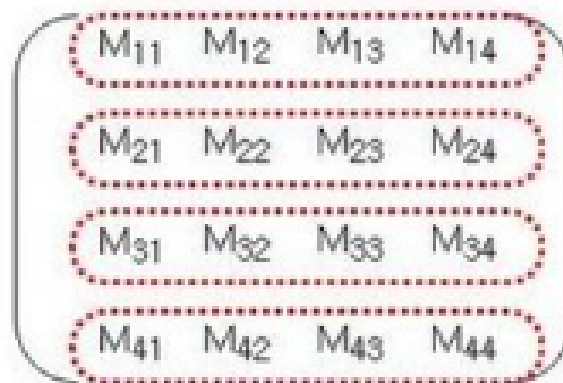


3D 그래픽스 행렬

Dartpy의 Skel 파일들 >> D3D 방식(행 우선 방식)

Opengl >> 열 우선 방식

열 우선 방식



행 우선(row-major)

>> 열 우선 방식의 변환 행렬 정리

변환 행렬(Transform Matrix) 정리

● 단위행렬

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

● 이동변환행렬

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ T_x & T_y & T_z & 1 \end{pmatrix}$$

● 크기변환행렬

$$\begin{pmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

● X축 회전행렬

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos & \sin & 0 \\ 0 & -\sin & \cos & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

● Y축 회전행렬

$$\begin{pmatrix} \cos & 0 & -\sin & 0 \\ 0 & 1 & 0 & 0 \\ \sin & 0 & \cos & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

● Z축 회전행렬

$$\begin{pmatrix} \cos & \sin & 0 & 0 \\ -\sin & \cos & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

행 우선 방식



행 우선(row-major)

```
<transformation>1 2 3 90 0 0</transformation>
```

```
print(joint.getTransformFromChildBodyNode())
```

```
>>[[ 1.          -0.          0.          1.          ]
 [ 0.          -0.44807362 -0.89399666  2.          ]
 [ 0.           0.89399666 -0.44807362  3.          ]
 [ 0.           0.          0.          1.          ]]
```

ex) (0,0,0)을 위의 매트릭스와 계산할 시 결과 값

```
>> (1,2,3)
```

```
>> 누전변환에서의 순서
```

```
>> 스케일링 >> 회전 >> 평행이동
```

Dartpy 함수

1. Skeleton

```
World.getSkeleton(number)
>> number : 추출하고자 하는 Skeleton의 순번
>> 첫번째일시 0, 두번째는 1 ...

Skeleton.getNumJoints()
>> 현재 주어진 Skeleton 내 존재하는 모든 Joint의 개수

Skeleton.getNumDofs()
>> 현재 주어진 Skeleton 내 존재하는 모든 Dof의 개수

Skeleton.getJoint(number)
>> number : 추출하고자 하는 Joint의 순번
>> 첫번째일시 0, 두번째는 1...

Skeleton.getPositions()
>> 현재 주어진 Skeleton 내 존재하는 Positions

Skeleton.getVelocities()
>> 현재 주어진 Skeleton 내 존재하는 Velocities
```

2. Joint

```
Joint.getNumDofs()
>> 현재 주어진 Joint 내 존재하는 모든 Dof의 개수

Joint.getPositions()
>> 현재 주어진 Joint 내의 Initial Position
>> Free joint : 6 dim vector
>> Ball joint : 3 dim vector
>> universal joint : 2 dim vector
>> Revolute joint : 1 dim vector

Joint.getVelocities()
>> 현재 주어진 Joint 내의 Initial Velocity
>> Free joint : 6 dim vector
>> Ball joint : 3 dim vector
>> universal joint : 2 dim vector
>> Revolute joint : 1 dim vector
```

3. Body

`Body.getWorldTransform()`

>> 현재 주어진 Body의 절대좌표계 변환행렬을 표현

>> 4*4행렬

`Body.getWorldTransform().translation()`

>> 현재 주어진 Body의 절대좌표계에서의 3차원 좌표

`Body.getWorldTransform().rotation()`

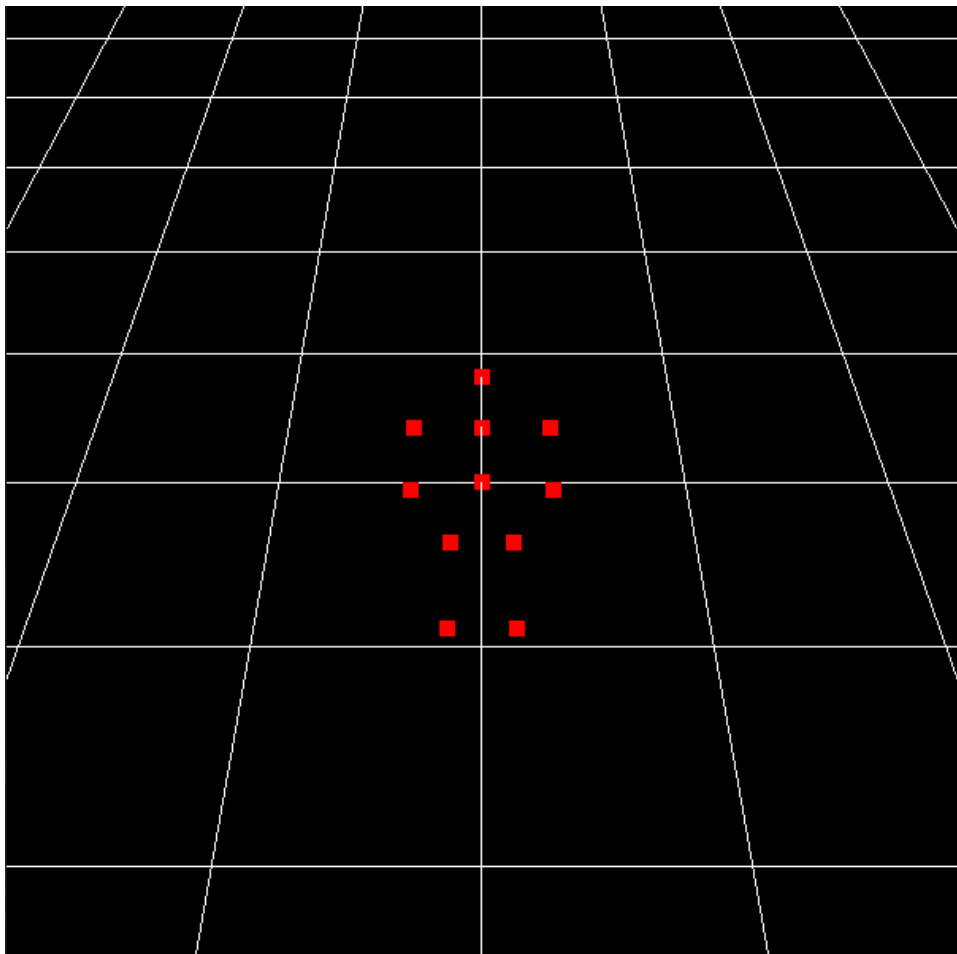
>> 현재 주어진 Body의 절대좌표계에서의 Euler 각도

Skeleton 그리기

Skeleton Body의 중심점 그리기

```
def Drawskeleton(self):
    world=dart.utils.SkelParser.readWorld(os.path.abspath('column.          skel'))
    Human=world.getSkeleton(0)
    Numbody = Human.getNumBodyNodes()
    Numjoint = Human.getNumJoints()

    BodyMetric = np.zeros(Numbody)
    JointMetric = np.zeros(Numjoint)
    for i in range(Numbody):
        Body=np.array(Human.getBodyNode(i).getWorldTransform().          translation())
        glPointSize(10)
        glBegin(GL_POINTS)
        glColor3f(255, 0, 0)
        glVertex3f(Body[0],Body[1],Body[2])
        glEnd()
```



Skeleton Body의 Shape 그리기

>> 어떤 함수로 Body의 visualization_shape나 collision_shape의 type과 size를 추출하는지 조사 필요

>> 추출될 경우, 어떤 식으로 그릴지 결정 필요

>> 바깥 테두리만 그리기

>> 바깥 테두리의 점들은 어떤 식으로 정할 것인가?

>> 그 점들은 연결하는 선들은 어떤 식으로 정할 것인가?

>> 삼각형으로 나누기

>> 삼각형을 어떻게 나눌 것인가?

>> 삼각형을 나눴다면, 그 점들의 위치는 어떻게 정할 것인가?

>> opengl에서 제공하는 사각형 그리기로 작성