

Modul 24

Objektorientierte Programmierung



VITRUVIUS
HOCHSCHULE

Oliver Ziegler

Eingabe (Tastatur und Maus)

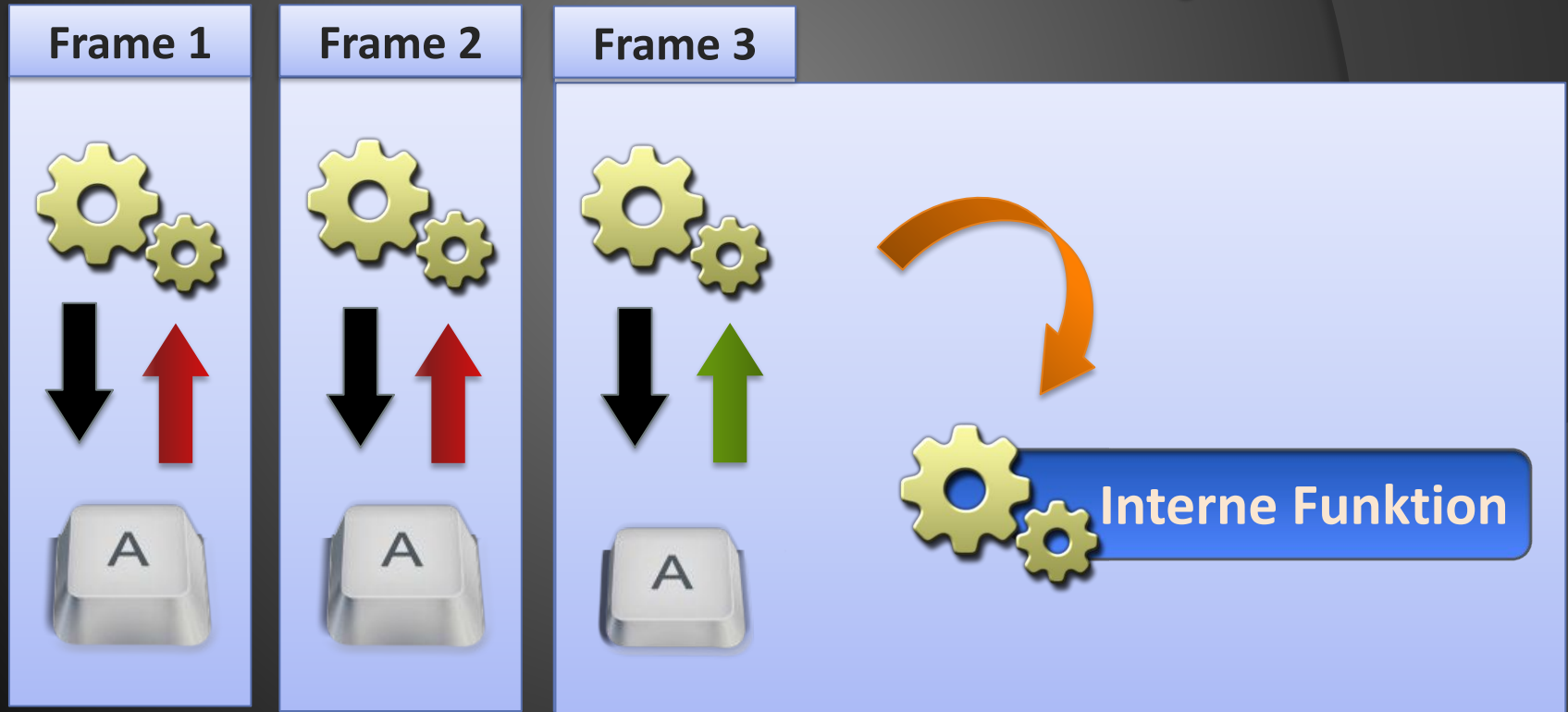
- **Keyboard – Abfrage**
- **Maus - Buttons**

Buttons/Keys

- **Buttons der Maus und Tastatur haben Indices (Zahlen) über die sie abgefragt werden**
- **Buttons haben Zustände die einzelne abgefragt werden können (Down, Pressed, Up)**
- **Meist in der Update, um den Moment des Drückens nicht zu verpassen**

Funktion vom System

- Automatische Funktionen
- Beispiel Tastaturabfrage



Button States

DOWN

GetButtonDown



Einmalig
Wenn nach unten
gedrückt

PRESSED

GetButton



Jeden Frame
Während
gedrückt

UP

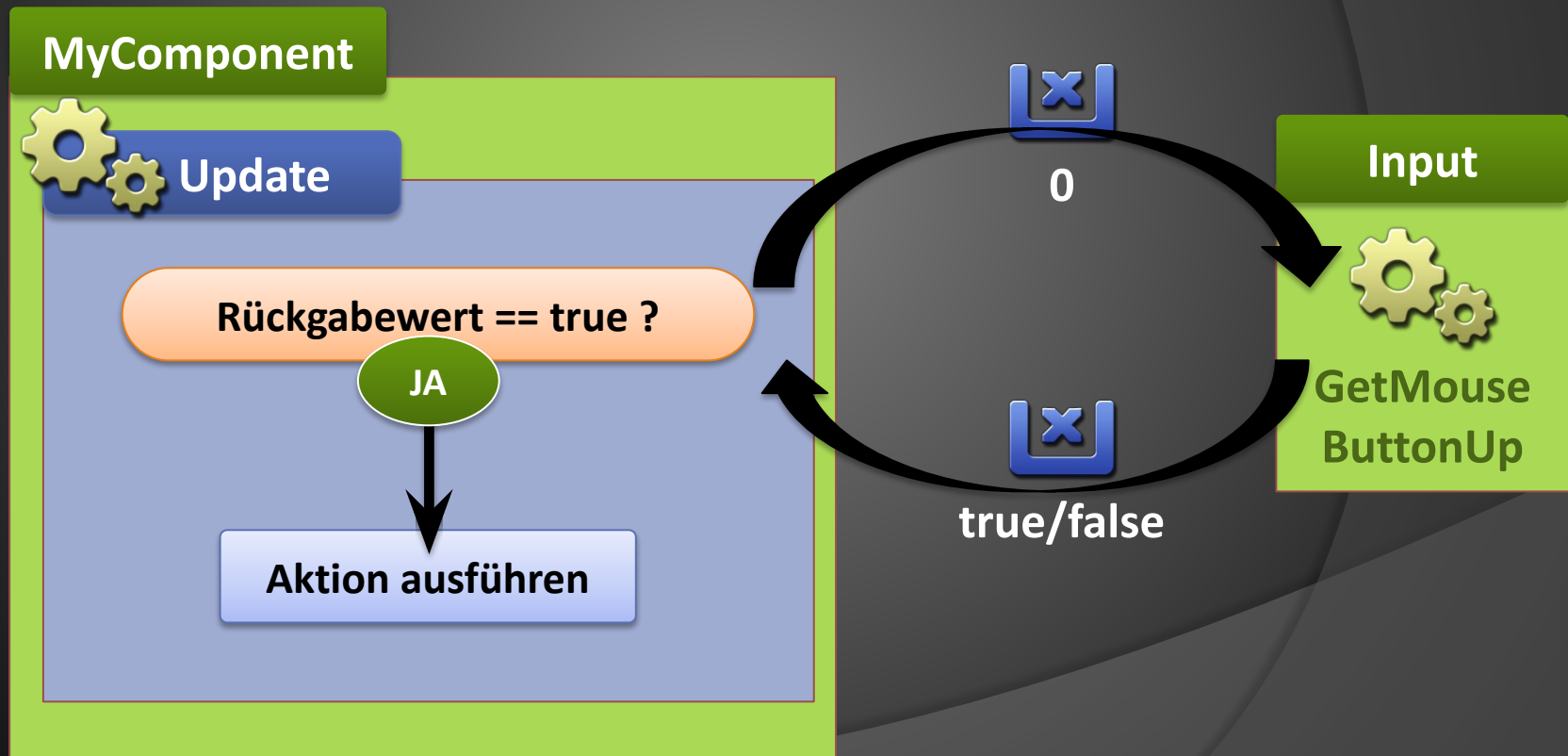
GetButtonUp



Einmalig
Wenn Taste
losgelassen

Maustasten

- Linke Maustaste = 0
- Rechte Maustaste = 1
- ...



Button States

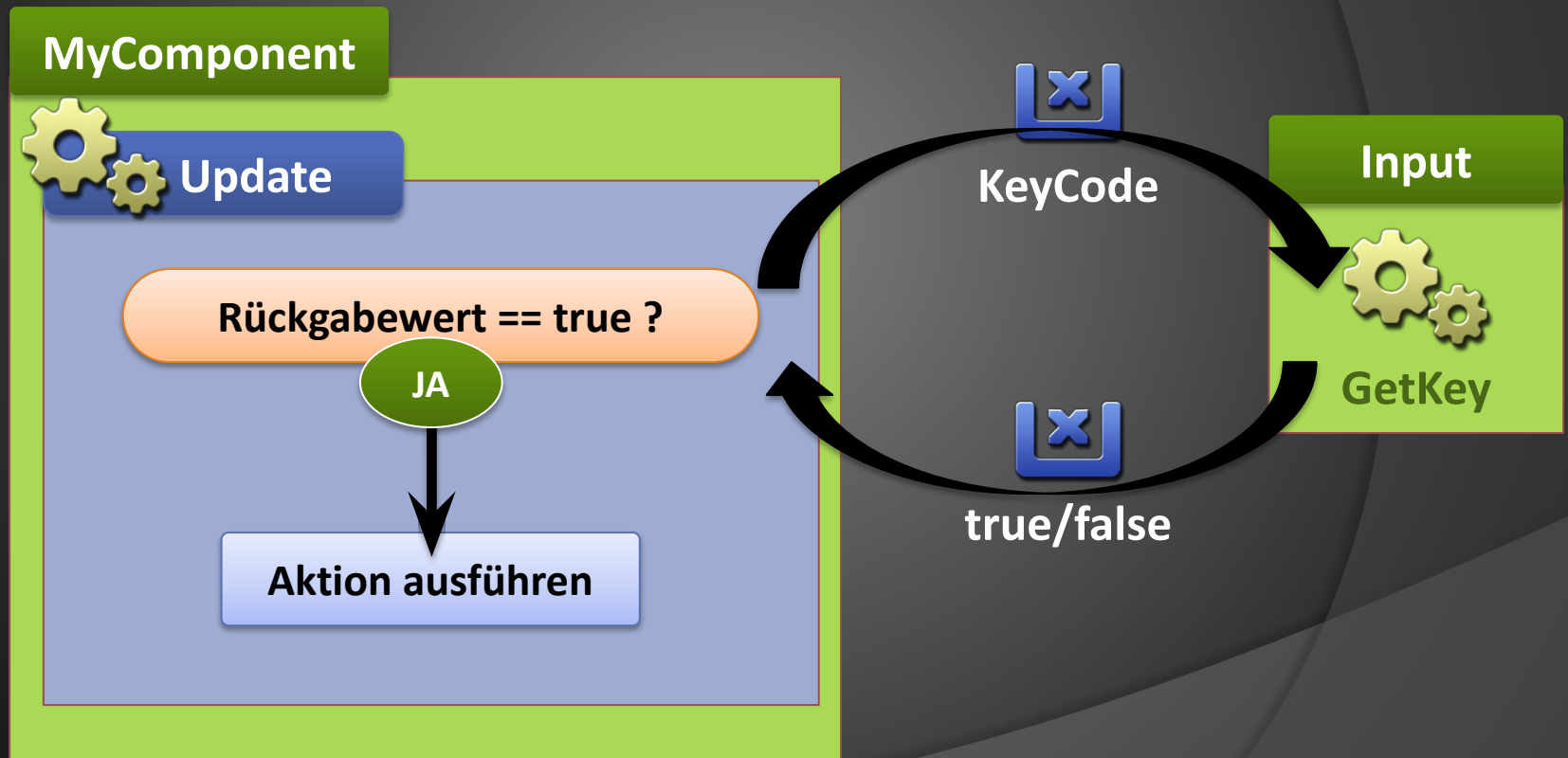
- Im Code wird nicht der State eines bestimmten Buttons abgefragt
(Rückgabewert : Zustand)
- Sondern, ob der Button einen bestimmten State hat
(Rückgabewert : Wahr oder falsch)

```
if( Input.GetMouseButtonDown( 0 ) )           // 0 = linke Maustaste  
{ Tue_Etwas_Beim_Herunterdrücken(); }
```

```
if( Input.GetMouseButton ( 0 ) )  
{ Tue_Etwas_Immer_Wieder_Während_Gedrückt(); }
```

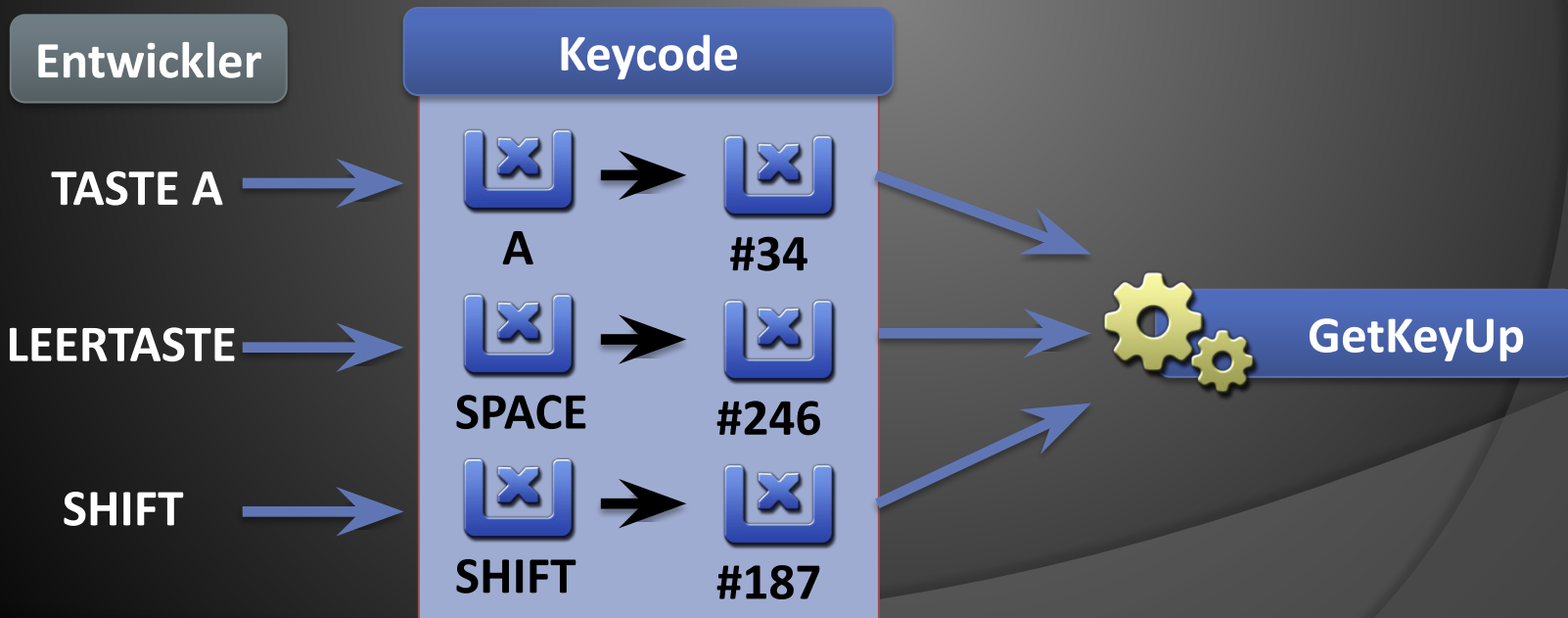
```
if( Input.GetMouseButtonUp( 0 ) )  
{ Tue_Etwas_Beim_Loslassen(); }
```

Tastatur



KeyCode

- Ein Enumerator (Enum), der auf Zahlen verweist
- Erhöht die Lesbarkeit



Button States

- Im Code wird nicht der State eines bestimmten Buttons abgefragt
(Rückgabewert : Zustand)
- Sondern, ob der Button einen bestimmten State hat
(Rückgabewert : Wahr oder falsch)

```
if( Input.GetKeyDown( KeyCode.LeftArrow ) )
```

```
{ Tue_Etwas_Beim_Herunterdrücken(); }
```

```
if(Input.GetKey ( KeyCode.LeftArrow ) )
```

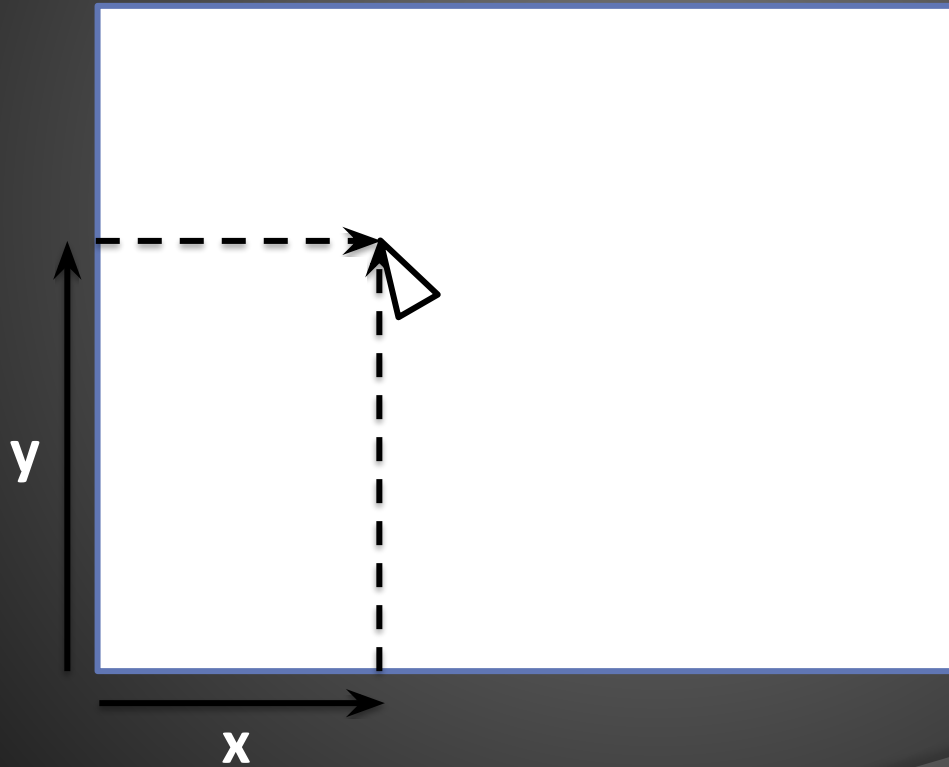
```
{ Tue_Etwas_Immer_Wieder_Während_Gedrückt(); }
```

```
if(Input.GetKeyUp( KeyCode.LeftArrow ) )
```

```
{ Tue_Etwas_Beim_Loslassen(); }
```

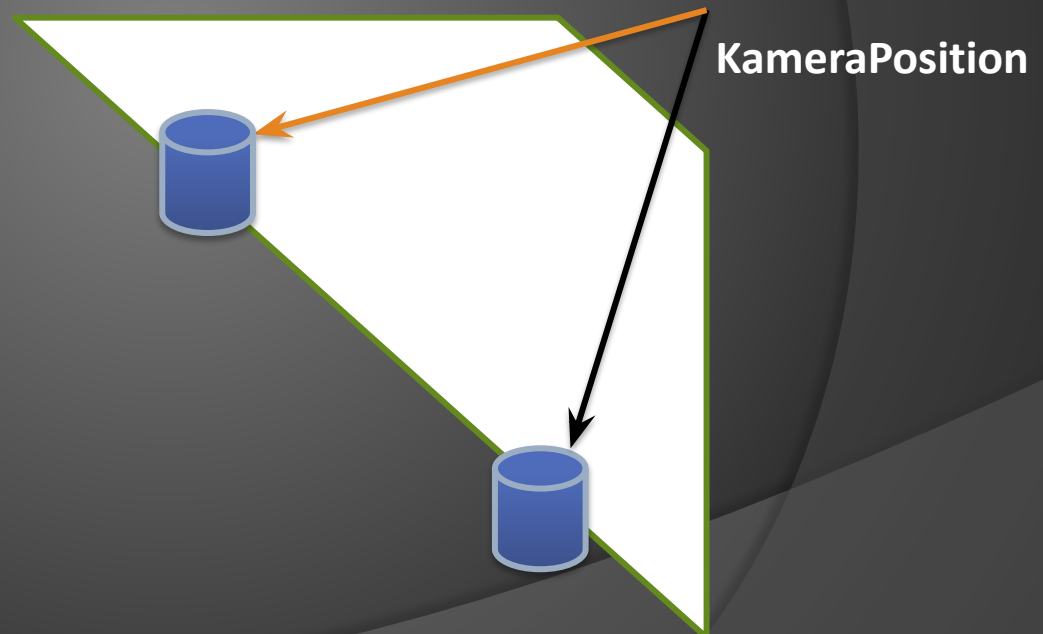
Maus - Position

- Position des Mauszeigers auf dem Screen (in Pixeln)



Maus-Interaktion

- Via Raycasting (Siehe Folie Raycasting)
- Häufigstes Raycasting
 - Interaktion in 2D □ Projektion in 3D
- Ausgangspunkt: Maus - Position



Maus-Position □ Raycast

- Kameras haben diese Funktion bereits
 - Erstellen eines Strahls anhand einer 2D-Pos.

```
public void Update()
{
    Ray charles = Camera.main.ScreenpointToRay(Input.mousePosition);
    //Erzeugt einen Strahl von der Mausposition auf dem Kamerabild in Richtung der
    Kamera
    if(Physics.Raycast(charles))
    {
        // Maus ist über... irgendetwas
    }
}
```

Maus-Position □ Raycast

- Kann wie jeder Strahl verwendet werden

```
public void Update()
{
    Ray charles = Camera.main.ScreenpointToRay(Input.mousePosition);
    RaycastHit hitInfo;
    //Erzeugt einen Strahl von der Mausposition auf dem Kamerabild in Richtung der
    Kamera
    if(Physics.Raycast(charles, out hitInfo))
    {
        Debug.Log(hitInfo.collider.gameObject.name);
        // Mausstrahl trifft den Collider „hitInfo.collider“
        // Das entsprechende Objekt ist „hitInfo.collider.gameObject“
    }
}
```