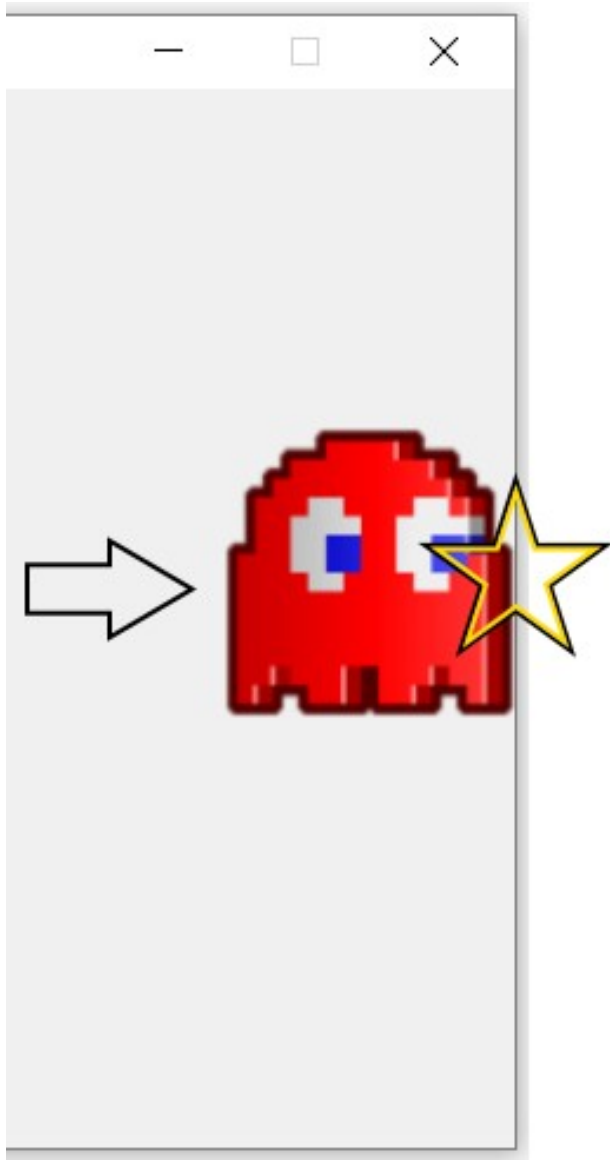


OOP mit Java

**get-Methoden
mit Parameter**

Bei Kollision: Sprite ändert Richtung



```
while (true)
{
    sp.bewege();

    if ( <Berührung> )
    {
        <Richtung ändern>
    }
    ...
}
```

Test auf Berührung

Benötigt Daten des **Sprite** und des **Fensters**

```
while (true)
{
    sp.bewege();

    if ( sp.getX() + sp.getWidth() >
         window.getWidth() )
    {
        <Richtung ändern>
    }

    ...
}
```

Test auf Berührung

Eleganter würde es so aussehen:

```
while (true)
{
    sp.bewege();

    if ( sp.berührtRechts(window) )
    {
        <Richtung ändern>
    }
    ...
}
```

Test auf Berührung

Eleganter würde es so aussehen:

```
while (true)
```

```
{
```

```
    sp.bewege();
```

```
    if ( sp.berührtRechts(window) )
```

```
    {
```

```
        <Richtung ändern>
```

```
    }
```

```
    ...
```

Fenster wird als
Parameter übergeben



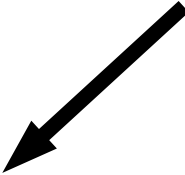
Test auf Berührung

Eleganter würde es so aussehen:

```
while (true)
{
    sp.bewege();

    if ( sp.berührtRechts(window) )
    {
        <Richtung ändern>
    }
    ...
}
```

Methode gibt true
oder false zurück



Get-Methode mit Parameter

```
public class Sprite  
{  
    private double x, y;  
    private GameImage image;  
    ...  
}
```

Get-Methode mit Parameter

(class Sprite)

...

```
public boolean berührtRechts(GameWindow pwin)
{
    if (x + image.getWidth() > pwin.getWidth())
    {
        return true;
    }
    else
    {
        return false;
    }
}
```


Get-Methode mit Parameter

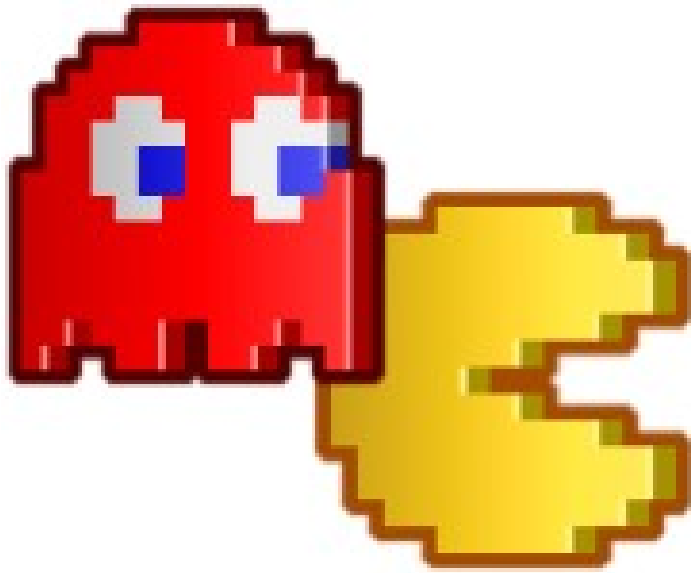
(class Sprite)

...

```
public boolean berührtRechts(GameWindow pwin)
{
    if (x + image.getWidth() > pwin.getWidth())
    {
        return true;
    }
    else
    {
        return false;
    }
}
```

Sprite-Daten
GameWindow-Daten
Rückgabewert

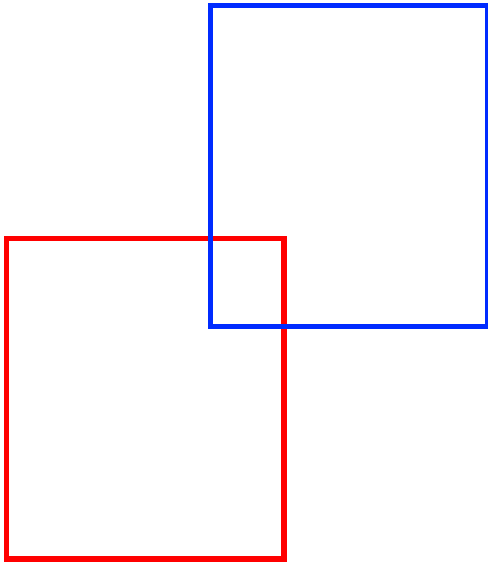
Kollision zweier Sprites



Oft reagieren Spiele,
wenn sich **zwei Sprites**
berühren:

- Punktgewinn
- Powerup
- Schaden nehmen
- ...

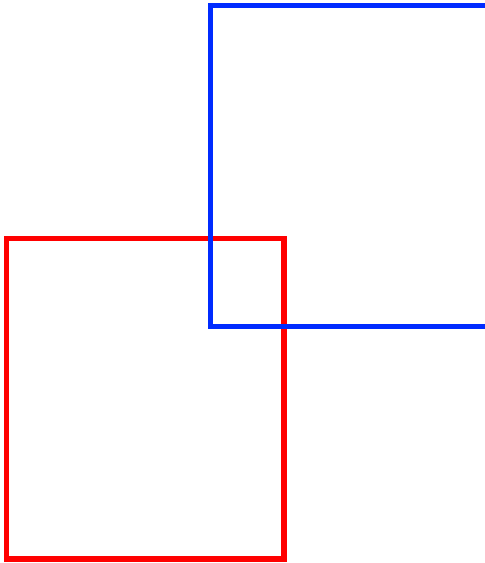
Kollision zweier Sprites



Die Frage, ob sich zwei **Sprites** berühren,

wird vereinfacht zur Frage, ob zwei **Rechtecke** überlappen.

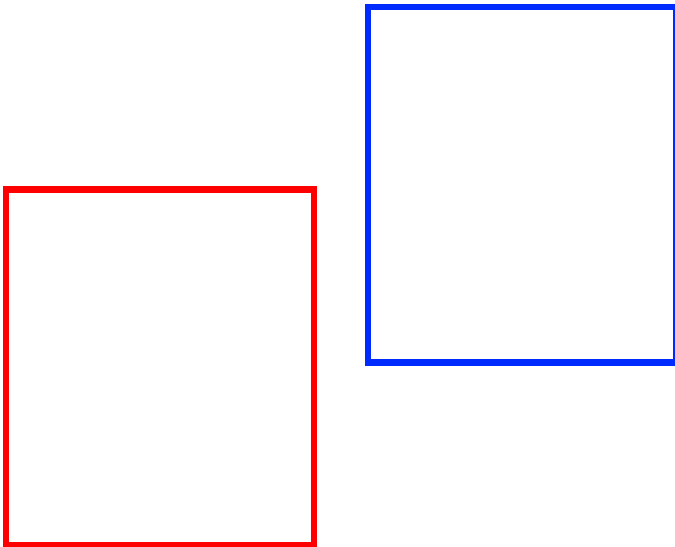
Überlappen zwei Rechtecke?



Prüfung ist aufwendig

Einfacher:
prüfen, ob Rechtecke
nicht überlappen.

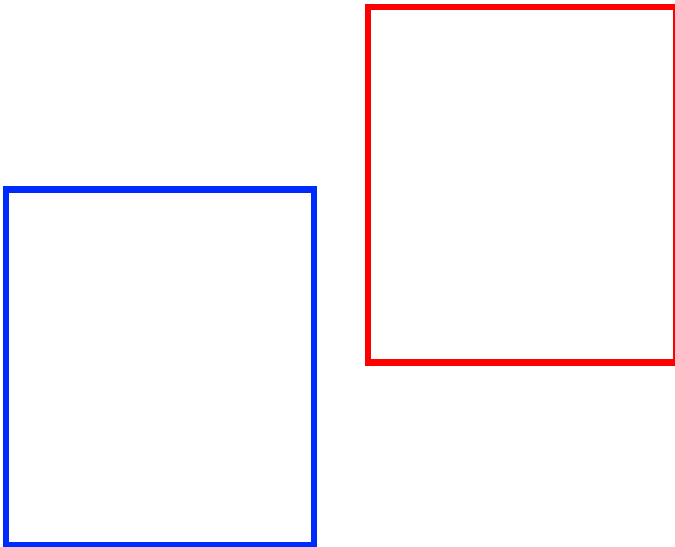
Keine Überlappung?



1. Fall:

$$x \text{ (blau)} > x + \text{Breite (rot)}$$

Keine Überlappung?



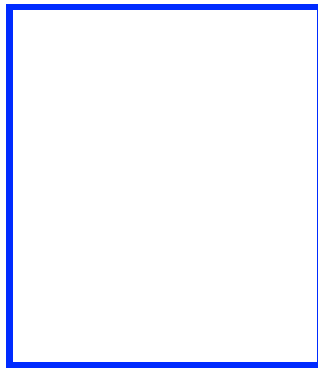
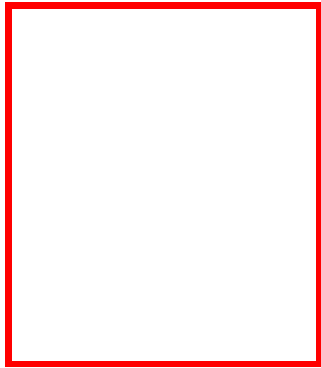
1. Fall:

$$x \text{ (blau)} > x + \text{Breite (rot)}$$

2. Fall (umgekehrt):

$$x \text{ (rot)} > x + \text{Breite (blau)}$$

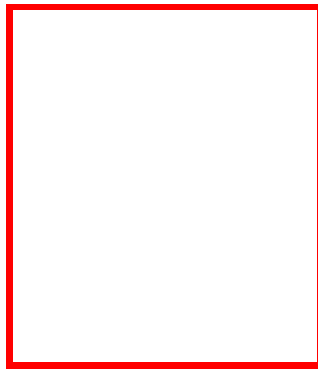
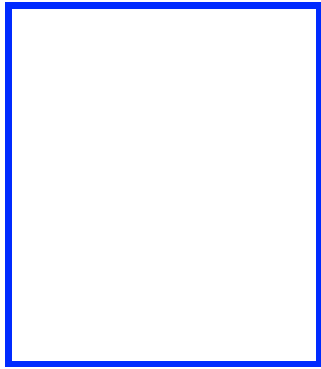
Keine Überlappung?



3. Fall:

$y \text{ (blau)} > y + \text{Höhe (rot)}$

Keine Überlappung?



3. Fall:

$$y \text{ (blau)} > y + \text{Höhe (rot)}$$

4. Fall (umgekehrt):

$$y \text{ (rot)} > y + \text{Höhe (blau)}$$

Methode: prüft ob sich Sprites berühren

```
public class Sprite
{
    public boolean berührt(Sprite ps)
    {
        if (x > ps.getX() + ps.getWidth()) // 1. Fall
        { return false; }
        else if (...) // 2. Fall
        { return false; }
        ... // 3., 4. Fall
        else
        { return true; } // 1.-4. ausgeschlossen
    }
}
```

Autor / Quellen

Autor:

- Christian Pothmann (cpothmann.de)
Freigegeben unter CC BY-NC-SA 4.0, Mai 2021



Grafiken:

- Pacman-Grafiken: strategywiki.org, Lizenz CC BY-SA 3.0