

# Modul 24

## Objektorientierte Programmierung



VITRUVIUS  
HOCHSCHULE

Oliver Ziegler

# Zeit und Timer

- **Zeit als Kommazahl**
- **Framerate**
- **Time.deltaTime**
- **Blaupause eines Timers**
- **Einfachere Wege: Invoke**

# Zeit als Kommazahl

- Es gibt die Möglichkeit Zeiten festzuhalten
- Das ist jedoch für einen Timer zu aufwändig
- Man verwendet float-Zahlen, die die Sekunden beschreiben
  
- Um Zeit mitzuzählen, zählt man also eine float-Variable hoch
- Doch um was?

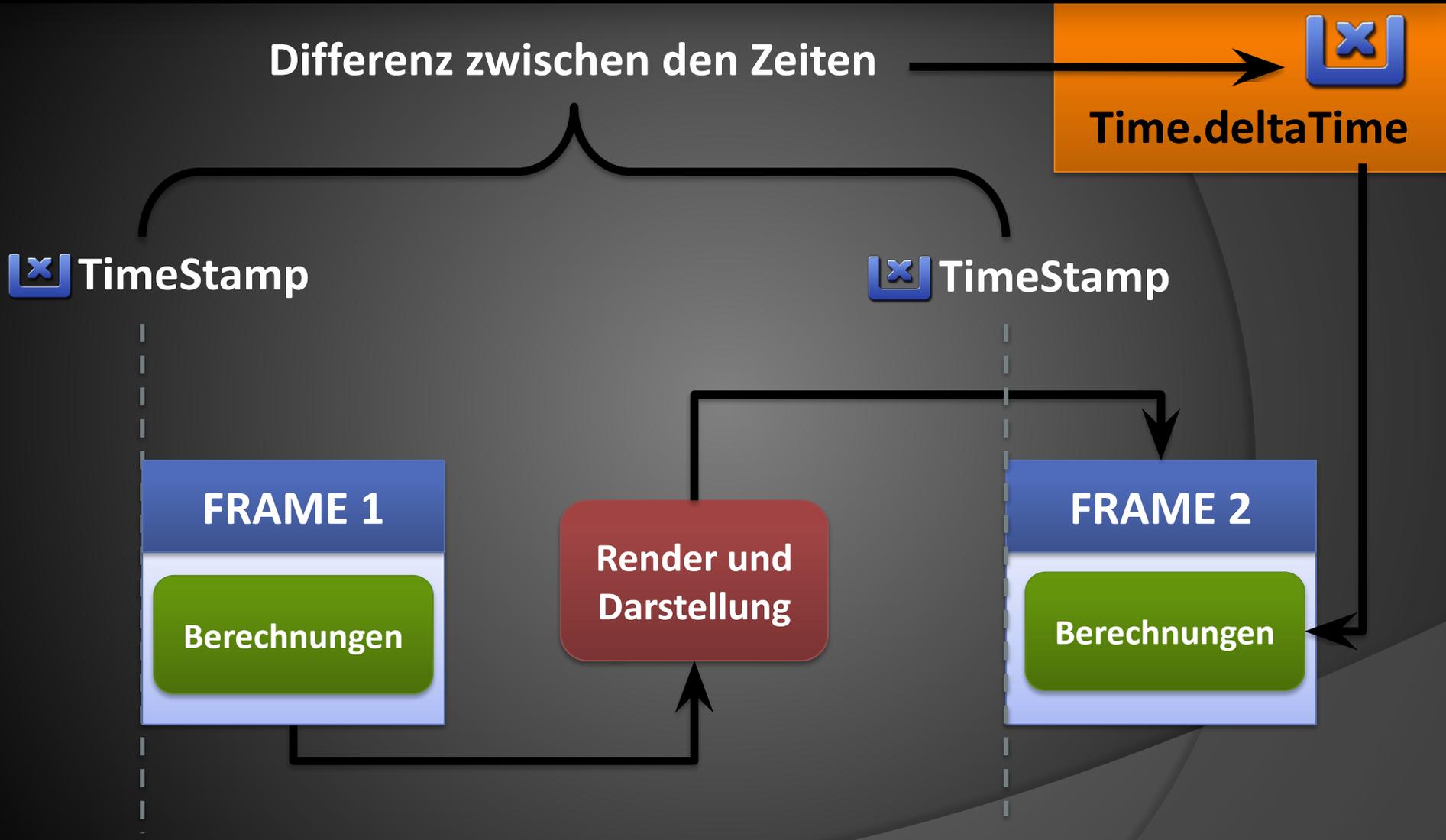
# Framerate

- Anfängerfehler ist es, in jedem Frame um einen Festen Wert zu zählen
- Ergebnis:
  - zeitverzögerte Funktion
  - Aber keine korrekte Zeit
- Framerate unterschiedlich, von Hardware und Spiel abhängig

# Time.DeltaTime

- Der gesuchte Wert
- Die Zeit, die zwischen dem letzten und dem jetzigen Frame verstrichen ist
- Sehr präzise, in Millisekunden

# Time.DeltaTime



# Zeit mitzählen

```
public float zeitBisher;  
  
public void Update()  
{  
    zeitBisher += Time.deltaTime;    // Zählt den Wert um die Sekunden hoch  
}
```

# Zeit mitzählen

```
public float zeitBisher;

public void Update()
{
    zeitBisher += Time.deltaTime; // Zählt den Wert um die Sekunden hoch

    float minutesRaw = zeitBisher / 60; // Zeit in Sekunden / 60 = Minuten

    float minutes = Mathf.Floor( minutesRaw ); // Rundet die Zahl nach unten ab

    float seconds = Mathf.RoundToInt( zeitBisher % 60 );
        // Rest von „zeitBisher / 60“ □ Gerundet

    Debug.Log(minutes + „:“ + seconds);

    // Beispiel: 5 : 23
}
```

# Timer

```
public float timer;

Public void StartSomethingInSomeTime()
{   timer = 4.6f;   }

public void Update()
{
    if( timer > 0)           // Nur zählen, wenn Timer gestartet
    {
        timer -= Time.deltaTime;    // Runterzählen
        if(timer <= 0)           // Zeitabgelaufen?
        {
            TimerFunction(); //Aktion!
            timer = -1;        // Timer stoppen
        }
    }
}
```

# Invoke

- Ruft eine spezielle Funktion nach einer speziellen Zeit auf
- Kein eigener Timer notwendig!
- Gut für nicht abbrechbare Funktionen, deren Timer nirgends gezeigt wird

```
public void JetztUndSpäter()
{
    DoStuffNow();           // Löst die Funktion „DoStuffNow“ sofort aus

    Invoke(„DoStuffLater“ , 3.4f);
        // Löst die Funktion „DoStuffLater“ in 3,4 Sekunden aus
}

public void DoStuffLater() { }
```