

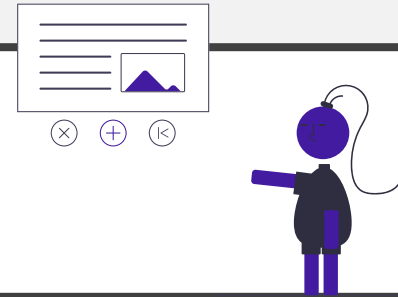


MINIGOLF

Programmieren II – Praktikum

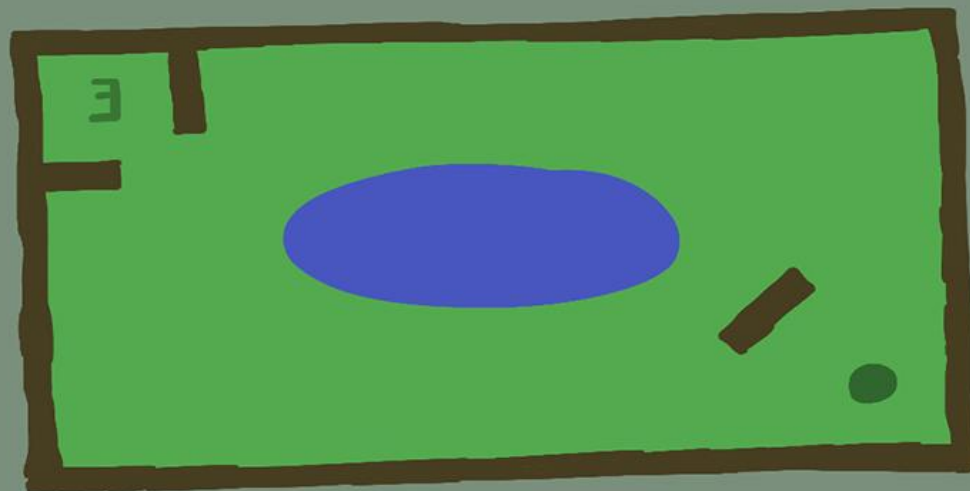
Christoph Gretenkort (2190167), Evelyn Nolden (2210050)

INHALT



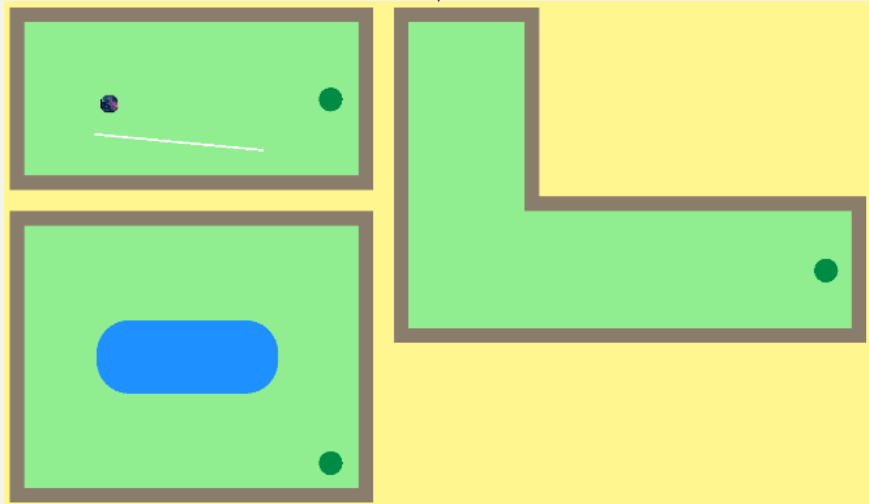
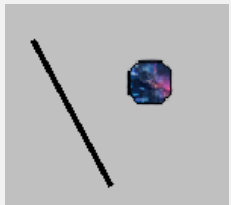
1. Erste Idee → Skizzen
2. Entwicklung der GUI
3. Balllogik
4. Bandenlogik
5. Zweispielerkomponente





HIGHSCORE / PUNKTE	
AAA	1111
AAA	1111
AAA	1111

ENTWICKLUNG



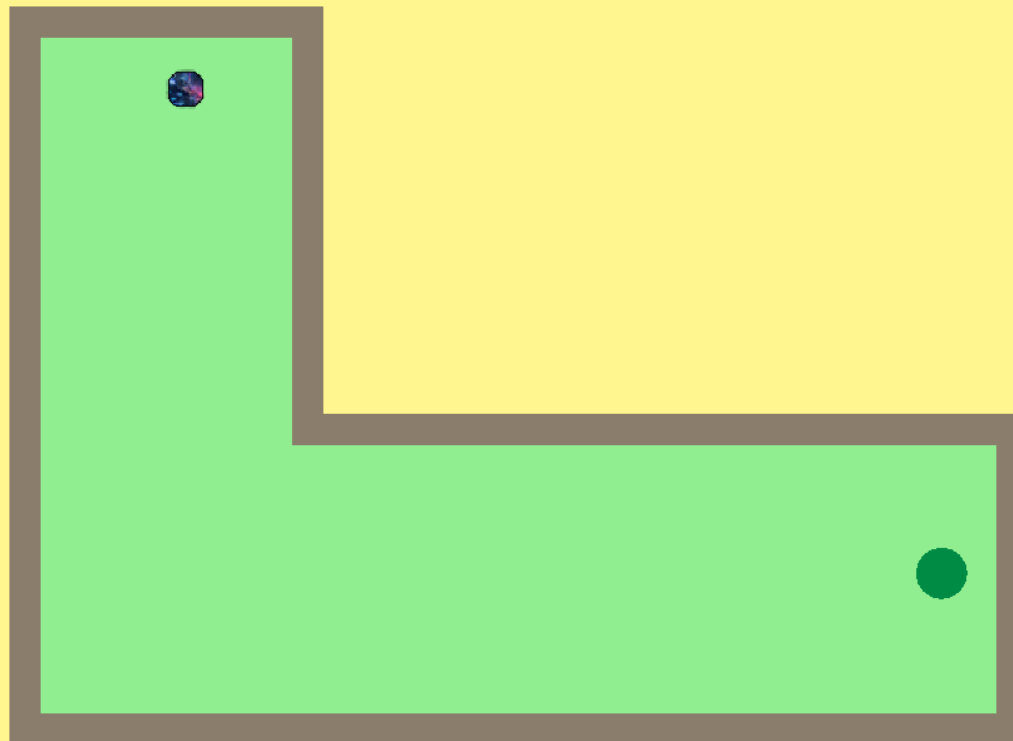
Spieler 2 ist Sieger!

Schlaganzahl - Spieler 1: 6
Schlaganzahl - Spieler 2: 6

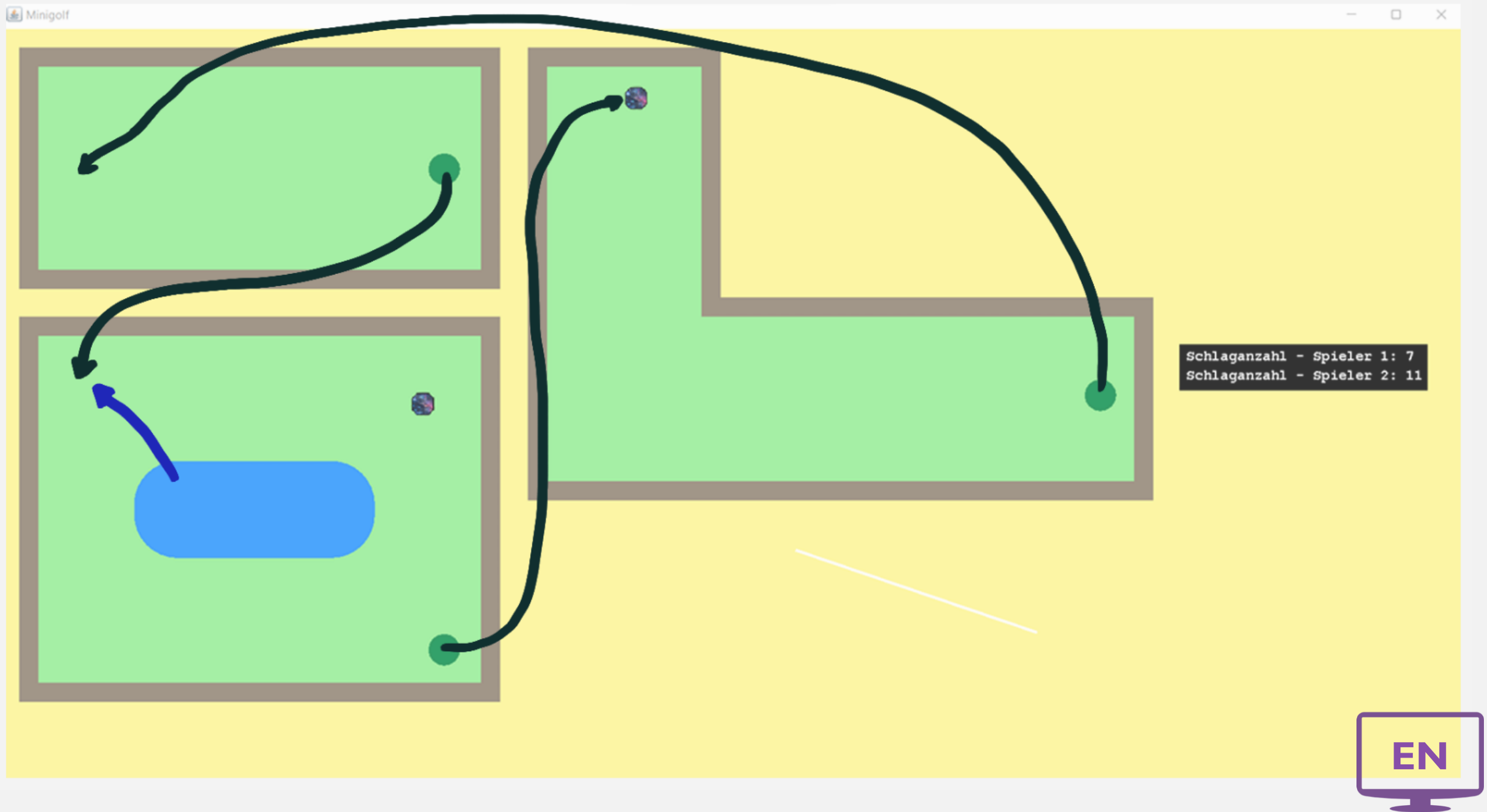
Spieler 2 ist Sieger!

Schlaganzahl - Spieler 1: 11
Schlaganzahl - Spieler 2: 15

EN



Schlaganzahl - Spieler 1: 7
Schlaganzahl - Spieler 2: 11



CODEBEISPIEL - VEKTORBERECHNUNG

```
//Bewegung nach unten rechts
if (endX >= startX && endY >= startY)
{
    ball.geschwX = (int) (Math.sqrt(endX-startX));
    ball.geschwY = (int) (Math.sqrt(endY-startY));
}
//Bewegung nach oben links
else if (endX <= startX && endY <= startY)
{
    ball.geschwX = ((int) (Math.sqrt(startX-endX))) * -1;
    ball.geschwY = ((int) (Math.sqrt(startY-endY))) * -1;
}
```

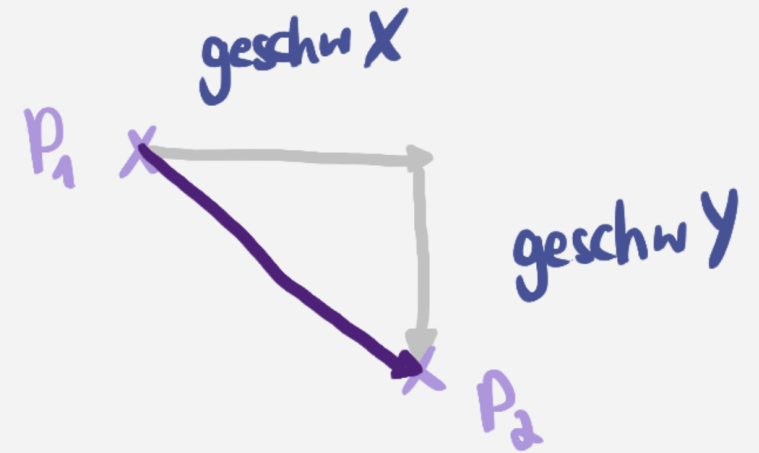
```
//Bewegung nach unten links
else if (endX <= startX && endY >= startY)
{
    ball.geschwX = ((int) (Math.sqrt(startX-endX))) * -1;
    ball.geschwY = ((int) (Math.sqrt(endY-startY)));
}
//Bewegung nach oben rechts
else if (endX >= startX && endY <= startY)
{
    ball.geschwX = ((int) (Math.sqrt(endX-startX)));
    ball.geschwY = ((int) (Math.sqrt(startY-endY))) * -1;
}
```


BEISPIEL: BEWEGUNG NACH UNTEN RECHTS

```
geschw X = (int) (Math.sqrt (end X - start X));  
geschw Y = (int) (Math.sqrt (end Y - start Y));
```

geschw X &
geschw Y
sind Integer

$\sqrt{\text{end X} - \text{start X}}$ bzw.
 $\sqrt{\text{end Y} - \text{start Y}}$



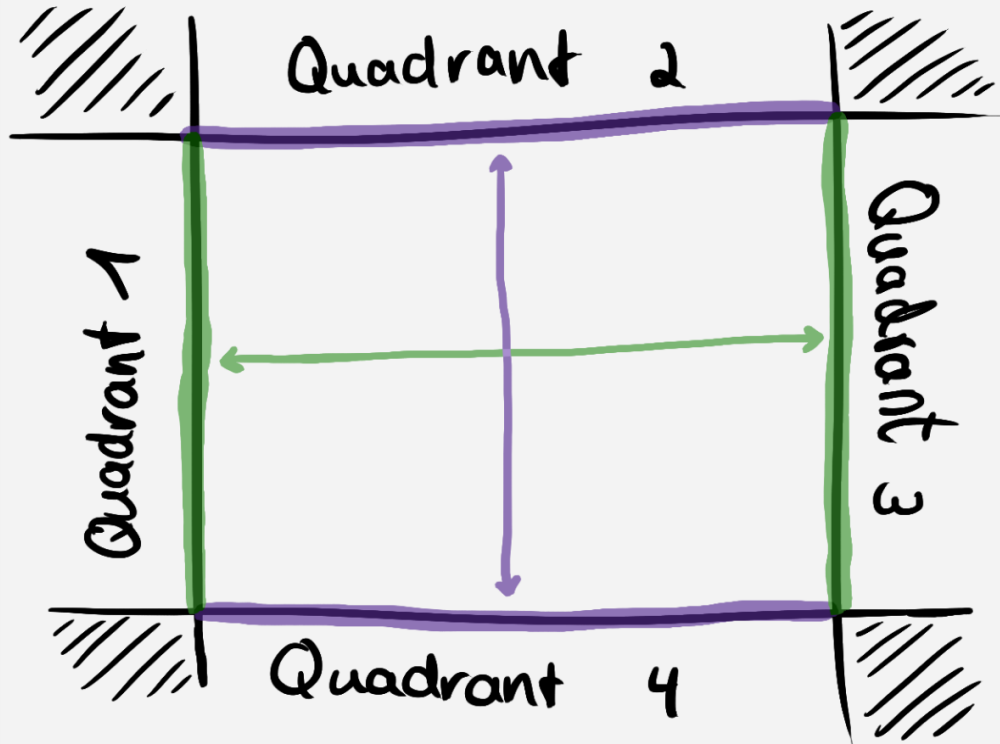
P_1 (start X | start Y)
 P_2 (end X | end Y)

BANDENLOGIK

- Bestimmt , ob und wie der Ball abprallen soll
- Musste überdacht und neu programmiert werden (*siehe nächste Folie*)
- **Banden-Klasse:** Vereinfachung der Arbeit mit Banden
→ Werte für Position der Außenkanten und Startposition des Balles werden dort gespeichert
- **Abprallen:** Realisierung durch Invertierung der dazugehörigen Geschwindigkeit



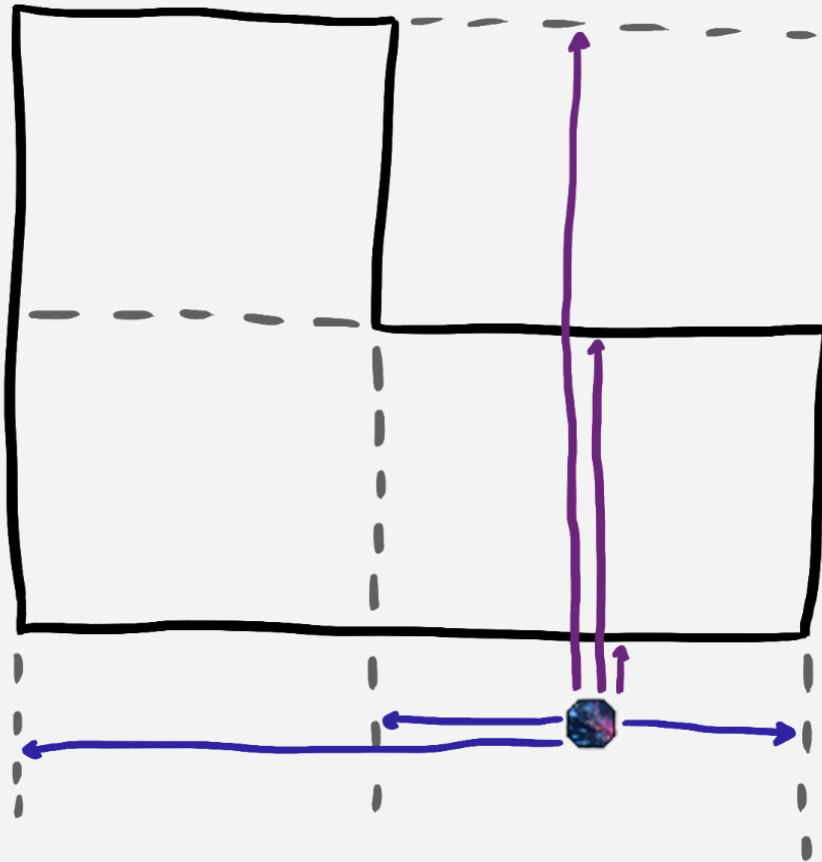
BANDENLOGIK



Vorher:

- i. In welchem Quadranten außerhalb?
- ii. Anhand der Info aus i. spiegeln

BANDENLOGIK



Nachher:

- i. Abstand zur Orthogonalen der Banden bestimmen
- ii. Aus diesen Abständen den geringsten bestimmen
- iii. Abprallen mithilfe des geringsten Wertes sowie der Richtung des Balles

CODEBEISPIEL - BOUNCE

```
// Kehrt die Geschwindigkeit in die gewünschte Richtung um
// p=0: Linker Rand
// p=1: Rechter Rand
// p=2: Oberer Rand
// p=3: Unterer Rand
private void invert(int p) {
    Ball ball = game.getAktPlayer().getBall();
    switch (p) {
        case 0:
            if (ball.geschwX < 0) {
                this.invert('x', ball);
            }
            break;
        case 1:
            if (ball.geschwX > 0) {
                this.invert('x', ball);
            }
            break;
        case 2:
            if (ball.geschwY < 0) {
                this.invert('y', ball);
            }
            break;
        case 3:
            if (ball.geschwY > 0) {
                this.invert('y', ball);
            }
            break;
    }
}
```

```
/*
    Hilfsmethode für die Obige, um diese zu verkleinern
    und doppelten Code zu vermeiden:
*/
private void invert(char direction, Ball pBall) {
    if (direction == 'x') {
        pBall.geschwX *= -1;
    } else {
        pBall.geschwY *= -1;
    }
}
```



ZWEISPIELERKOMPONENTE

- **Zweiter Spieler:** Sämtliche Attribute und Funktionen des Ersten
- **Abwechselndes Spiel**
 - boolean bestimmt, welcher Spieler der Aktive ist
 - boolean wechselt nach Treffen des Lochs



ENDE

Vielen Dank für Eure Aufmerksamkeit!