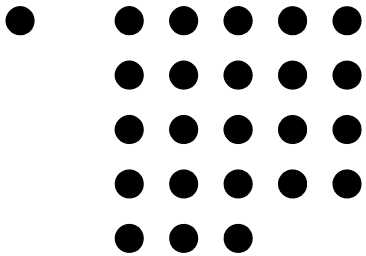
Android:
Property Animation
– Techniken für Fortgeschrittene –

Prof. Dr. Carsten Vogt



Fachhochschule Köln
Cologne University of Applied Sciences

Android: Property Animation

1.) Einfache und komplexe Techniken

2.) TypeEvaluator

3.) TimeInterpolator

4.) ValueAnimator

5.) Weitere Informationen

Fachhochschule Köln
Cologne University of
Applied Sciences

Einfache Techniken

- bisher: einfache Techniken zur Animation
  - mit PropertyAnimator:
    - ```
      myView.animate().
            x(targetX).y(targetY).etc.
                setDuration(...).start();
      ```
  - mit ObjectAnimator:
    - ```
      ObjectAnimator anim =
            ObjectAnimator.ofFloat(
                        myView,"x",targetX);
      anim.setDuration(...);
      anim.start();
      ```

Fachhochschule Köln
Cologne University of
Applied Sciences

Einfache Techniken

- Details im vorherigen Video
  und zugehörigen Programmcode
  - *www.nt.fh.koeln.de/vogt/vma/videos.html*
- Vorteil:
  relativ geringer Programmieraufwand
- Nachteil:
  eingeschränkte Flexibilität

Fachhochschule Köln
Cologne University of
Applied Sciences

Komplexere Techniken

- jetzt: komplexere Techniken
  - mehr Programmieraufwand
  - große Flexibilität
- grundlegende Klassen und Interfaces:
  - `TypeEvaluator`:
    Berechnung der animierten Property-Werte
  - `TimeInterpolator`:
    Festlegung des Zeitverhaltens
  - `ValueAnimator`:
    Definition und Steuerung der Animation

Fachhochschule Köln
Cologne University of
Applied Sciences

Android: Property Animation

1.) Einfache und komplexe Techniken

2.) TypeEvaluator

3.) TimeInterpolator

4.) ValueAnimator

5.) Weitere Informationen

Fachhochschule Köln
Cologne University of
Applied Sciences

TypeEvaluator

- **TypeEvaluator**:
  - berechnet aktuelle Property-Werte im Verlauf der Animation
  - z.B. x-Positionen eines animierten Views
- **Berechnung**:
  - üblicherweise **linear**
  - Parameter: aktueller Zeitpunkt im Ablauf
    - als “elapsed fraction of the animation”
    - also $0.0 \leq \text{fraction} \leq 1.0$
  - z.B. xAktuell = xStart + fraction * (xEnd-xStart)

Fachhochschule Köln
Cologne University of Applied Sciences

TypeEvaluator

- Interface `TypeEvaluator<T>`
  - `<T>`: Typ der animierten Property
- zu implementierende Methode:

```
T evaluate(float fraction,
                  T startValue, T endValue)
```

  - Parameter:
    - `fraction`: aktuelle Zeitpunkt im Ablauf
    - `startValue`: Anfangswert der Property
    - `endValue`: Endwert der Property
  - Resultat: aktueller Wert der Property

Fachhochschule Köln
Cologne University of
Applied Sciences

TypeEvaluator

- **Implementationen des Interfaces:**
  - `IntEvaluator`, `FloatEvaluator`
  - `IntArray/FloatArrayEvaluator`
  - `ArgbEvaluator`
  - `PointFEvaluator`, `RectEvaluator`
  - **selbstgeschriebene Klassen**
- **Verwendung bei Erzeugung von Animatoren:**
  -
```
ObjectAnimator.ofObject(
    Object target, String propertyName,
    TypeEvaluator evaluator, ...)
```

Fachhochschule Köln
Cologne University of
Applied Sciences

Android: Property Animation

1.) Einfache und komplexe Techniken

2.) TypeEvaluator

3.) TimeInterpolator

4.) ValueAnimator

5.) Weitere Informationen

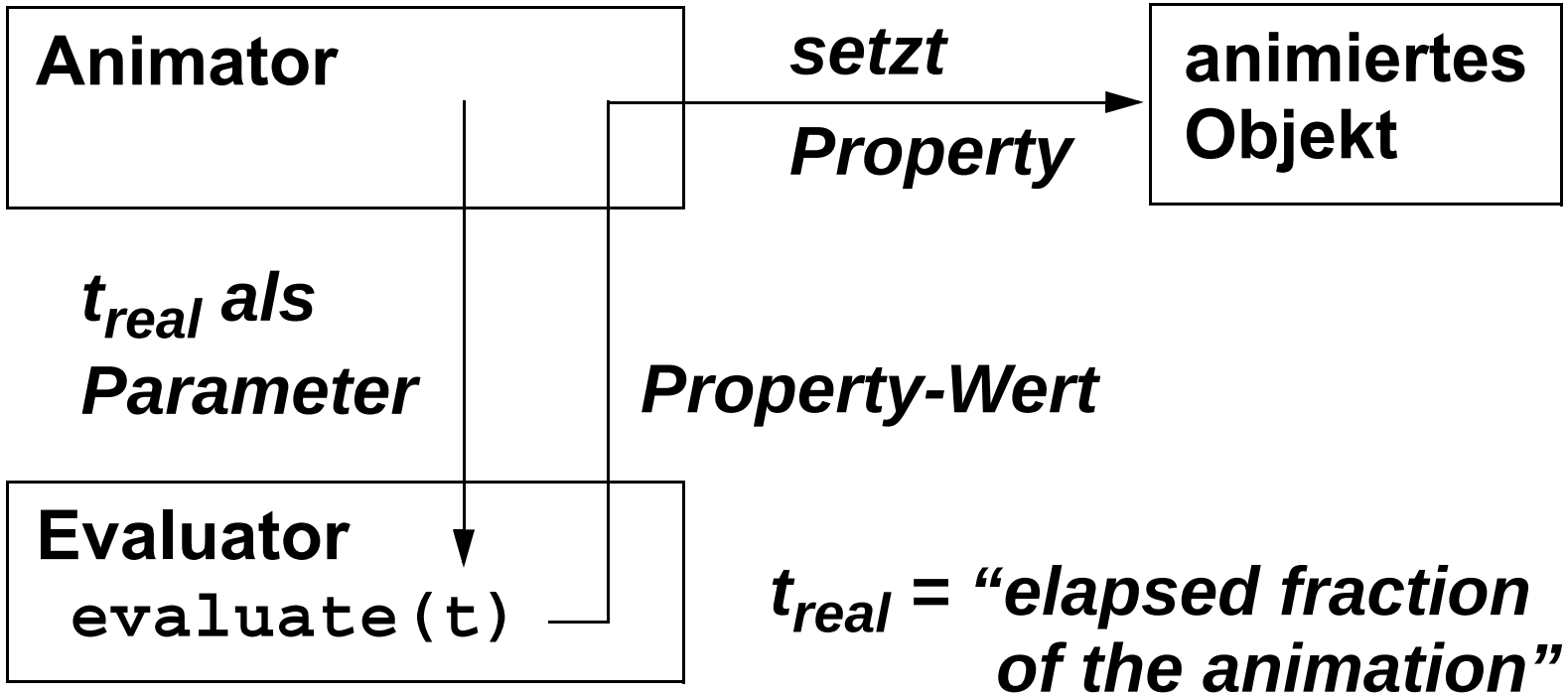
Fachhochschule Köln
Cologne University of
Applied Sciences

TimeInterpolator

- spezifiziert das Zeitverhalten der Animation
- im vorherigen Video:
  Nutzung vordefinierter TimeInterpolatoren
  - `AccelerateInterpolator`
  - `OvershootInterpolator`
  - `...`
- hier:
  Programmierung eigener TimeInterpolatoren

TimeInterpolator

- Ablauf ohne TimeInterpolator:

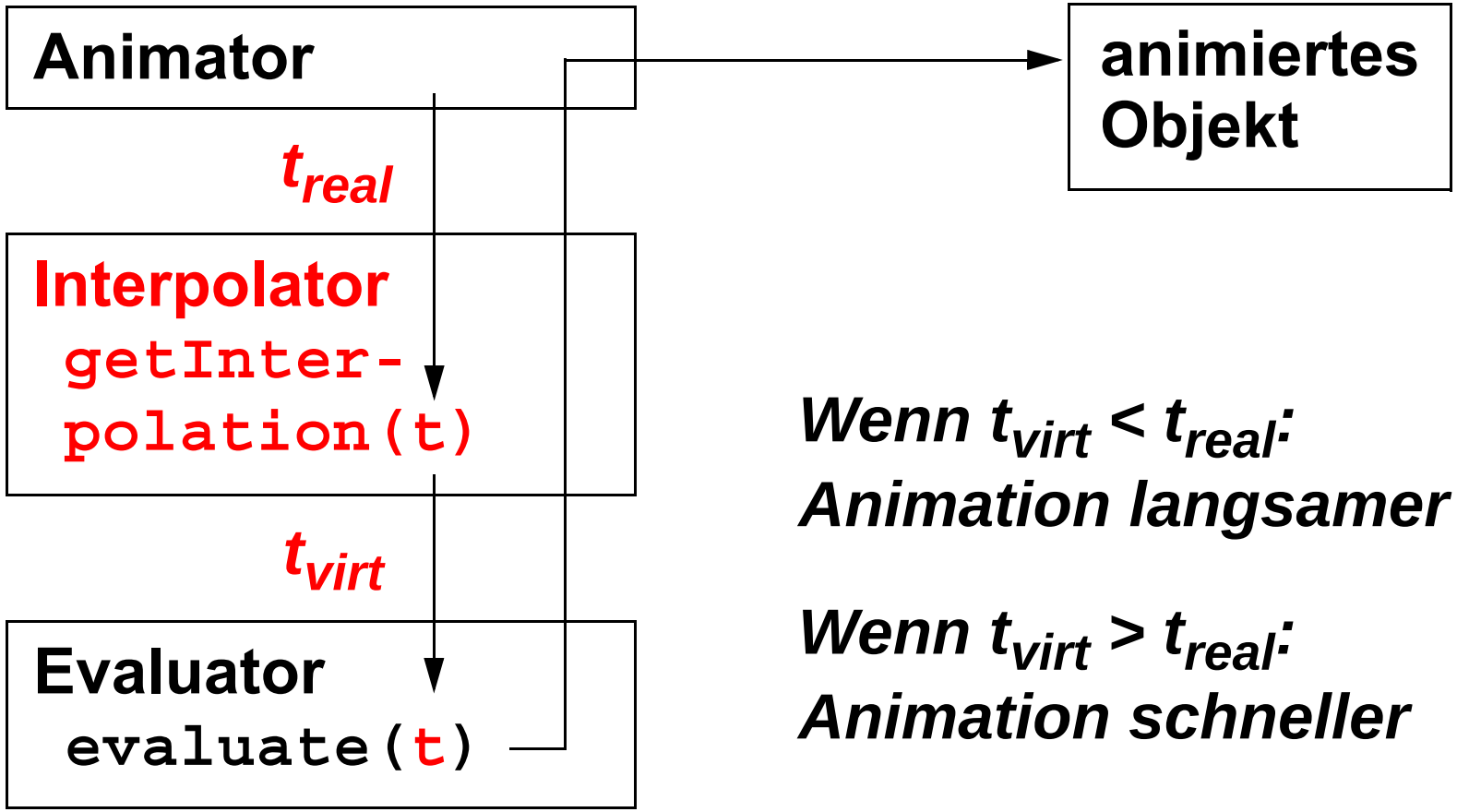


- `evaluate()`: (üblicherweise) linear
- damit: konstante Geschwindigkeit

Fachhochschule Köln
Cologne University of Applied Sciences

TimeInterpolator

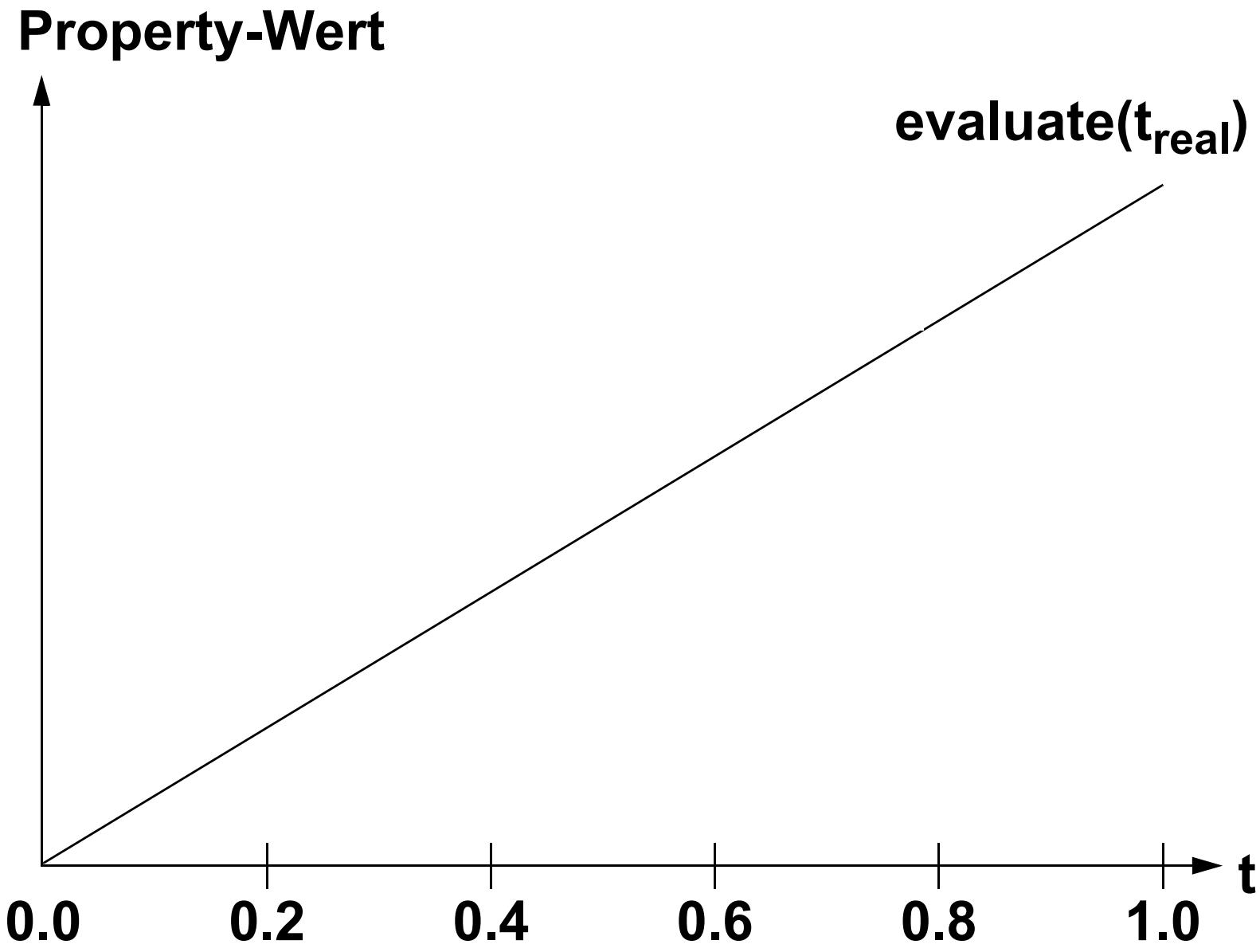
- Ablauf mit TimeInterpolator:



Wenn $t_{virt} < t_{real}$: Animation langsamer

Wenn $t_{virt} > t_{real}$: Animation schneller

TimeInterpolator

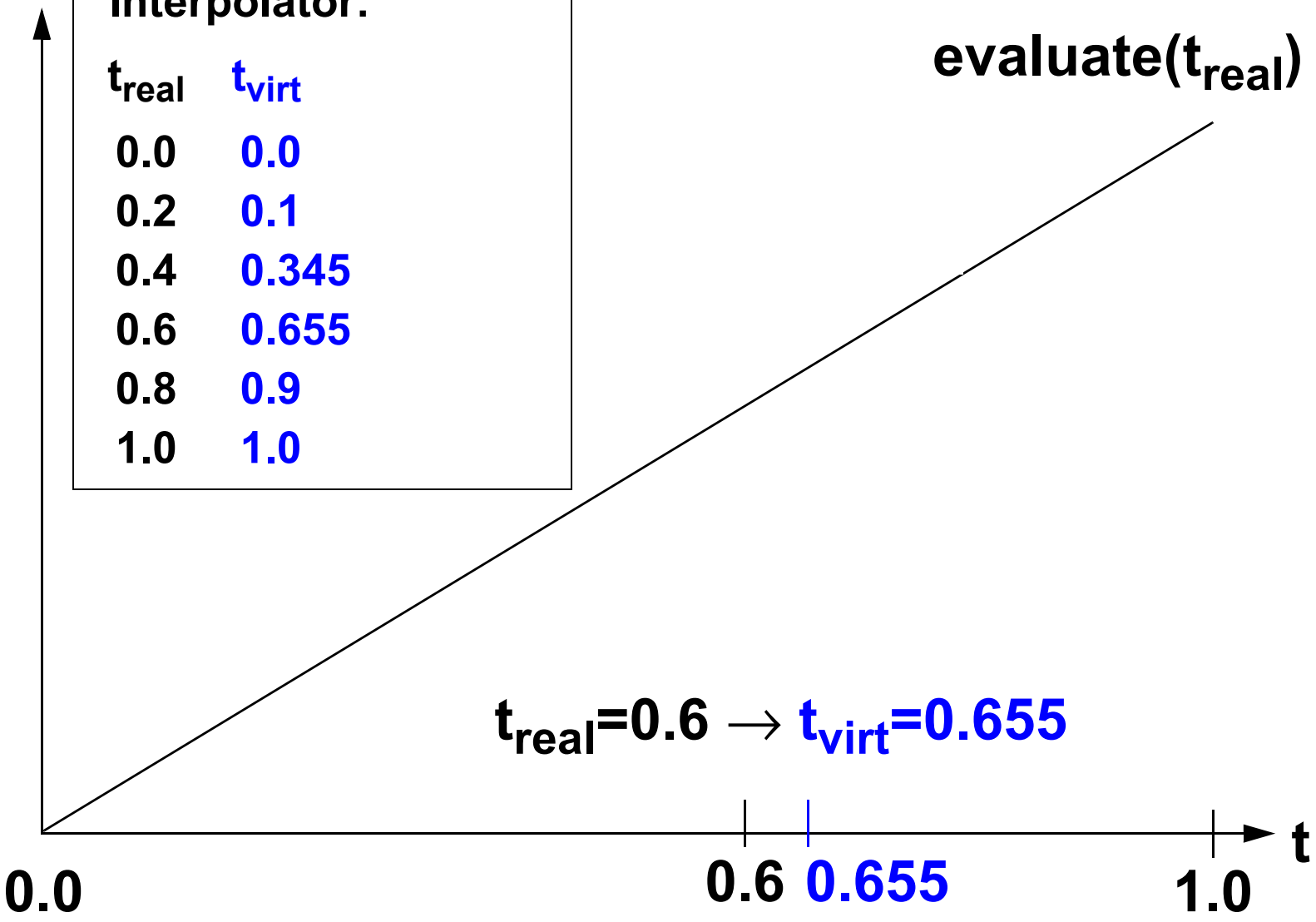


Fachhochschule Köln
Cologne University of
Applied Sciences

TimeInterpolator

AccelerateDecelerate Interpolator:

t_{real}	t_{virt}
0.0	0.0
0.2	0.1
0.4	0.345
0.6	0.655
0.8	0.9
1.0	1.0

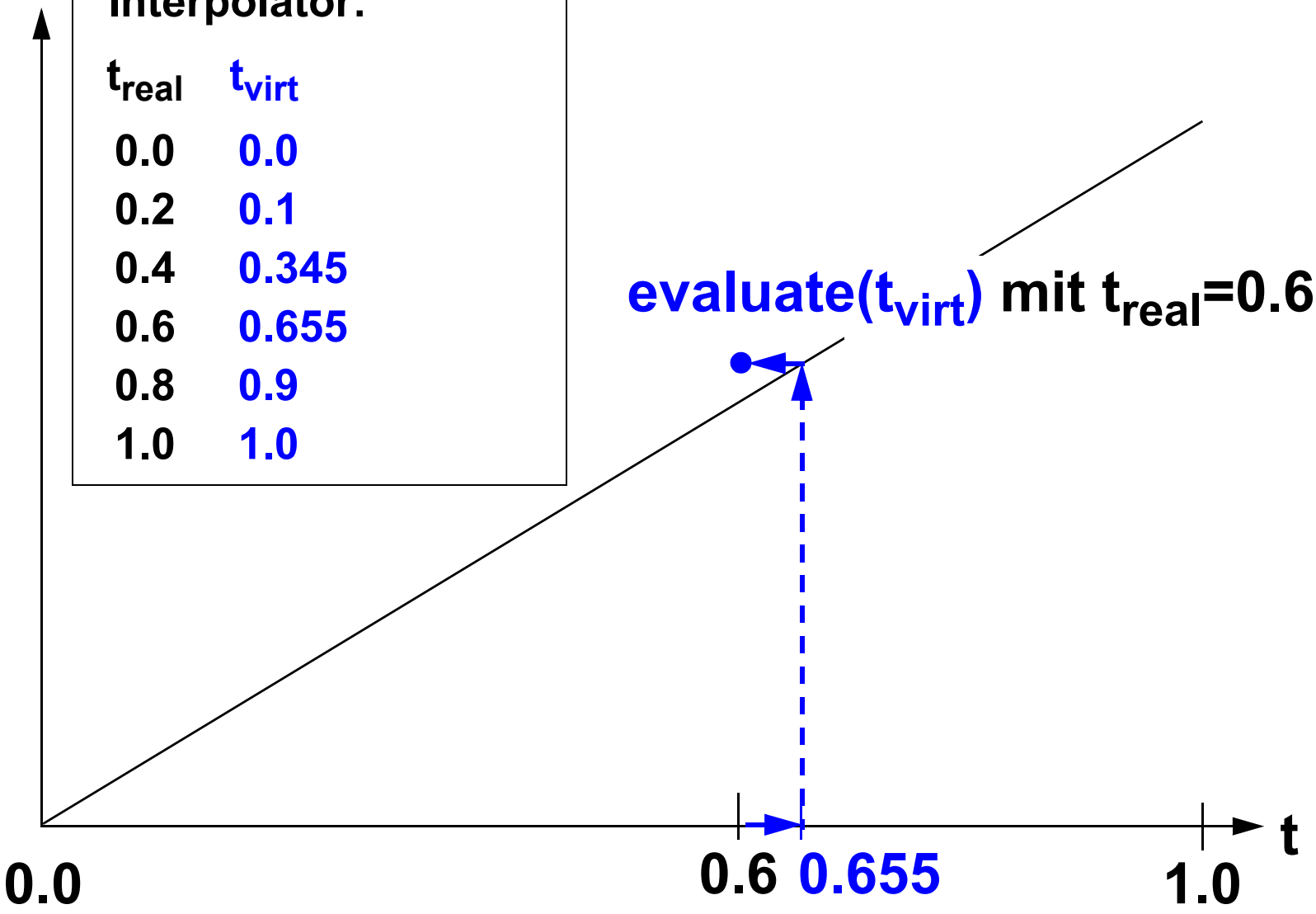


Fachhochschule Köln
Cologne University of Applied Sciences

TimeInterpolator

AccelerateDecelerate Interpolator:

t_{real}	t_{virt}
0.0	0.0
0.2	0.1
0.4	0.345
0.6	0.655
0.8	0.9
1.0	1.0

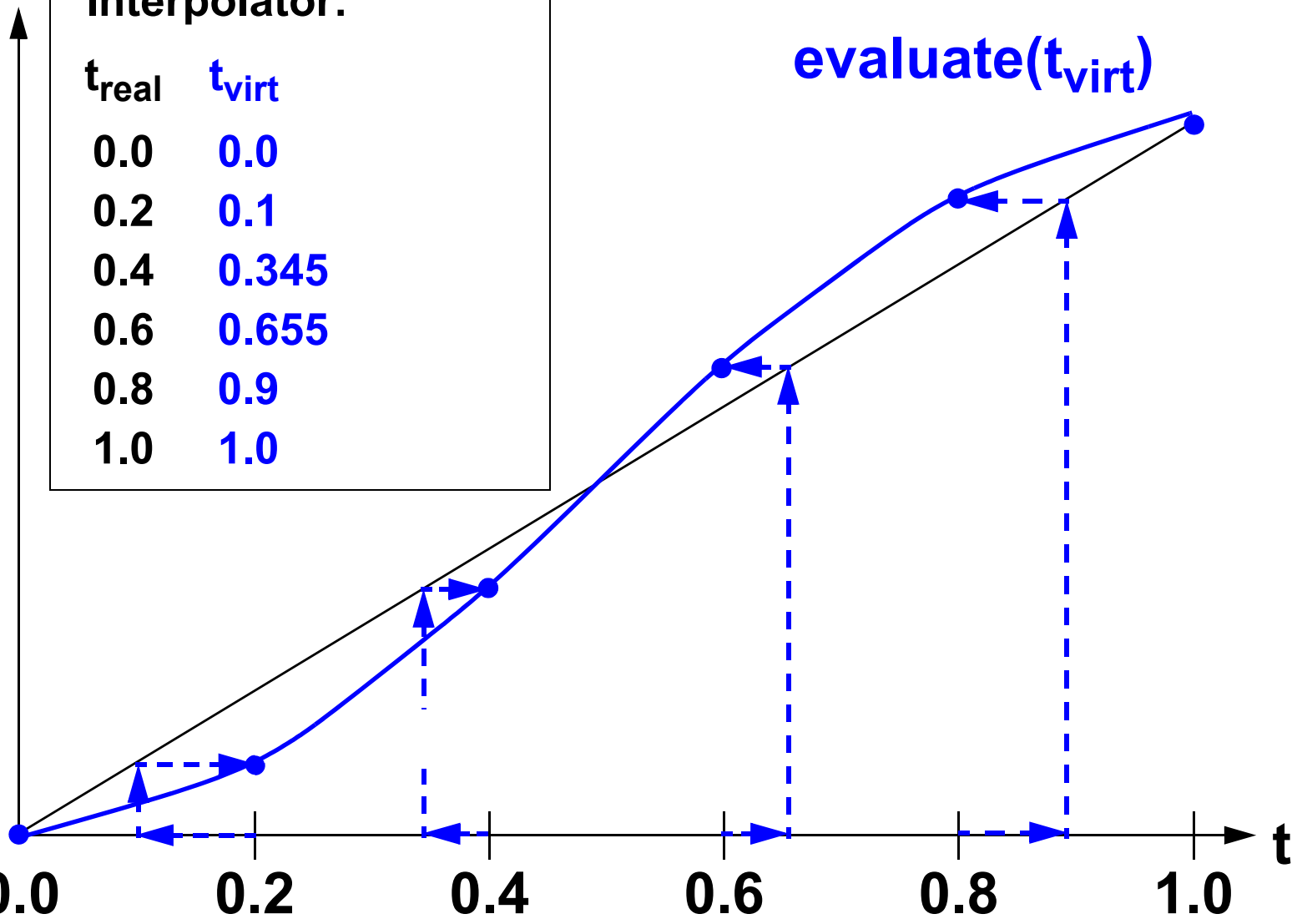


Fachhochschule Köln
Cologne University of Applied Sciences

TimeInterpolator

AccelerateDecelerate Interpolator:

t_{real}	t_{virt}
0.0	0.0
0.2	0.1
0.4	0.345
0.6	0.655
0.8	0.9
1.0	1.0



Fachhochschule Köln
Cologne University of Applied Sciences

TimeInterpolator

- Interface `TimeInterpolator`
- zu implementierende Methode:
  `float getInterpolation(float input)`
  - Parameter `input`: reale Zeit t_{real}
  - Resultat: virtuelle Zeit t_{virt}
    - geht dann als Parameter in `evaluate()`, also in die Berechung des Property-Werts
  - beide als "elapsed fraction of the animation"
    - also $0.0 \leq t_{real}, t_{virt} \leq 1.0$

Fachhochschule Köln
Cologne University of
Applied Sciences

Android: Property Animation

1.) Einfache und komplexe Techniken

2.) TypeEvaluator

3.) TimeInterpolator

4.) ValueAnimator

5.) Weitere Informationen

Fachhochschule Köln
Cologne University of
Applied Sciences

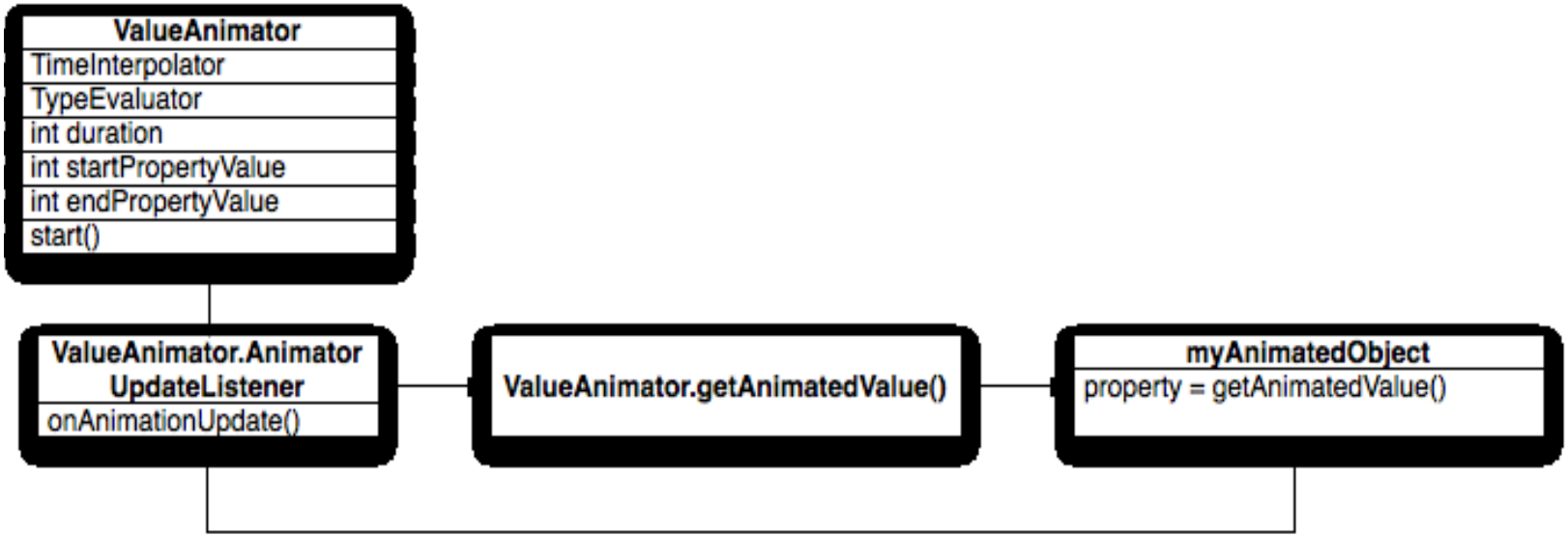
ValueAnimator

- **ValueAnimator**:
  Verallgemeinerung eines ObjectAnimators
  - **zur Definition und Steuerung einer Animation**
  - **gestützt auf**
    - **TypeEvaluator und**
    - **TimeInterpolator**
  - **mit UpdateListener**
    - **explizite Aktualisierung von Property-Werten**
- **Klasse `ValueAnimator`**
  - **Oberklasse von `ObjectAnimator`**

Fachhochschule Köln
Cologne University of
Applied Sciences

ValueAnimator

- grundlegende Zusammenhänge:



- Quelle: *http://developer.android.com*

Android: Property Animation

1.) Einfache und komplexe Techniken

2.) TypeEvaluator

3.) TimeInterpolator

4.) ValueAnimator

5.) Weitere Informationen

Fachhochschule Köln
Cologne University of
Applied Sciences

Weitere Details und Beispiel-Apps

Programmcode zu dieser Präsentation:
http://www.nt.fh.koeln.de/vogt/vma/videos.html

Android-Online-Dokumentation:
http://developer.android.com/guide/topics/graphics/prop-animation.html

- **auch:**
  API-Dokumentation der jeweiligen Klassen

Vorgänger-Video zu einfachen Techniken und Video zur Thread-gesteuerten Animation:
http://www.nt.fh.koeln.de/vogt/vma/videos.html

Fachhochschule Köln
Cologne University of Applied Sciences