```



Draw in Minecraft with Python

Step 4-2. test 2 3D_Draw.py

```
# test 2 for random points
import random
import time
for i in range(0, 1000):
    t2_r = random.random()*50
    t2_z = random.random()*49+1
    t2_theta = random.random()*2*pi
    DRAW(t2_r, t2_theta, t2_z)
    time.sleep(0.1)
```



Draw in Minecraft with Python

Step 4-2. test 2(continue)



Draw in Minecraft with Python

Step 5. clear

clear.py

```
def clearwithZ():
    mc.setBlocks(-150, 1, -150, 150, 256, 150, 0)
    # z 축 (BLUE)
    mc.setBlocks(0, 1, 0, 0, 256, 0, 251, 11)

def clearALL():
    mc.setBlocks(-150, 1, -150, 150, 256, 150, 0)
    mc.setBlocks(-150, 0, -150, 150, 0, 150, 1)
```

Draw in Minecraft with Python

Step 5. clear(continue)

clearwithZ



clearALL

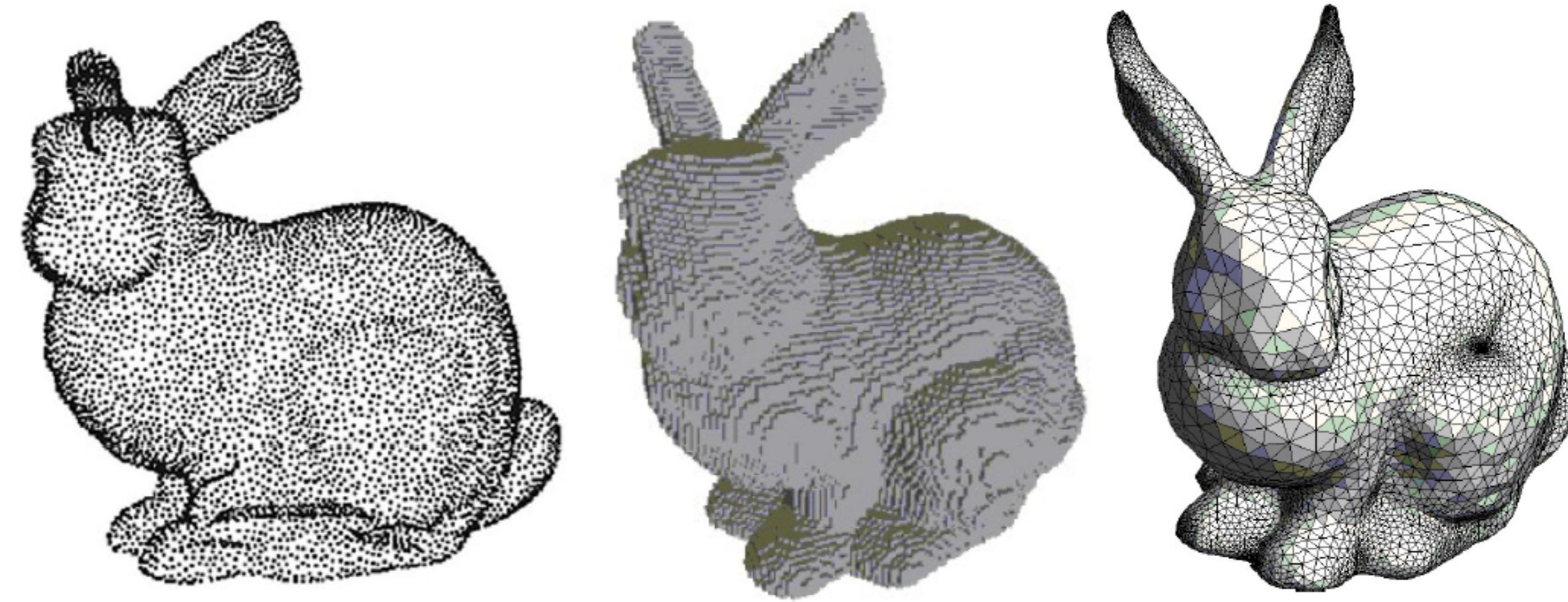




3. 3D File - Point Cloud

Save Scan Result as File

3D Model



Point Cloud

- Irregular
- Polygon X
- Continuous
- Hard Polygonizable

Voxel

- Regular
- Polygon X
- Discrete
- Easy Polygonizable
- (Low) Resolution

Mesh

- Regular (\triangle)
- Polygon O
- Continuous

Save Scan Result as File

Using library Open3d : <http://www.open3d.org/>

Open3d_pcl.py

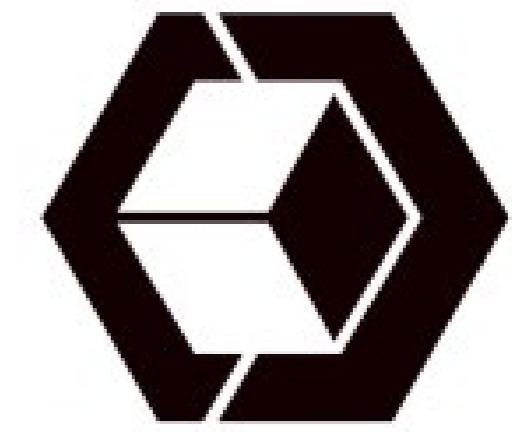
```
# 포인트클라우드를 만들기 위해
# 1. 각 점들의 x, y, z 좌표를 numpy 2D array로 저장
# 2. open3d의 pointcloud type 변수 생성
# 3. open3d의 Vector3dVector 함수로 1번에서 만든 2D numpy array를 변환해 2번의 pointcloud형 변수로 저장
PointCloud_array = temp_np_array

# PointCloud 생성(초기화)
PointCloud_data = o3d.geometry.PointCloud()

# Numpy Array로 입력받아 PointCloud로 변환
PointCloud_data.points = o3d.utility.Vector3dVector(PointCloud_array)
print(PointCloud_data)

# PointCloud data를 담을 수 있는 .xyz 파일로 저장
o3d.io.write_point_cloud("pcd_test1.xyz", PointCloud_data)

# visualization : open3d의 함수를 이용해 pointcloud 시각화
print(np.asarray(PointCloud_data.points))
o3d.visualization.draw_geometries([PointCloud_data])
```



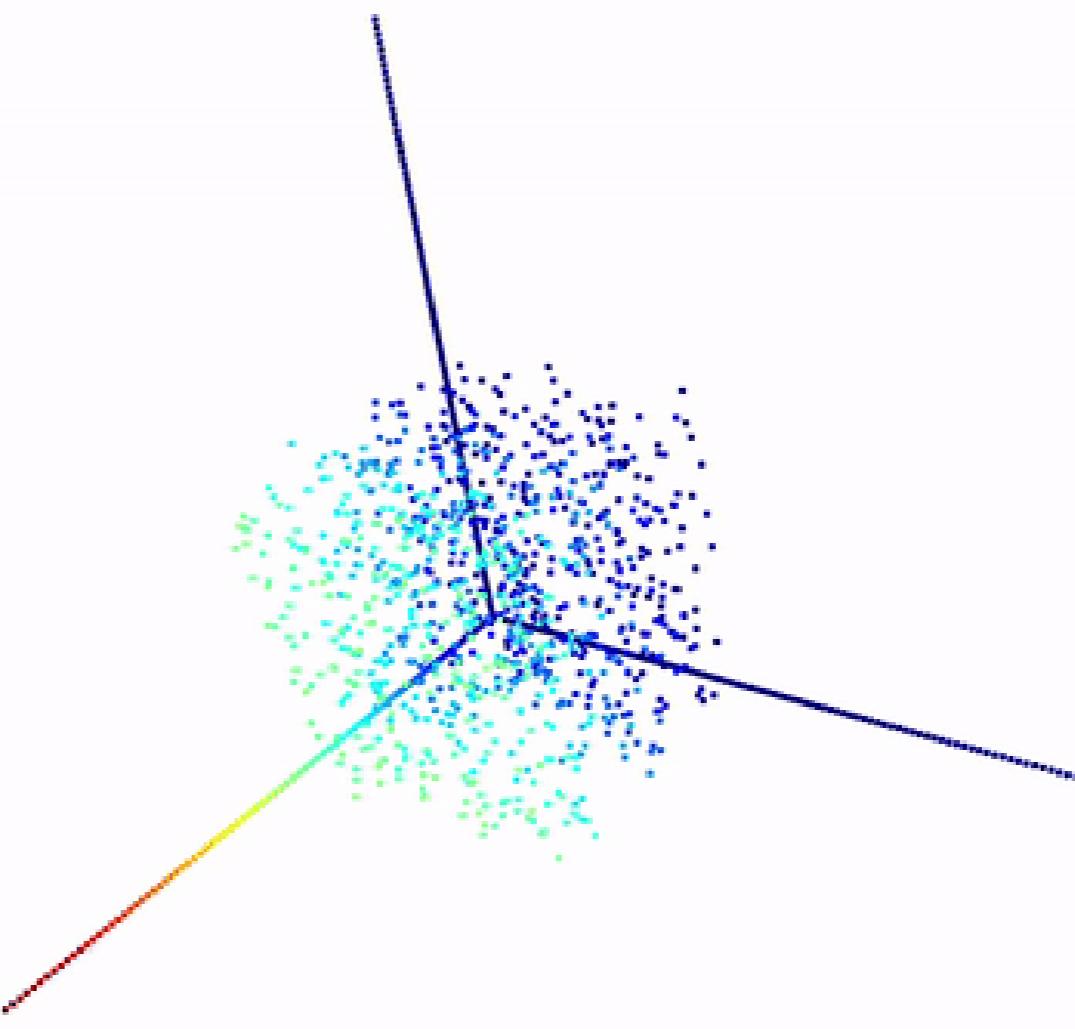
OPEN3D

Save Scan Result as File

Test Using Open3d

Open3d_pcl.py

```
# test1
temp_np_array = np.array([[0, 0, 0]], dtype=np.float32)
# x축
for i in range(1, 100):
    x, y, z = i, 0, 0
    temp_np_array = np.append(temp_np_array, np.array([[x, y, z]]), axis=0)
# y축
for i in range(1, 100):
    x, y, z = 0, i, 0
    temp_np_array = np.append(temp_np_array, np.array([[x, y, z]]), axis=0)
# z축
for i in range(1, 100):
    x, y, z = 0, 0, i
    temp_np_array = np.append(temp_np_array, np.array([[x, y, z]]), axis=0)
# 랜덤 점 1000개 생성
for i in range(0, 1000):
    x = random.random() * 50
    y = random.random() * 50
    z = random.random() * 50
    temp_np_array = np.append(temp_np_array, np.array([[x, y, z]]), axis=0)
```

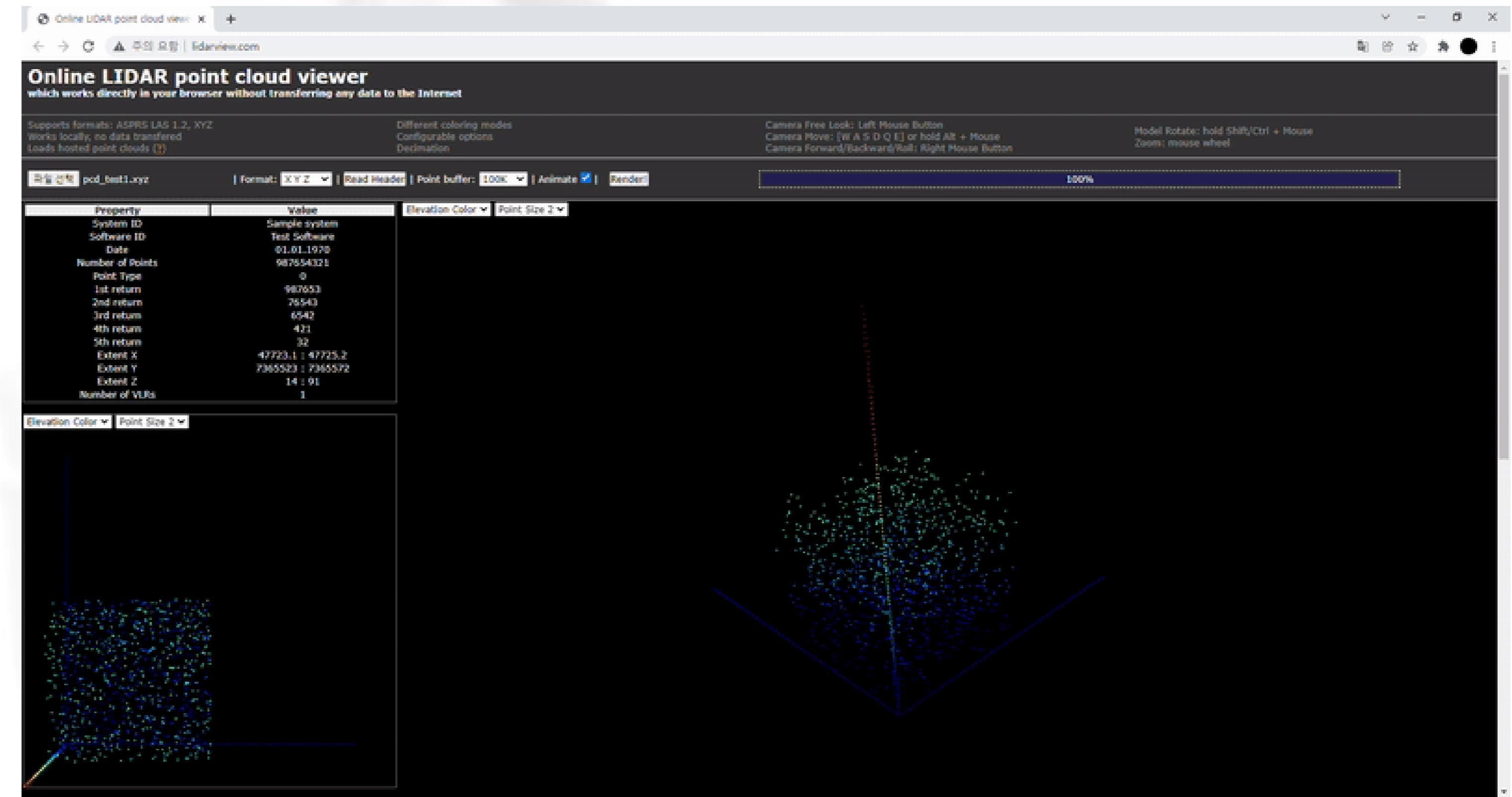


Save Scan Result as File

Test Using Open3d(continue)

pcd_test1.xyz

<http://lidarview.com/>

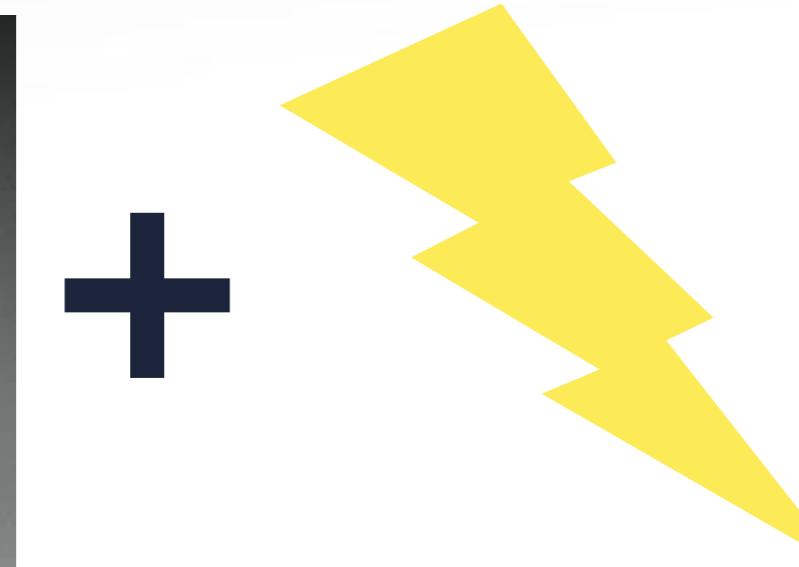




Marketability

4

제품 단가



About 600g

Electricity



Acryl 3T Plate

623 bearing * 3
604 bearing * 12
608 bearing * 2

M2.5x10 * 4
M2.5 nut * 4
M4x10 * 22 + a
M4x16 * 8 + a
M4 nut * 26 + a
M8x15 * 2
고무발 * 6



20,000 KRW ↓

하우징

2,000 KRW ↓

기계 부품

1,000 KRW

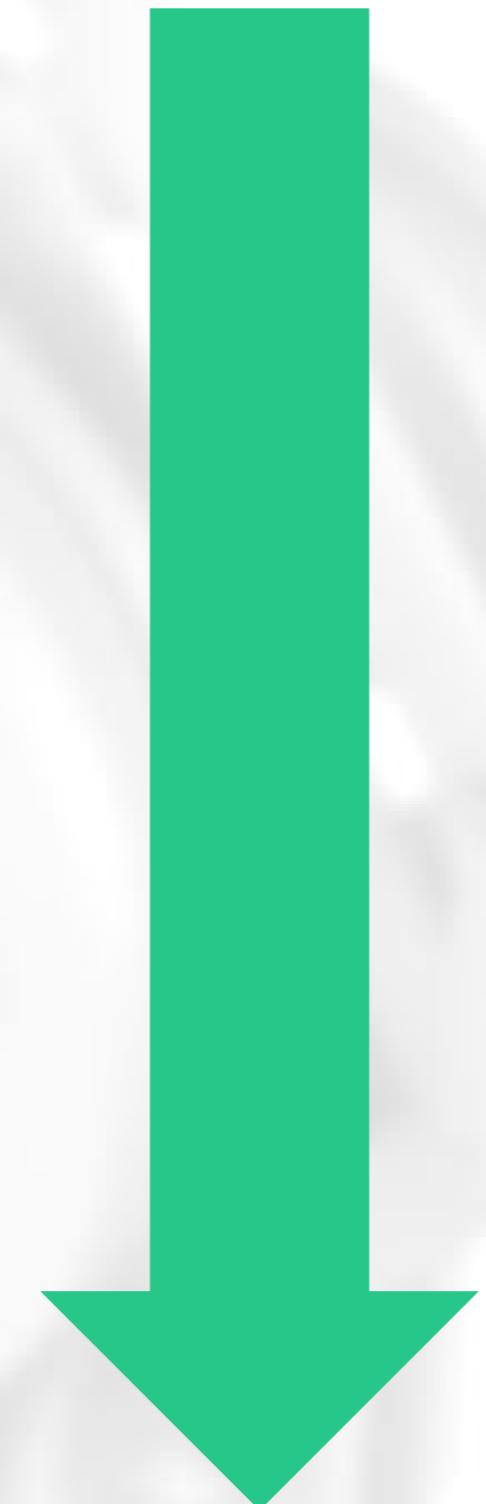
조립 공정

23,000 KRW

원가

39,900 KRW

판매가



42%

MARGIN



Need to
Improvement

5

Problems in Robotics

모디 모터 오류

전원을 키고 모디가 초기화되지 않으면 모터가 멋대로 돌아가는 현상 발생.

이로 인해 출력물이 한계 범위를 넘어서 돌아서 휘는 등의 문제 발생.

엔코더 파트 오류

모디 IR이 모듈별로 센서값이 너무 큰 폭으로 차이가 나고, 엔코더가 얇다보니 투과되는 현상으로 엔코더 정상 동작 불가능.

전체적인 유격 문제 여전히 심함.

1번째 출력물 실패로 얻은 개선사항이, 오히려 2번째 출력물에 실패요인을 만듬.

ex) 안끼워져서 유격줌 -> 너무 헐렁해짐 / 팔이 너무 약해서 팔 두께 키움 -> 팔에 간섭 생김.

-> 전자부품 문제는 기존 아두이노에서 사용되는 스텝퍼 모터나 앤드스탑 스위치 등을 사용하고, 제작 환경도 모디가 아닌 아두이노 등으로 변경해야 해결될 듯 함.

유격 문제의 경우, 조인트를 통해 가동범위 내에서 구동 상황을 시뮬레이션 해봐야 함.

Problems in Software

불연속적인 블럭

프루브의 출력 좌표에만 점을 찍다 보니, 출력 간격이 넓어지면 포인트가 불연속적이게 되는 문제 발생

마인크래프트 소프트웨어 구동 과정 너무 복잡함

마인크래프트와 스피롯 버킷이 모디스캔 E1의 드라이버격인 프로그램이나, 실행 과정이 상당히 번거로움. 실행 프로그램 자동화 필요

연동 필요

로보틱스 파트 제작이 늦어져, 아직 3D 스캐너와 마인크래프트 프로그램, 을 연동시켜보지 못함. 완성 후 테스트 필요



Thanks!

Any questions?