# 3D User Interfaces – Tutorial 2 (Video)

Speaker: Linda Rudolph, M.Sc. (Teaching Assistant)

Responsible Professor: Prof. Gudrun Klinker, Ph.D.

Summer Semester 2023

Exam-Relevant

# Topics Today

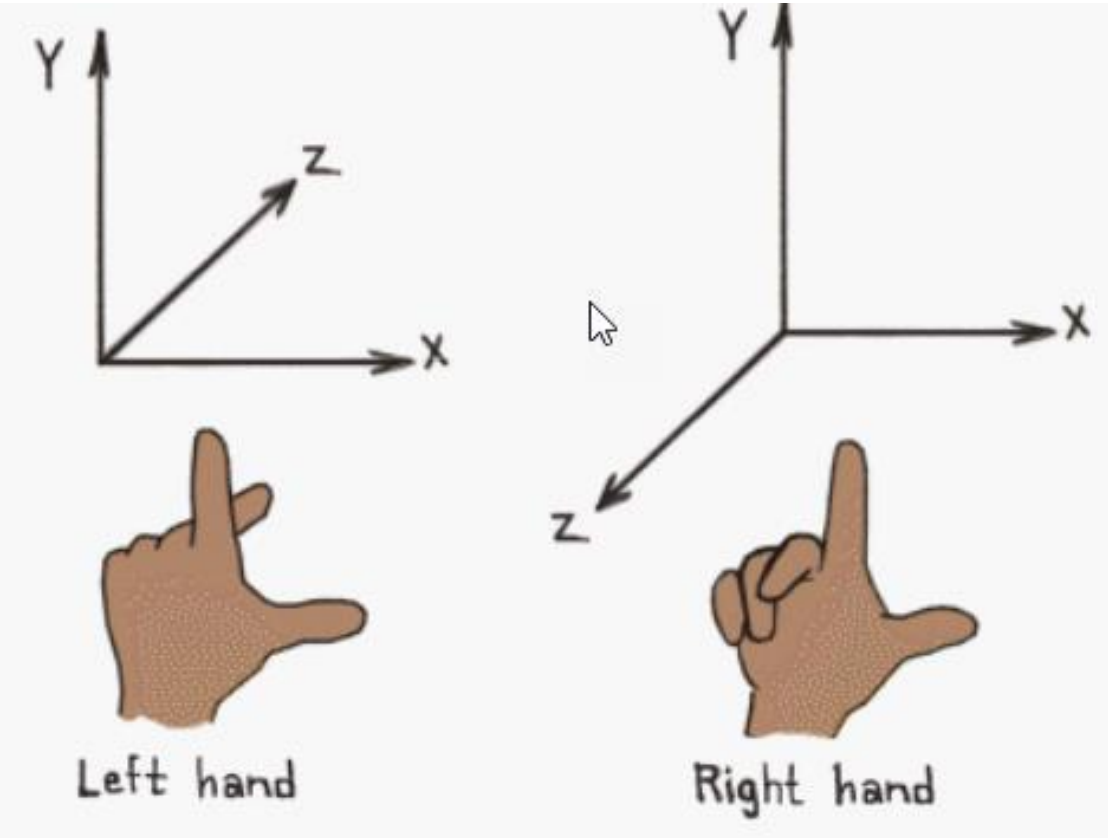
- Quaternions (for practical use)
- Left & right handed coordinate systems
- Observer & Publisher/Subscriber Pattern
- Event Systems in Unity

Exam-Relevant

# Quaternions (for practical use)

- unit quaternions can represent rotations
- Are typically stored as a vector (x, y, z) and a scalar w
- Attention! Quaternions have their own calculus,
  you cannot multiply them with the same operations as vectors or scalars
- Allow for smooth interpolations between orientations (SLERP)
- There are conversion formulars to other rotation representations
  (euler angles, axis/angle, 3x3 Matrix)
- Less processing heavy and more precise then rotation matrices
- „4-dimensional extension of imaginary numbers“

# Left- and Right-Handed Coordinate Systems




Source: https://www.oreilly.com/library/view/learn-arcore/9781788830409/03e5338d-02f1-4461-a57a-ef46a976f96b.xhtml

Exam-Relevant

# Changing the basis of a coordinate system

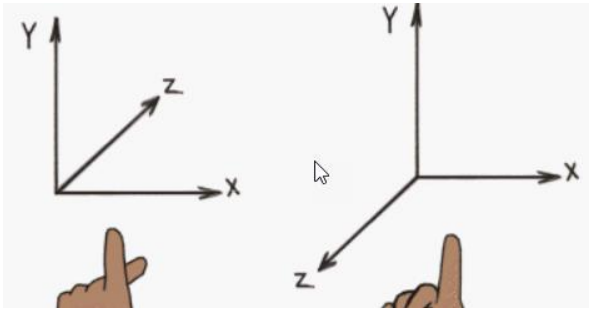
- ~~„just“ multiplying on axis by -1~~ works for positions, but not orientations

(1)

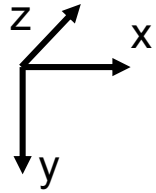
Correct solution (similarity transform)

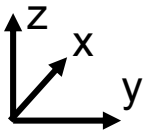
B = P * A * inverse(P)

$$P_{(1)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$




A … Input Pose Matrix (4x4)
B … Output Pose Matrix (4x4)
P … Correction Matrix (depends on how the coordinate systems differ)




Source and further explanations:
https://www.dariomazzanti.com/uncategorized/change-of-basis/

https://towardsdatascience.com/change-of-basis-3909ef4bed43

(2)

$$P_{(2)} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

# Left- and Right- Handed Coordinate Systems


Source: https://ahmetburul.medium.com/coordinate-systems-of-3d-applications-guide-ddfa2194ed88

Exam-Relevant

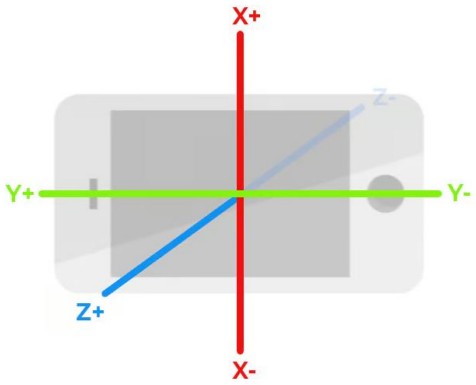
# Exercise: Think about it …

**Using slide 5 & 6**

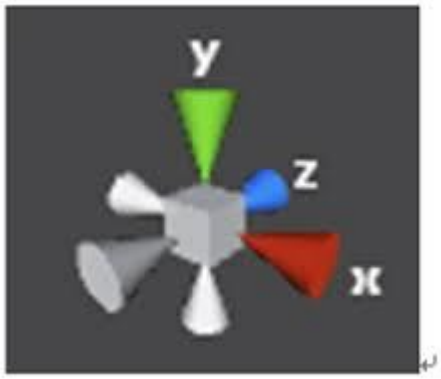
P(1) represents the conversion from which software tools to which?

For which software can you use P(2)?

The blog post for Homework 1 shows the following coordinate system for smartphones. Which correction matrix P(3) needs to be applied to give the pose in Unity Coordinate systems?*

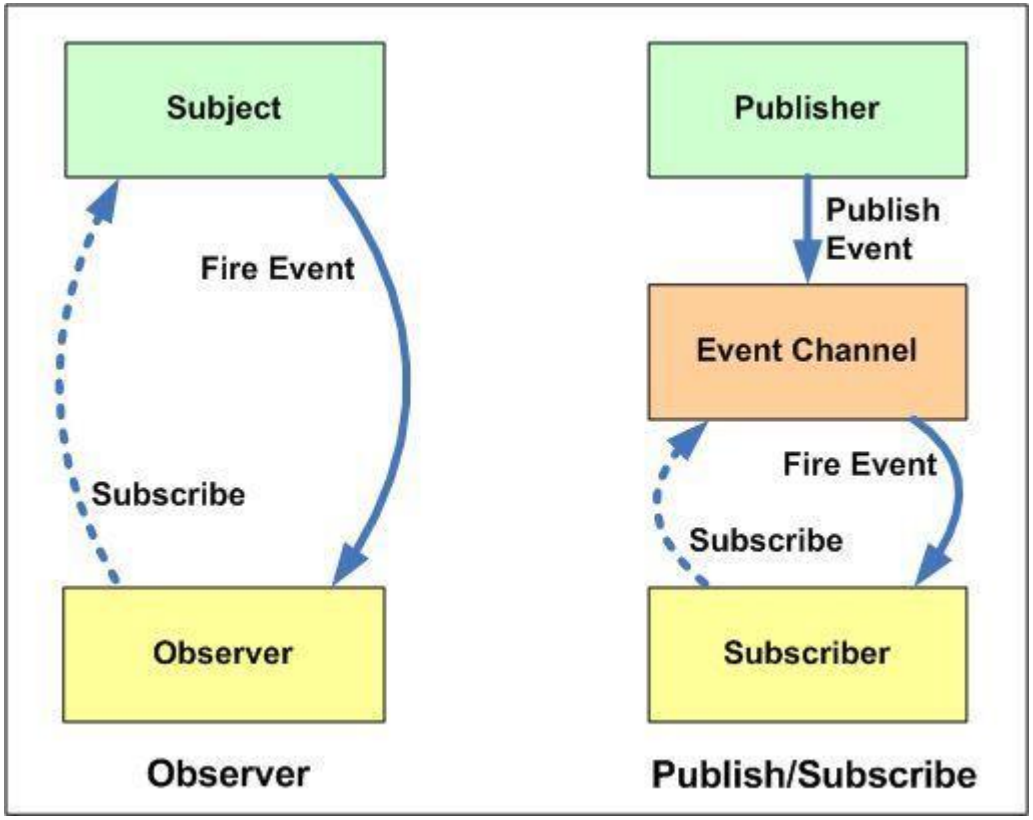



P(3)?




*= If the graphic in the blog post is meant that way, this transformation is done in Unity internally! You do not need to perform this operation manually in homework 1.

# Pub/Sub and Observer Patterns




Equivalent? -> Ongoing controversy

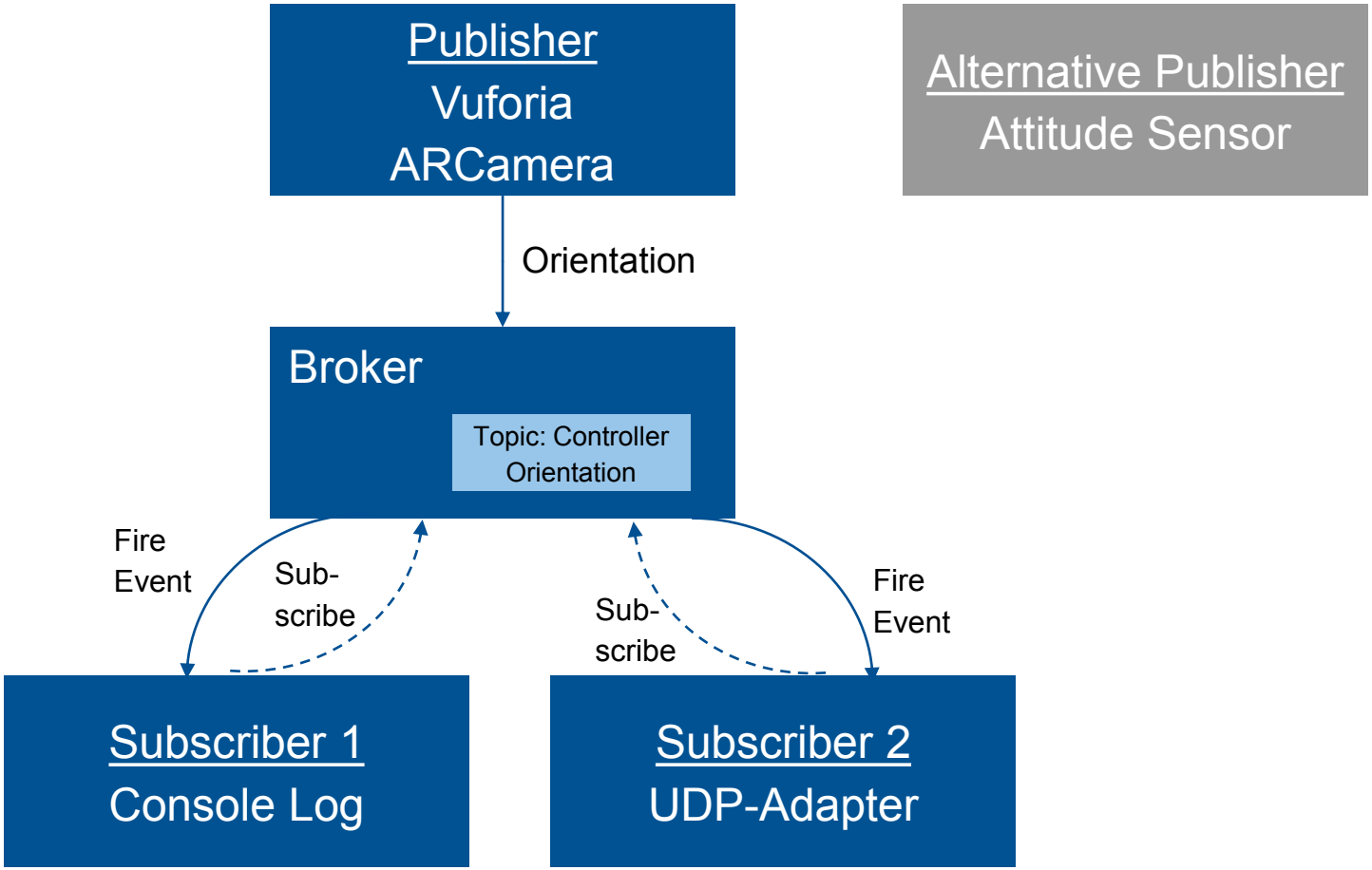
<u>If you want to differentiate</u>

Observer
- Tightly coupled
- Synchronous implementation

Pub/Sub
- Publisher & Subscriber do not know each other
- Broker/Event Channels
- Asynchronous

https://hackernoon.com/observer-vs-pub-sub-pattern-50d3b27f838c

# Pub/Sub in the homework



# Events in Unity

Exam-Relevant

**Version 1 (High-Level): Unity Events**

Can be used via Unity Inspector View

- Better for collaborative work with designers

```
using UnityEngine;
using UnityEngine.Events;

[Serializable]
public class FloatEvent : UnityEvent <float> { }

public class PlayerHealth : MonoBehaviour
{
    public FloatEvent onPlayerHurt;
    public void TakeDamage(float damage)
    {
        onPlayerHurt.Invoke(damage);
    }
}
```


Default UnityEvent =>

<= custom (with dynamic parameters)

```
using UnityEngine;
using UnityEngine.Events;
public class PlayerHealth : MonoBehaviour
{
    float health=100;
    public UnityEvent onPlayerDeath;
    public void TakeDamage(float damage)
    {
        health -= damage;
        if(health < 0)
        {
            onPlayerDeath.Invoke();
        }
    }
}
```


Source & Further information:
https://gamedevbeginner.com/events-and-delegates-in-unity/#:~:text=Events%20in%20Unity%20are%20a,from%20within%20their%20own%20class

Exam-Relevant

# Events in Unity

**Version 2 (mid-level): C# events & Actions**

- Specialized C# delegates for event systems
- Delegates => C# Data container for functions (similar to Java lambda expressions)
- Actions => specific delegates, convenient for usage with event declarations

- If no class subscribes to an event, a Nullpointer-Exception can be triggered
  - Therefore OnGameOver?.Invoke()

- Not scene-related, can be used “game-wide” (esp. if combined with Singletons)

```
using System;

public class PlayerHealth : MonoBehaviour
{
    public static event Action<string> OnGameOver;
    public void TakeDamage(float damage)
    {
        health -= damage;
        if(health < 0)
        {
            OnGameOver?.Invoke("The game is over");
        }
    }
}


public class AnotherClass : MonoBehaviour
{
    void RestartGame()
    {
        // Restart the game!
    }
    private void OnEnable()
    {
        PlayerHealth.onGameOver += RestartGame;
    }
}
```

Source & Further information:
https://gamedevbeginner.com/events-and-delegates-in-unity/#:~:text=Events%20in%20Unity%20are%20a,from%20within%20their%20own%20class