

# Das Ballwurfspiel in GLOOP

## Lernziel

Mit diesem Projekt lernst du, automatisierte Bewegungen und Animationen in GLOOP zu programmieren. Dazu lernst du eine neue Kontrollstruktur kennen.

## Ziel des "Ballwurfspiels"

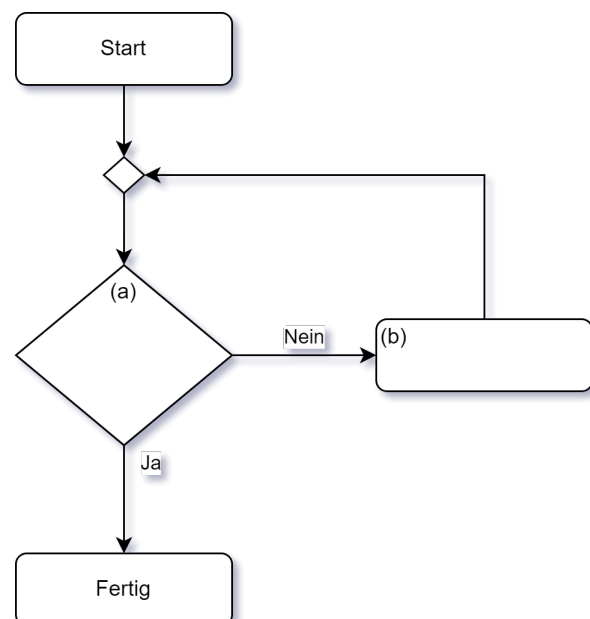
Der Ball soll mit Hilfe der Pfeiltasten (oben, unten, links, rechts) positioniert und mit der Leertaste geworfen werden. Trifft er die Zielscheibe, hast du gewonnen.



## Vorbereitung: Programmabschnitte wiederholen

- ① Im Zusammenhang mit Verzweigungen hast du bereits Flussdiagramme kennengelernt. Das abgebildete, unvollständige Flussdiagramm soll das Herabfallen eines Balls aus der Luft modellieren.

- A) Vervollständige das Flussdiagramm, sodass es das Herabfallen korrekt darstellt.  
Ergänze dazu im Element (a), welche Bedingung darüber entscheidet, ob das Herabfallen des Balls abgeschlossen („Fertig“) ist.  
Ergänze außerdem im Element (b), welche Aktion der Computer ausführen muss, wenn die Bedingung (noch) nicht zutrifft.



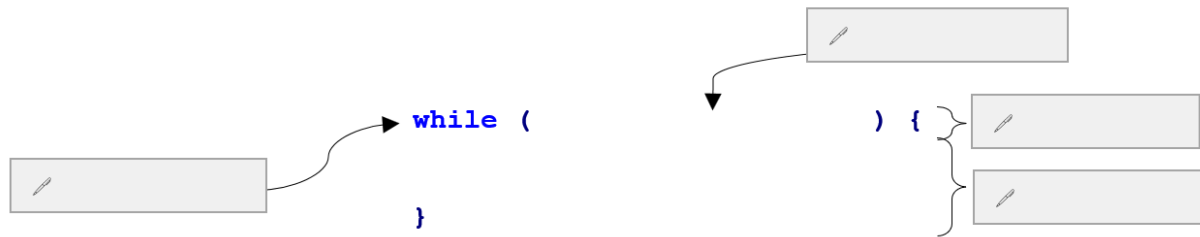
- B) Erkläre die Bedeutung des Pfeils, der vom Element (b) aus nach oben führt, im Sachzusammenhang.

- C) Formuliere umgangssprachlich den Programmablauf, der durch das Flussdiagramm dargestellt wird. Verwende dazu die folgenden Formulierungshilfen:

- *Höhe über dem Boden*
- *Verschiebe den Ball*
- *Solange*

- D) In IServ findest du im Kursordner das BlueJ-Projekt "Ballfall". Lade es herunter, entpacke es und starte mit einem Doppelklick auf "package.bluej" das Projekt. Vervollständige dann das lückenhafte Programm in der Klasse "Ballfall", um die Animation eines herabfallenden Balls in GLOOP sichtbar zu machen.

## Sicherung: Kontrollstruktur „while-Schleife“



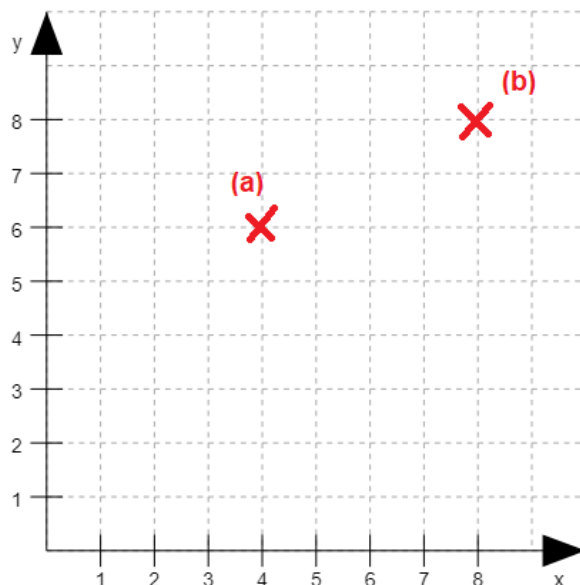
② Setze die folgenden Begriffe in die Lücken ein, um die Bestandteile der while-Schleife korrekt zu benennen.

- *Schleifenkopf*
- *Bedingung*
- *Schlüsselwort „while“*
- *Schleifenkörper*

## Das Ballwurfspiel entwickeln

- ③ In IServ findest du im Kursordner das BlueJ-Projekt „Ballwurfspiel“. Lade es herunter, entpacke es und starte mit einem Doppelklick auf „package.bluej“ das Projekt.
- A) Teste das Programm in seinem aktuellen Zustand, um zu prüfen, was bereits funktioniert und was noch nicht funktioniert.
- B) Analysiere den Programmcode, um zu identifizieren, welche Abschnitte des Codes für die bereits funktionierenden Teile des Spiels verantwortlich sind.
- C) Erweitere das Programm, damit auch die folgenden Aspekte umgesetzt werden:
- Die Pfeiltasten „links“ und „rechts“ können den Ball nach links bzw. rechts bewegen.
  - Die Leertaste schießt den Ball in Richtung der Z-Achse (max. 1000 Pixel weit).

## Treffer?



④ Um zu prüfen, ob der Ball die Zielscheibe getroffen hat, müssen wir den Abstand zwischen beiden Objekten berechnen.

- A) Berechne den Abstand zwischen den Punkten (a) und (b) mit Hilfe des *Satzes des Pythagoras*.
- B) Übertrage deine Berechnung aus Aufgabenteil A) in Java-Code (per Hand mit Stift auf Papier oder Tablet).



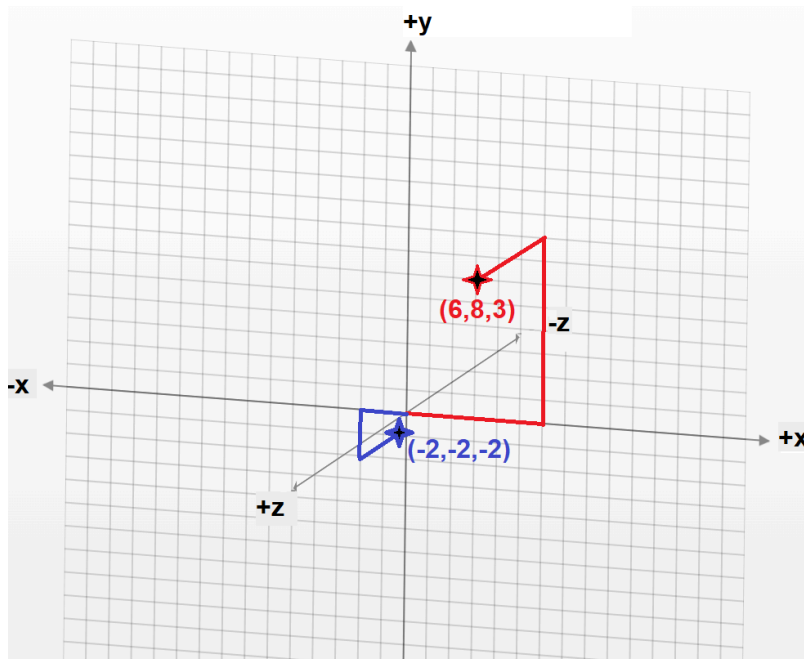
### Berechnungen in Java

Java stellt einige Methoden bereit, mit denen man Berechnungen durchführen kann:

- `Math.sqrt(n)` liefert  $\sqrt{n}$  zurück.
- `Math.pow(b,e)` liefert  $b^e$  zurück.

C) Wende die Berechnung des Abstands nun auf das Ballwurfspiel an, um zu ermitteln, ob der Ball die Zielscheibe getroffen hat.

## Für Sprinter: Pythagoras in 3D?



Der virtuelle Raum in GLOOP, genauso wie die reale Welt, besitzt drei Dimensionen. Zwei Objekte können also nicht nur in den Dimensionen X und Y eine gewisse Distanz zueinander besitzen, sondern auch in der Dimension Z.

Die beiden Punkte im abgebildeten, dreidimensionalen Koordinatensystem besitzen einen Abstand von deutlich mehr als 13 Einheiten. Lässt man die Z-Achse außer acht, ergibt sich jedoch ein Abstand von nur ca. 12,81 Einheiten.

### ⑤ Für Sprinter:

- Analysiere das dargestellte, dreidimensionale Szenario, indem du dem nebenstehend abgebildeten Link/QR-Code folgst, die zwei Punkte in das 3D-System einträgst und durch Drehen und Schwenken von allen Seiten betrachtest.
- Berechne den Abstand zwischen den beiden abgebildeten Punkten mit Hilfe des Satz des Pythagoras nur in Y- und Z-Richtung (lasse also die X-Dimension außer acht).
- Modifiziere nun den Satz des Pythagoras so, dass auch eine dritte Dimension in die Berechnung einbezogen wird. Notiere deinen Vorschlag.
- Berechne den tatsächlichen Abstand zwischen den beiden abgebildeten Punkten, indem du deinen Vorschlag ausprobierst.  
*Kontrollergebnis: 13,75 Einheiten.*



<https://technology.cpm.org/general/3dgraph/>