The GAME Engineers

CG #3 – Game Loop und VSync

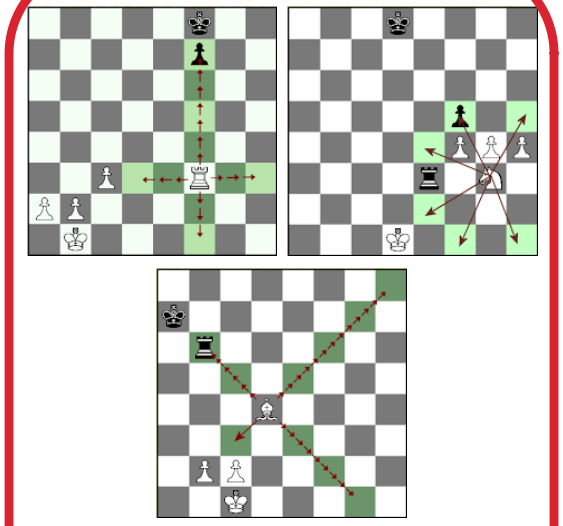
Inhalt

- Videospiele
  - Reale Spiele
  - Spieler
  - Spielmechanik
  - Ausflug: MVC-Pattern
  - Repräsentation

- Game Loop und VSync
  - Game Loop
  - VSync
  - Double Buffering

Videospiele – Reale Spiele

Spieler

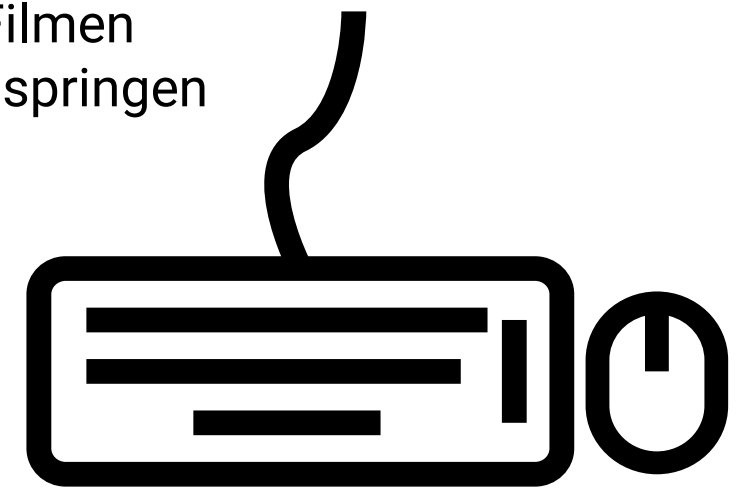


Spielregeln

Repräsentation

Videospiele – Spieler

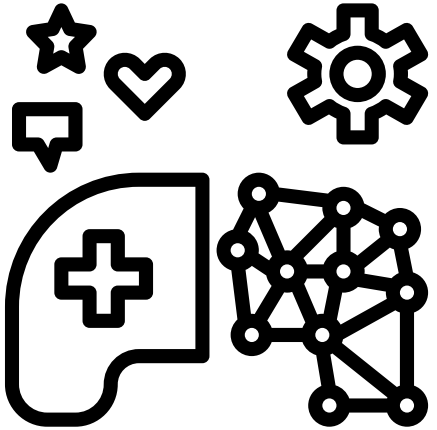
- Spieler bedeutet Interaktion
  - Interaktionen haben Auswirkungen auf die Spielwelt
  - Zeitpunkt unbekannt
    - im Gegensatz zu Filmen
    - Bsp.: zu früh/spät springen



Benutzereingaben

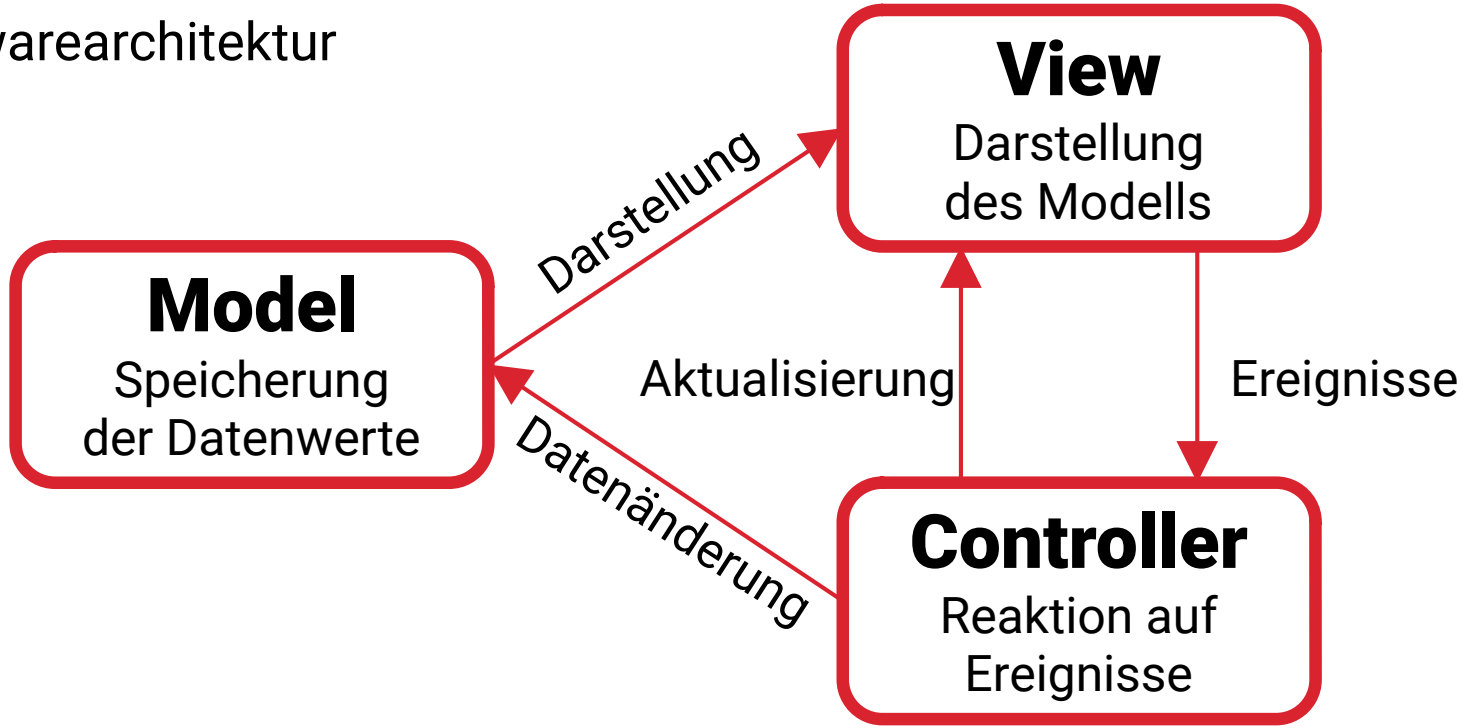
Videospiele – Spielmechanik

- Spielmechanik = Spielregeln + Spielstatus
  - Spielregeln
  - Spielstatus
    - In realen Spielen ist der Spielstatus am Spielbrett ablesbar
      → Die Informationen werden vom Auge erfasst und im Gehirn verarbeitet
      → Spielstatus und Repräsentation sind bei realen Spielen also verzahnt
      → Bei Videospielen ist das eher schwierig und daher entkoppelt



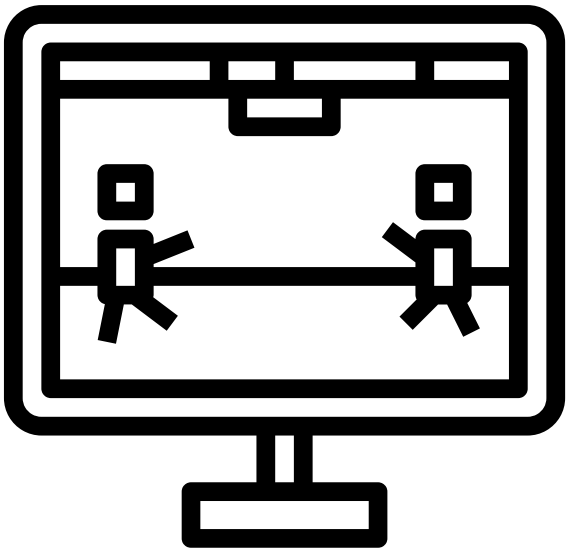
Videospiele – Ausflug: MVC-Pattern

- **M**odel – **V**iew – **C**ontroller
  - Softwarearchitektur

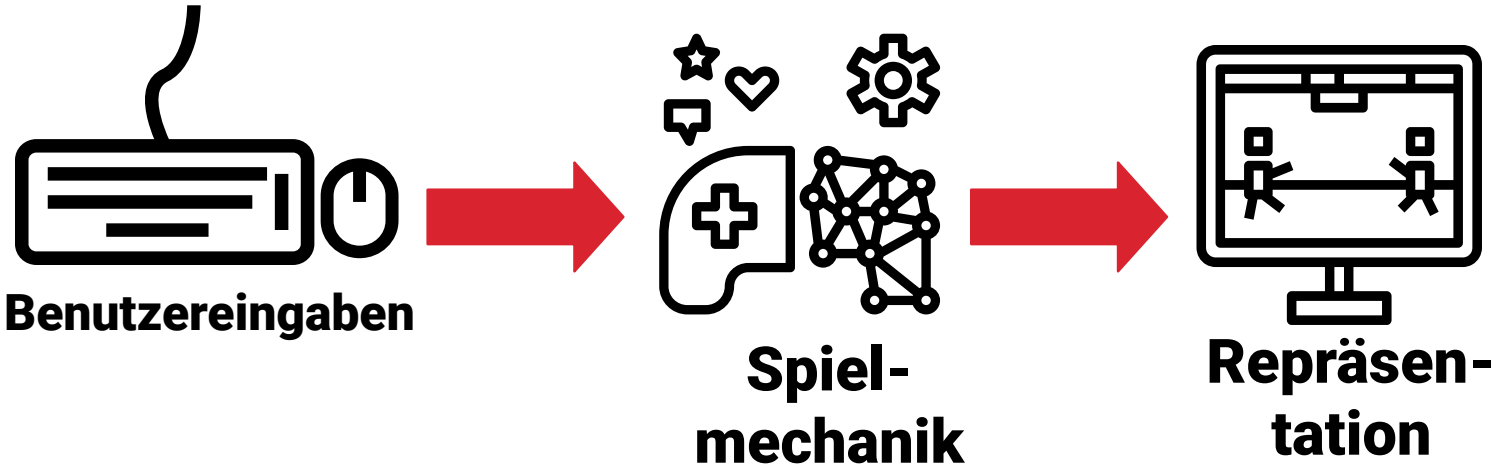


Videospiele – Repräsentation

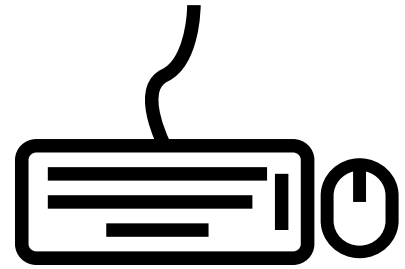
- Darstellung der Spielwelt
  - Folge von Bildern am Monitor



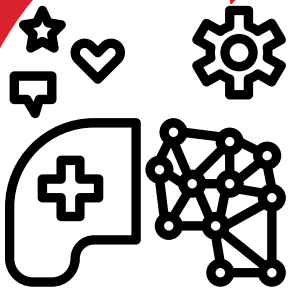
Videospiele – Zusammenspiel



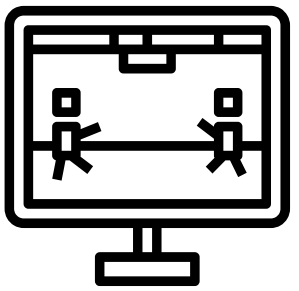
Game Loop und VSync – Game Loop



Benutzereingaben



Spiel-
mechanik

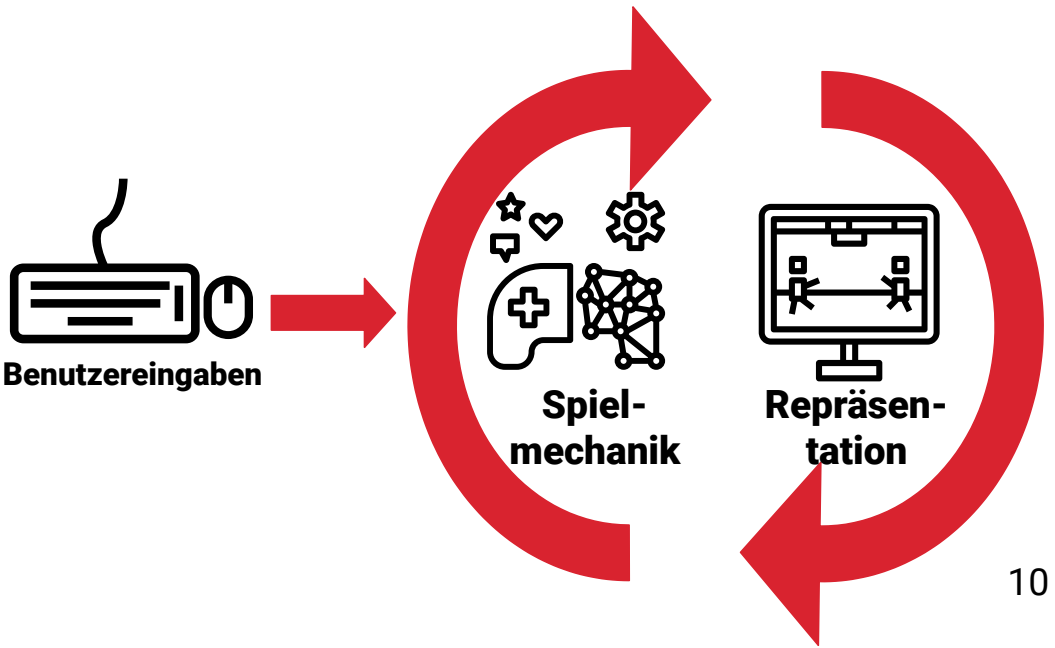


Repräsen-
tation

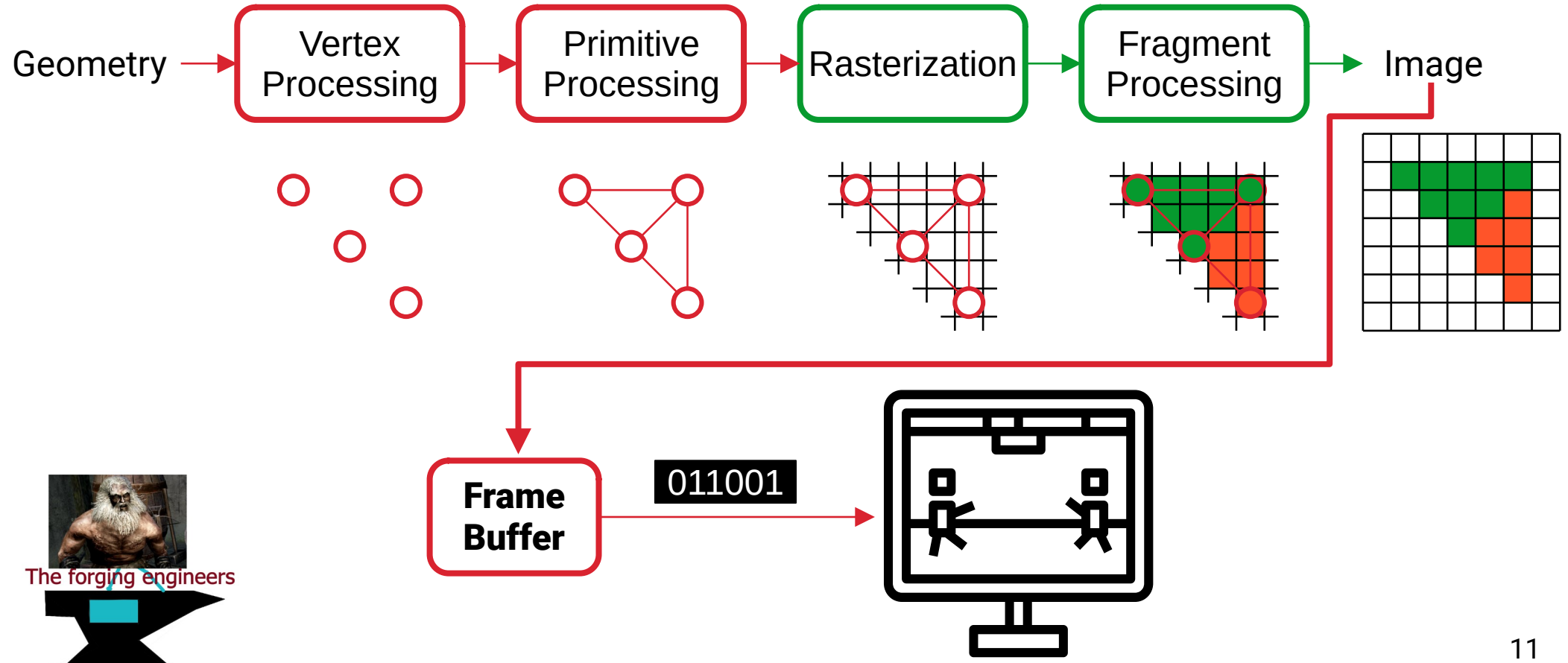
x mal pro
Sekunde

Game Loop und VSync – Game Loop

```
while (!finished)
{
    input = getInputState();
    updateGameState(input);
    drawGameWorld();
}
```

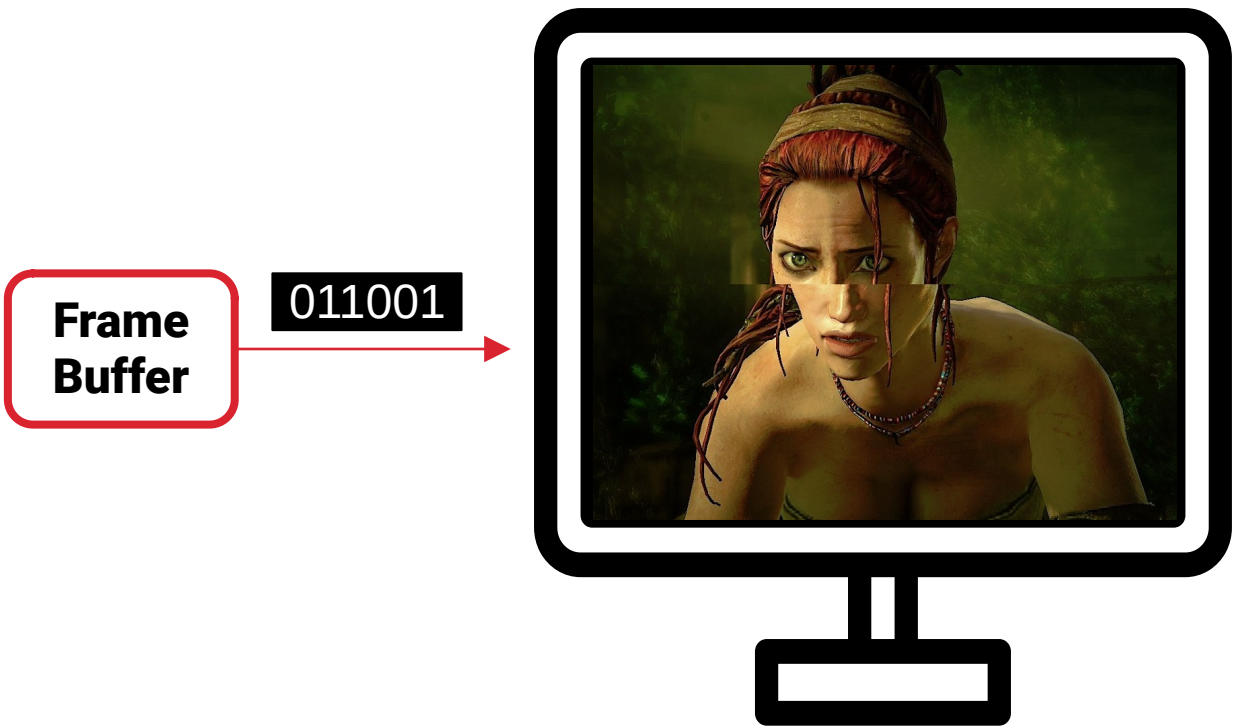


Game Loop und VSync – Einschub: Rendering Pipeline



Game Loop und VSync – Screen Tearing

- `drawGameWorld()` erstellt Bild und schreibt Daten in den Frame Buffer
  - Bilddaten aus dem Frame Buffer werden Sequentiell an den Monitor gesendet
  - Während z.B. die Hälfte der Bilddaten an den Monitor gesendet wurden, können die Daten im Frame Buffer sich ändern, weil ein neues Bild berechnet wurde



Screen tearing


Screen tearing
Old frame
New Frame

Game Loop und VSync – Screen Tearing

- Monitor baut Bild Zeile für Zeile auf
  → Vertikaler Aufbau

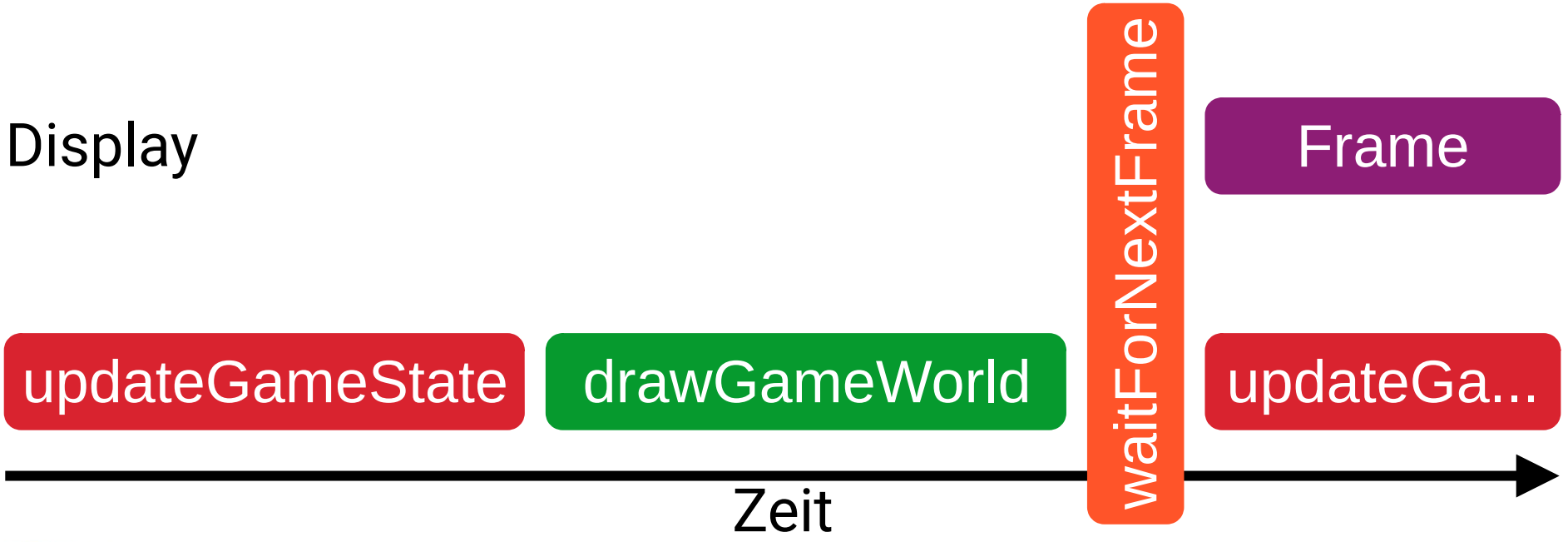
- Wann entsteht Screen Tearing?
  - Wenn Computer Bilder schneller/langsamer berechnet, als der Monitor anzeigen kann
    → Monitor Frequenz (oft 60 Hz oder mehr) != Bildfrequenz

Game Loop und VSync – Screen Tearing

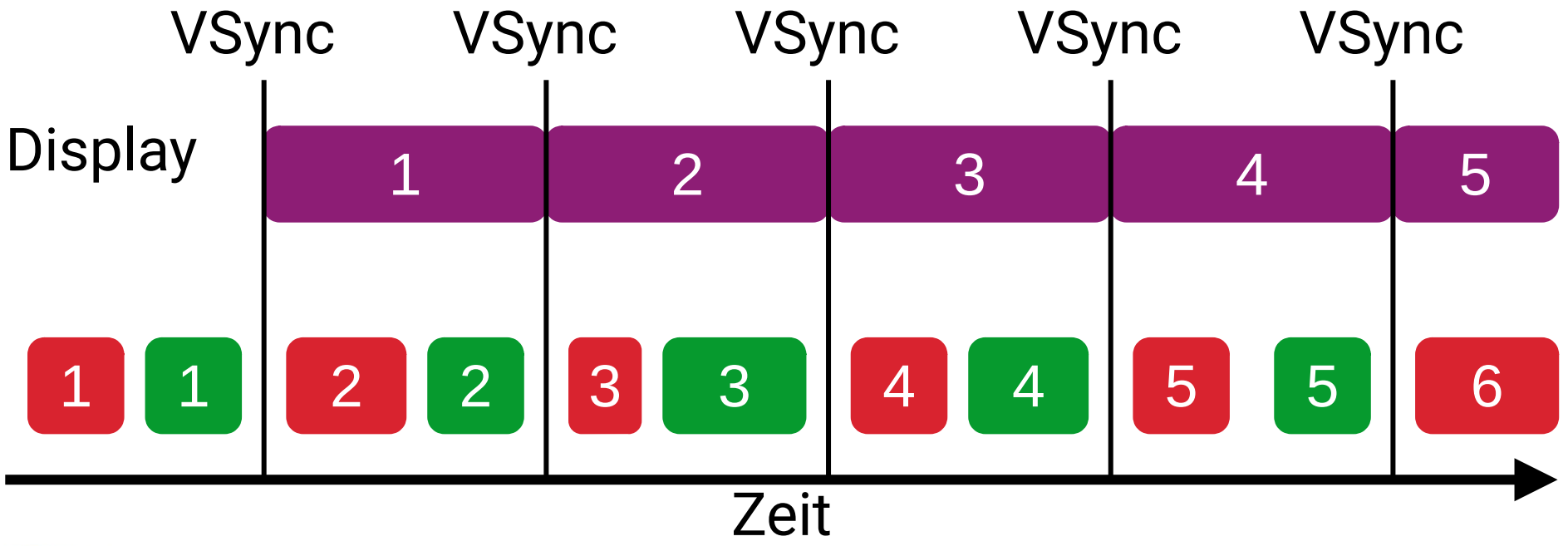
- Wie Screen Tearing vermeiden?
  - Warten, bis Bild vertikal aufgebaut wurde, bevor neues Bild in Frame Buffer geschrieben wird
    → Vertikale Synchronisation (VSync)

```
while (!finished)
{
    input = getInputState();
    updateGameState(input);
    drawGameWorld();
    waitForNextFrame();
}
```

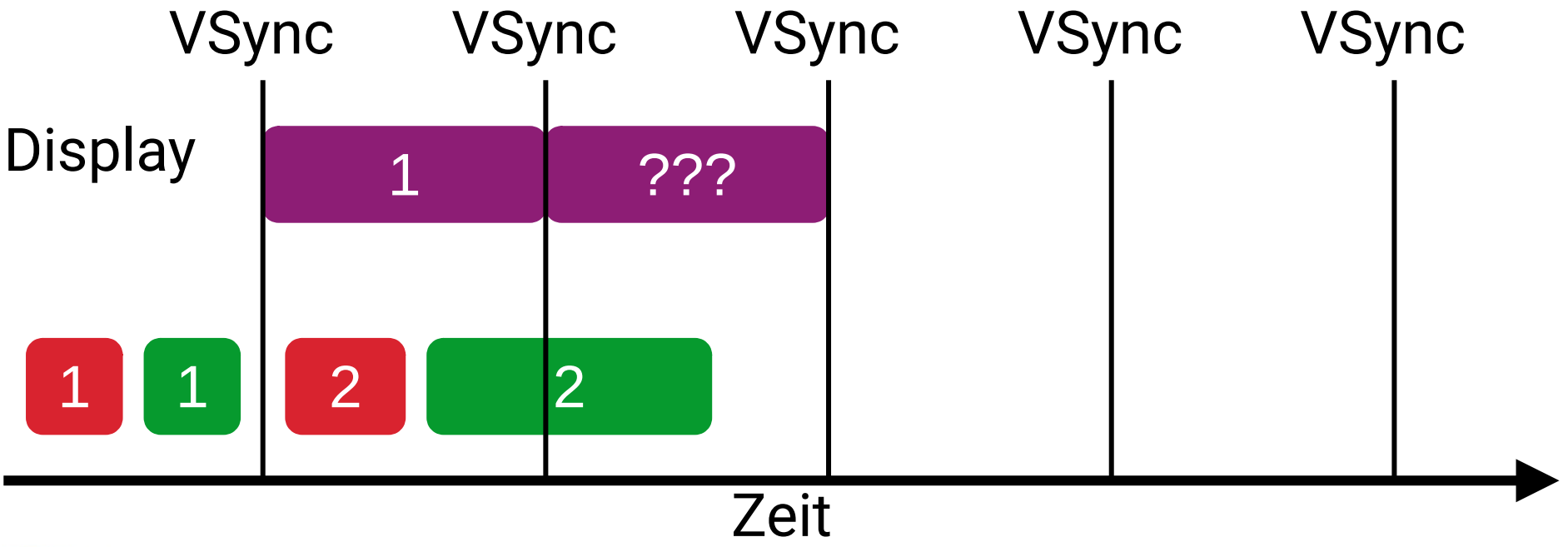
Game Loop und VSync – Vertikale Synchronisation



Game Loop und VSync – Vertikale Synchronisation

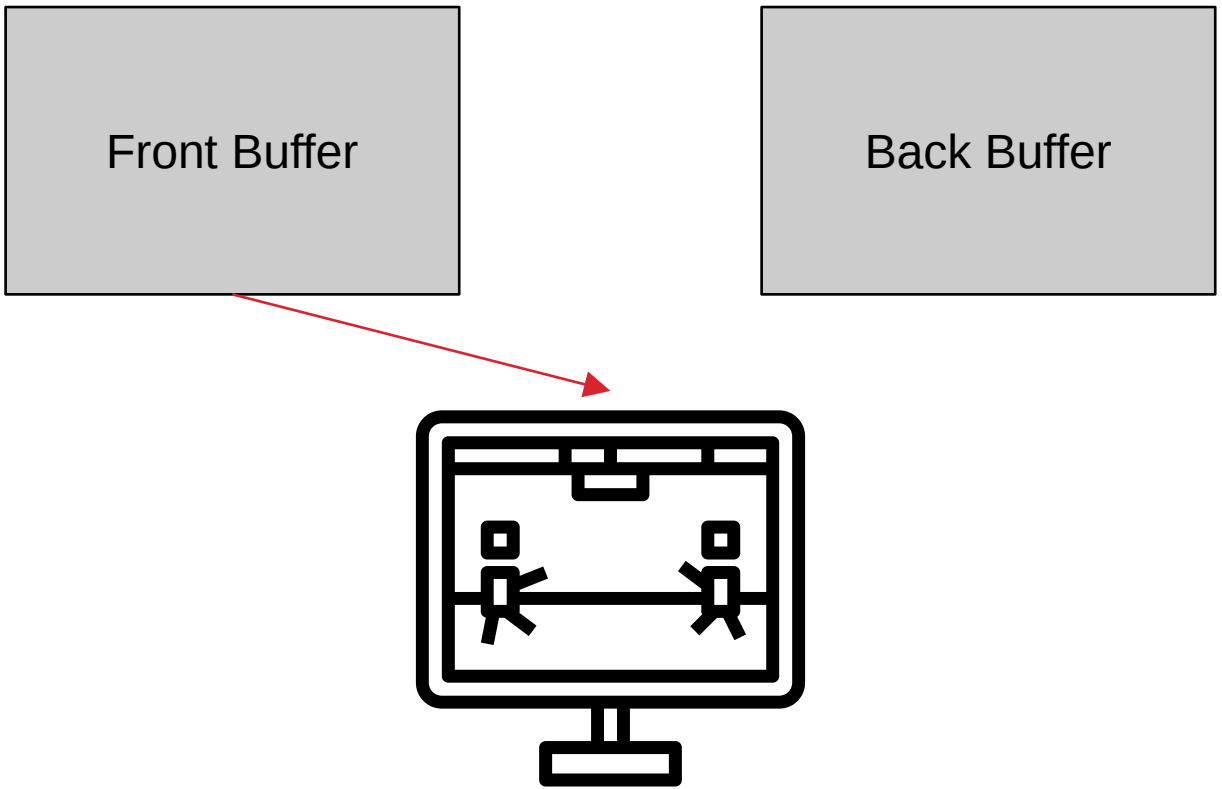


Game Loop und VSync – Vertikale Synchronisation



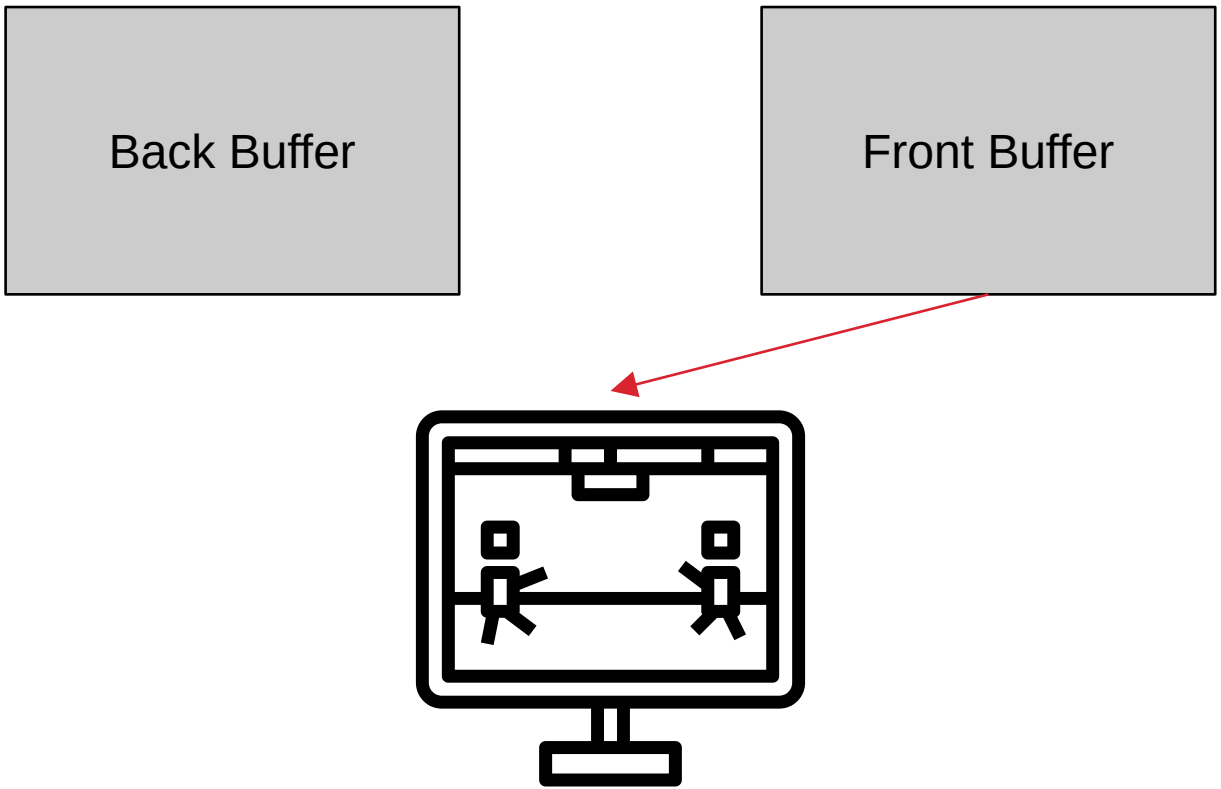
Game Loop und VSync – Double Buffering

- Zwei Frame Buffer
  - Front und Back Buffer

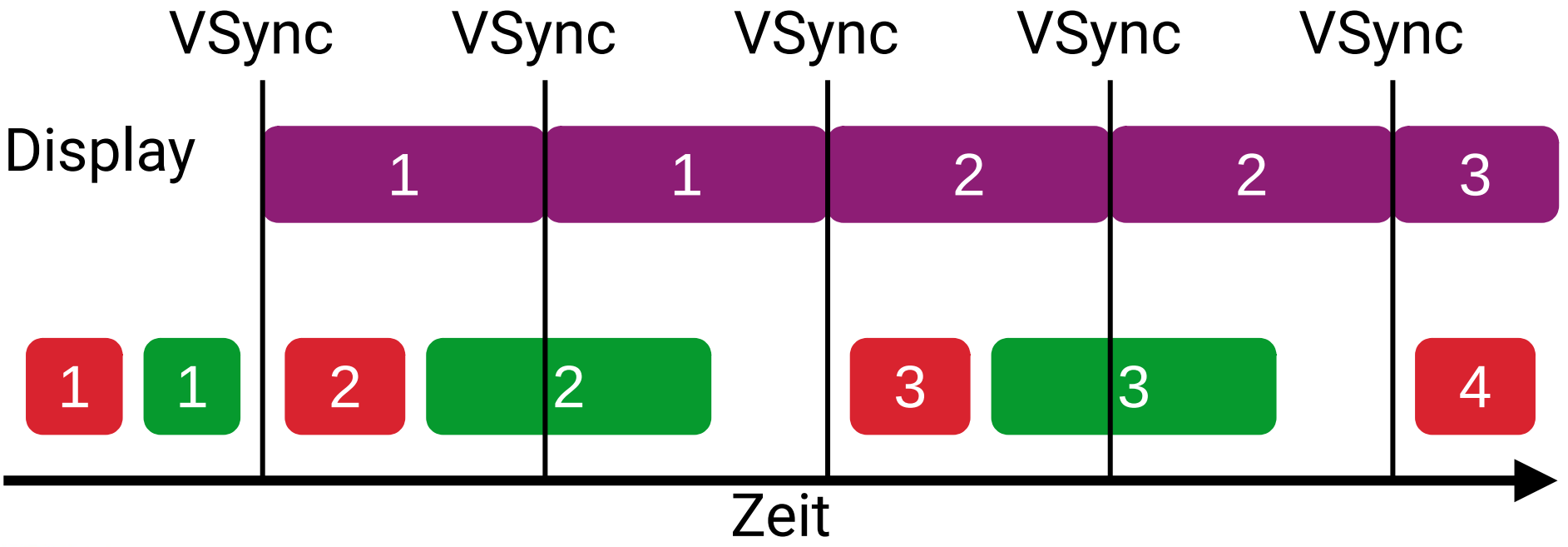


Game Loop und VSync – Double Buffering

- Zwei Frame Buffer
  - Front und Back Buffer



Game Loop und VSync – Double Buffering



Game Loop und VSync – Multi Buffering

- Zwei Frame Buffer
  - Front und Back Buffer

- In Realität ist es aber manchmal etwas komplexer, weil
  - Multi-threaded
  - GPU und CPU unterschiedlich berechnen
  - etc.

Ende

Fragen?