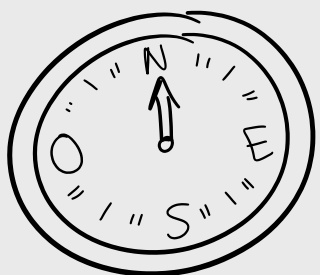
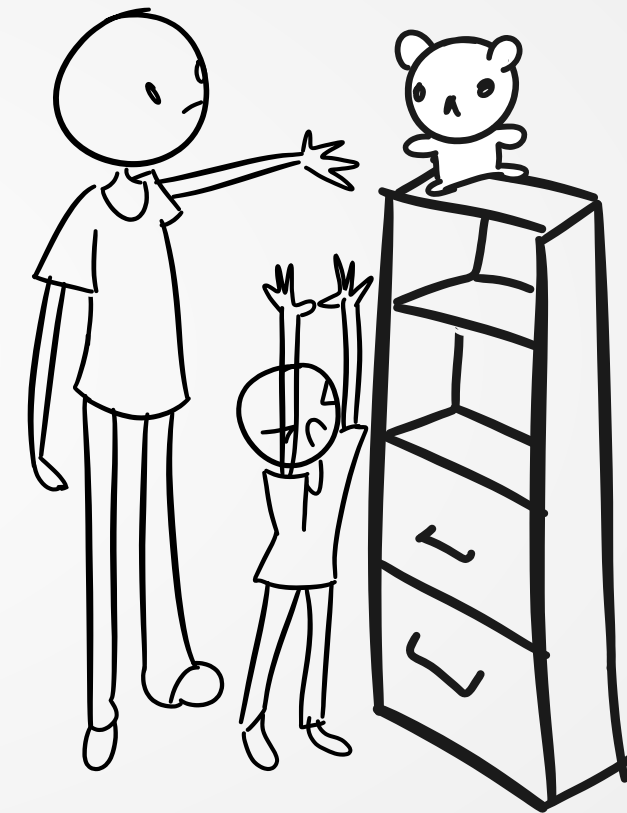
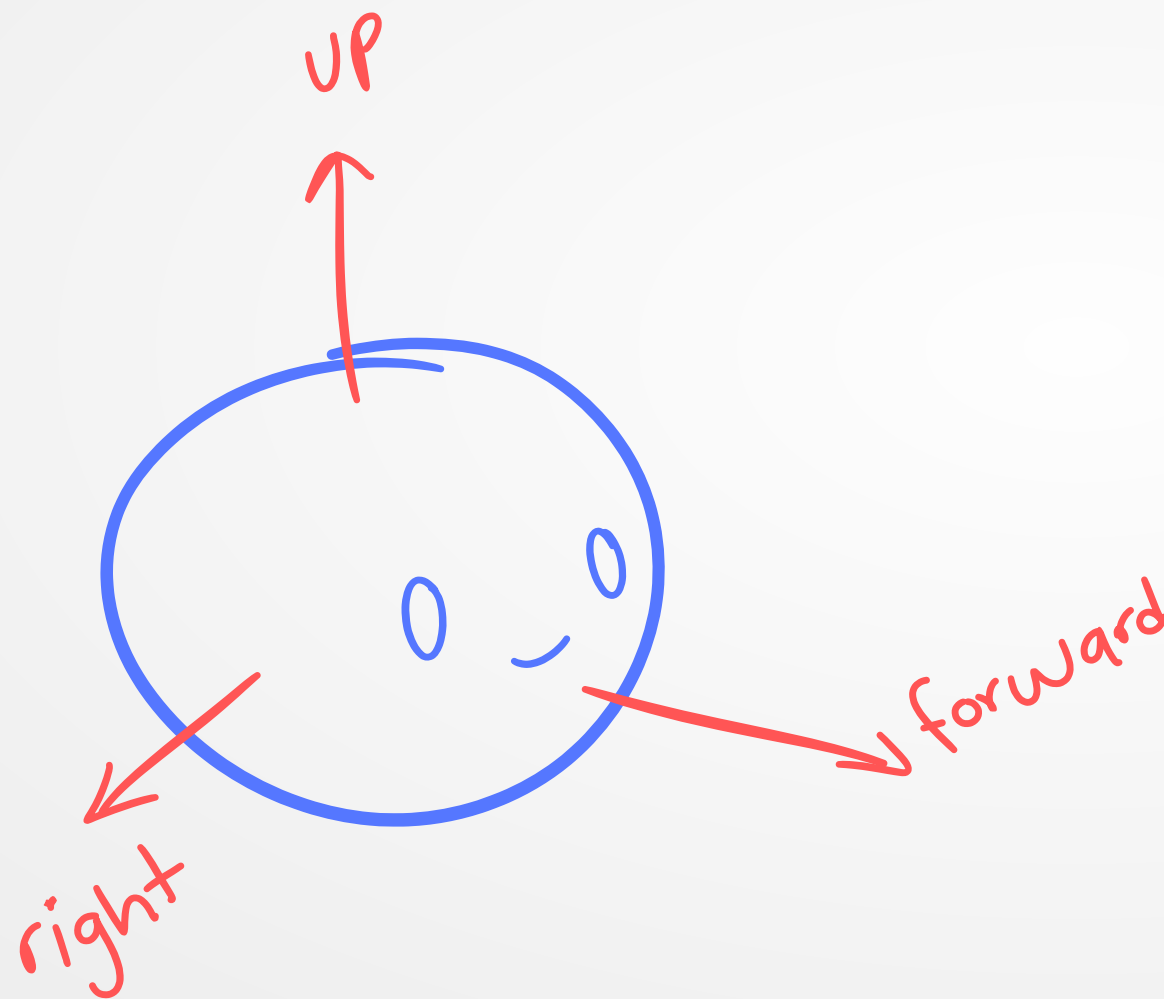
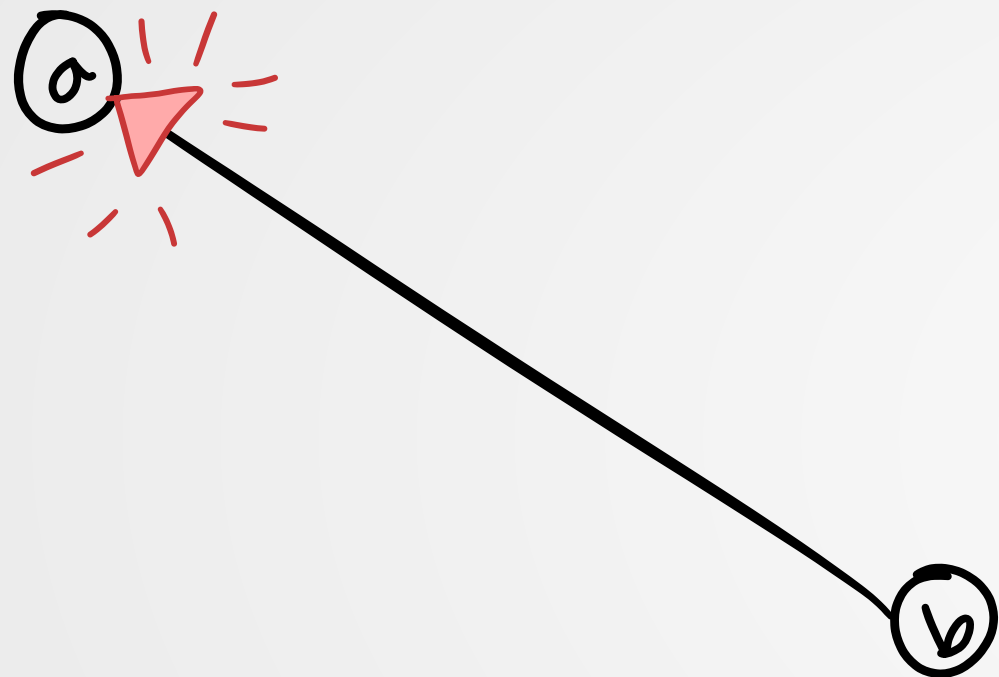


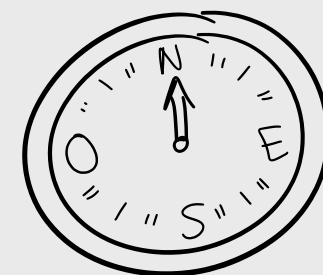
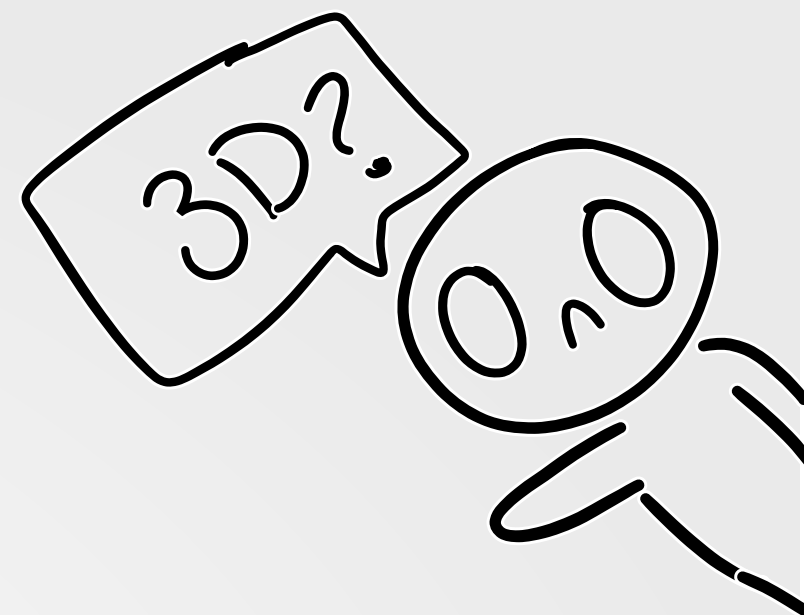
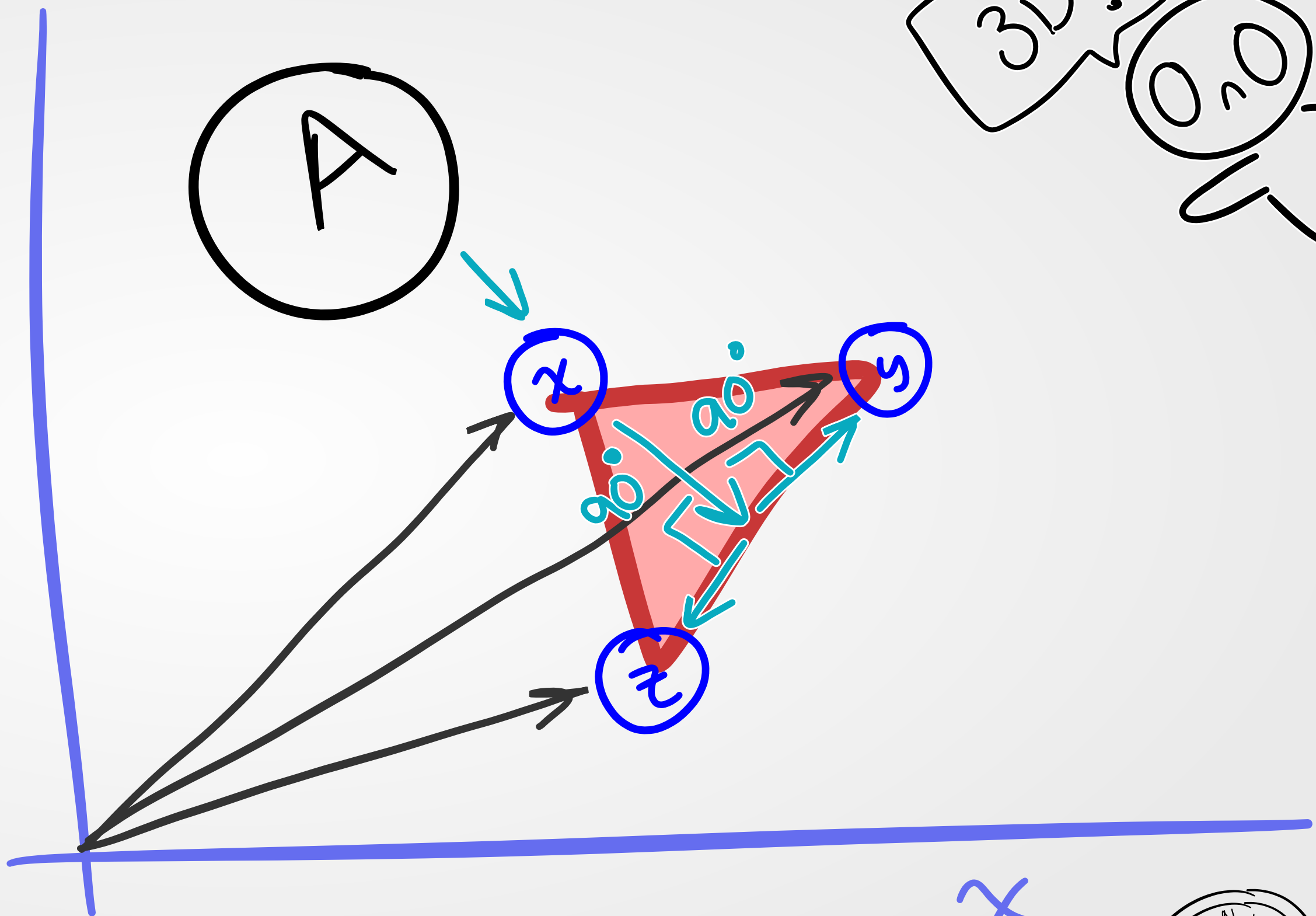
PoV y SCENE



PoV y SCENE



y

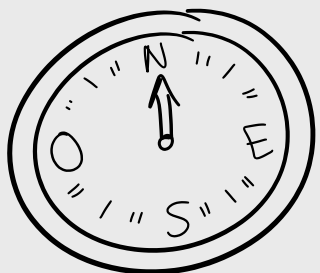


PoV y SCENE

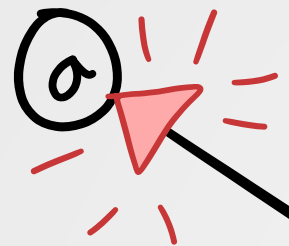
Handles Matrix

$$\begin{bmatrix} p_1 & p_2 & p_3 & p_4 \\ r_1 & r_2 & r_3 & r_4 \\ s_1 & s_2 & s_3 & s_4 \\ \emptyset & \emptyset & \emptyset & 1 \end{bmatrix}$$

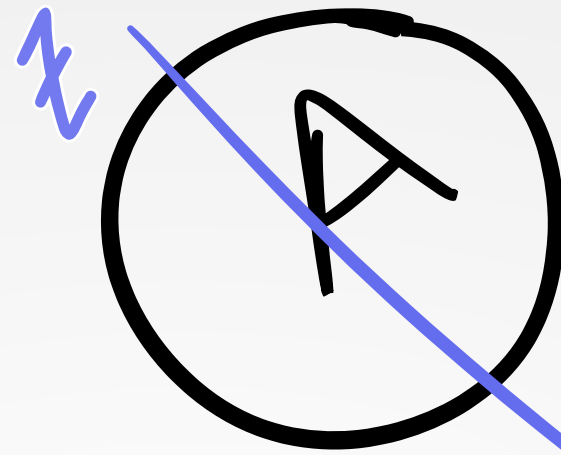
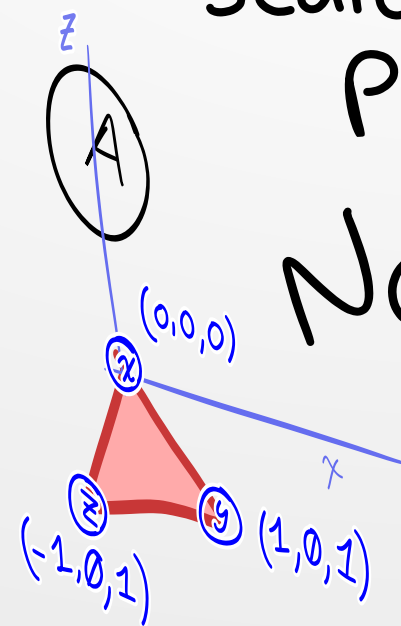
transformation
matrix
Matrix 4x4



PoV y SCENE



rotation,
Scale,
Position?
No Problem!

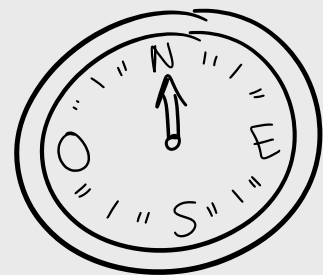


$(0,0,0)$

$(1,0,1)$



$(-1,0,1)$

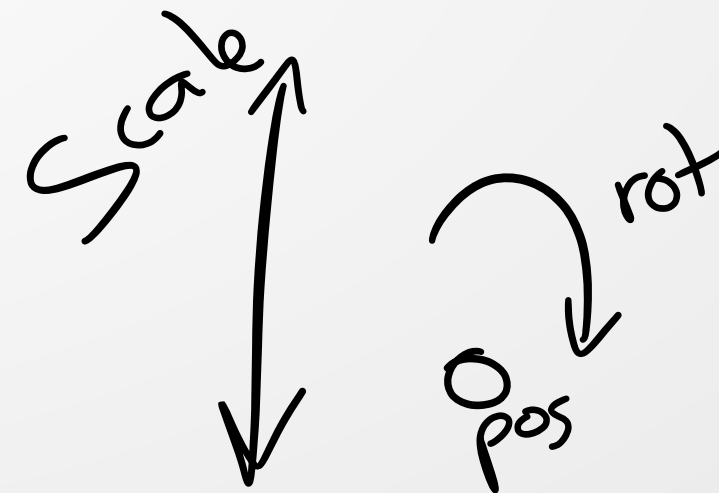
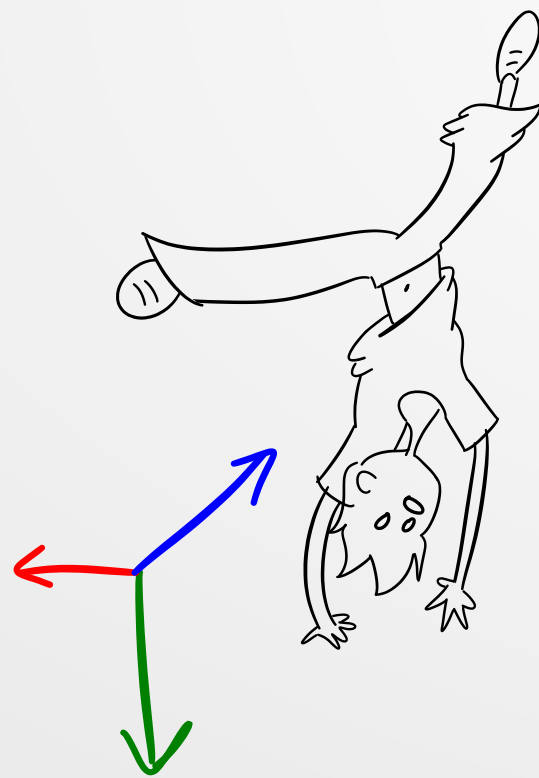


PoV y SCENE

Cómo crear una Matriz 4x4?

`transform.localToWorldMatrix`

`Matrix4x4.TRS(pos, rot, scale)`



P.O.V y SCENE

Cómo crear una Matriz 4x4?

x
► y

y
► x

$x * y$



Matrix 4x4 x

Matrix 4x4 y

$y * x$



Matrix 4x4 y

Matrix 4x4 x

PoV y SCENE

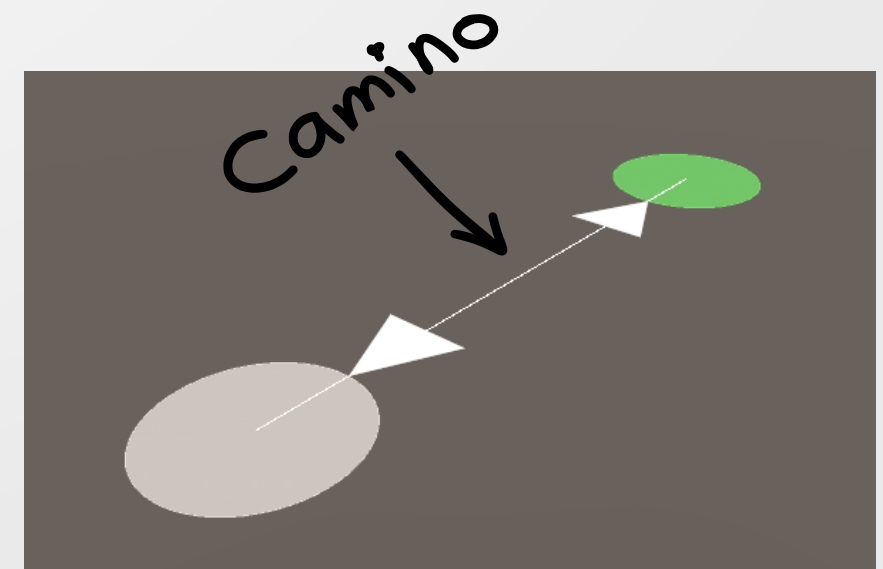
Por qué crear una Matriz 4x4?

```
public class Graph : MonoBehaviour {  
    [SerializeField]  
    Transform _pov = null;  
    public Transform PoV { get { if (!_pov) _pov = transform; return _pov; } }  
  
    public List<Vector3> vertex; ← en coordenadas  
    public List<Edge> edges;      locales  
  
    public Vector3 ToWorldPoint (Vector3 point) {  
        return PoV.TransformPoint(point);  
    }  
  
    public Vector3 ToLocalPoint (Vector3 point) {  
        return PoV.InverseTransformPoint(point);  
    }  
}
```

Dibuja Caminos entre vértices

```
public static void DrawEdges (Graph target) {  
    Matrix4x4 oldMatrix = Handles.matrix;  
    Handles.matrix = target.PoV.localToWorldMatrix;  
  
    for (int i=0; i<target.edges.Count; i++) {  
        foreach (int link in target.edges[i].links) {  
            Handles.DrawLine(target.vertex[link], target.vertex[i]);  
            CoolEditor.ArrowHead(target.vertex[link],  
                                target.vertex[link] - target.vertex[i],  
                                _buttonSize, _buttonSize);  
        }  
    }  
  
    Handles.matrix = oldMatrix;  
}
```

```
Handles.DrawLine(target.transform.position + target.vertex[link],  
                 target.transform.position + target.vertex[i]);
```



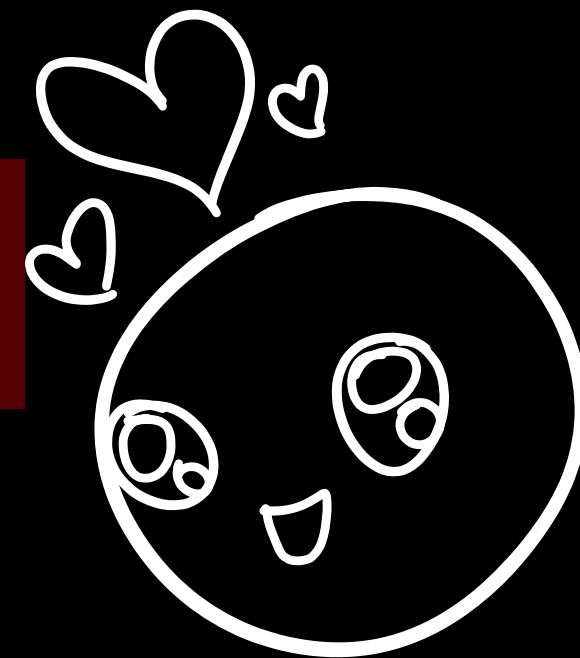
PoV y SCENE

Por qué crear una Matriz 4x4?

```
public static void ArrowHead (Vector3 position, Vector3 direction,
                              float size, float offset = 0) {
    direction.Normalize();
    Matrix4x4 old = Handles.matrix;
    Handles.matrix = old *
        Matrix4x4.TRS(position, Quaternion.LookRotation(direction),
                       Vector3.one);

    Handles.DrawAAConvexPolygon(new Vector3[] {
        new Vector3(0, 0, -offset),
        new Vector3(size/2f, 0, -size - offset),
        new Vector3(-size/2f, 0, -size - offset)
    });

    Handles.matrix = old;
}
```



PoV y SCENE

Devuelve todo a su lugar

```
public static void ArrowHead (Vector3 position, Vector3 direction,
                             float size, float offset = 0) {
    direction.Normalize();

    Matrix4x4 newMatrix = Handles.matrix *
        Matrix4x4.TRS(position, Quaternion.LookRotation(direction),
            Vector3.one);

    using (new Handles.DrawingScope(Handles.color, newMatrix)) {
        Handles.DrawAAConvexPolygon(new Vector3[] {
            new Vector3(0,0, -offset),
            new Vector3(size/2f, 0, -size -offset),
            new Vector3(-size/2f, 0, -size -offset)
        });
    }
}
```

```
public static void ArrowHead (Vector3 position, Vector3 direction,
                             float size, float offset = 0) {
    direction.Normalize();
    Matrix4x4 old = Handles.matrix;
    Handles.matrix = old *
        Matrix4x4.TRS(position, Quaternion.LookRotation(direction),
            Vector3.one);

    Handles.DrawAAConvexPolygon(new Vector3[] {
        new Vector3(0,0, -offset),
        new Vector3(size/2f, 0, -size -offset),
        new Vector3(-size/2f, 0, -size -offset)
    });

    // Handles.matrix = old;
}
```

