

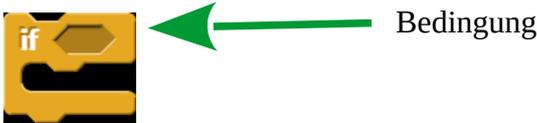
Einseitig bedingte Anweisung

 <p>Aufgabe 1</p>	<p>Die Simpsons</p> <p>Unter den Vorlagen findest du die Datei Simpsons.xml. Speichere sie auf deinem USB-Stick ab und öffne sie mit Snap! .</p>  <p>Homer jagt Bart nach, da dieser wieder einmal etwas angestellt hat. Dies möchten wir jedoch dieses Mal durch ein Programm nachstellen, indem der Benutzer Homer mit Hilfe der Tastatur steuern kann.</p> <p>Beachte dabei:</p> <ul style="list-style-type: none">• Lass den bereits vorhandenen Programmcode unverändert.• Durch Drücken der Leertaste kannst du immer wieder zum Ausgangszustand zurückkehren. 
---	---

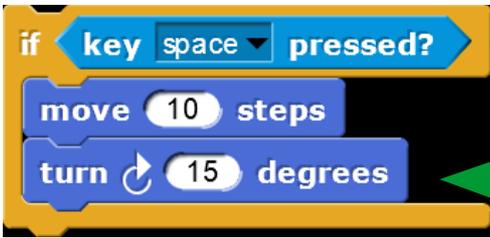
Wir stellen fest, dass wir diese Aufgabe zwar mit Hilfe des Bausteins **when** umsetzen können, jedoch sind dann Bewegungen der Figuren auch schon vor Start des Programms durch Klicken auf die grüne Flagge möglich. Dies möchten wir jedoch vermeiden.



Für solche Fälle gibt es einen weiteren wichtigen Baustein für Algorithmen – die **einseitig bedingte Anweisung**. Diese wird verwendet, um eine Sequenz – einen Programmablauf – in Abhängigkeit von einer Bedingung auszuführen. Diese Sequenz wird nur genau dann ausgeführt, wenn die entsprechende Bedingung eintritt.



Das Programm überprüft an der entsprechenden Stelle, ob die Bedingung wahr ist – und damit der Programmcode ausgeführt werden muss – oder, ob sie falsch ist und gar nichts ausgeführt wird.



Bedingung → führe den folgenden Programmcode aus, **falls die Leertaste gedrückt** wird

Programcode, der ausgeführt wird, **falls die Leertaste gedrückt** wird



Aufgabe 2

Die Simpsons

Realisiere mit Hilfe der einseitig bedingten Anweisung eine Art Spiel, indem der Spieler (Homer) Bart fangen muss.

Beachte dabei:

- Homer soll sich durch die **Pfeiltasten** steuern lassen.
- Denke daran, dass du während das Programm läuft **immer wieder** überprüfen musst, ob eine dieser Tasten gedrückt ist.

Teste anschließend, ob sich deine Figur wie gewünscht bewegen lässt.



Aufgabe 3

Die Simpsons

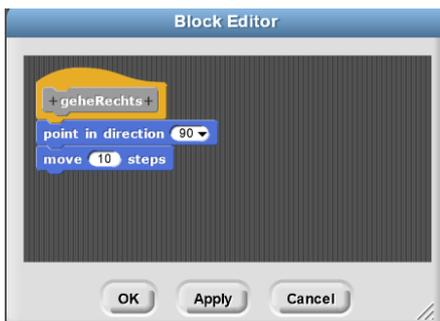
Nun soll auch Bart durch einen Spieler bewegt werden können. Verwende dazu die Tasten **W, A, S, D**.

Was stellst du fest?

Snap! bietet die Möglichkeit, **eigene Methoden bzw. Bausteine zu definieren**. Diese können dann im Programm **immer wieder verwendet** werden. Dies ist besonders nützlich, wenn diese **mehrfach** in demselben Programm benötigt werden.

Unter **Variables** gibt es den Button **Make a block**. Durch Anklicken gelangt man zu einem Fenster, indem man den Namen des neuen Bausteins, sowie seine Obergruppe (Motion, Control, etc.) festlegen kann.

Im folgenden Block-Editor kann man dann festlegen, was durch den neuen Baustein ausgeführt werden soll, und durch **Apply** bestätigen:



Aufgabe 4

Die Simpsons

Erstelle neue Bausteine **moveRight**, **moveLeft**, **moveUp** und **moveDown** und regle die Steuerung von Homer und Bart mit deren Hilfe.



Tipps und Tricks

- Um Sicherzugehen, dass deine Figur immer die richtige Laufrichtung hat, kann es nützlich sein, nicht die Bausteine **turn** zu verwenden, sondern folgenden Baustein:  Dieser sorgt dafür, dass sich die Figur – egal aus welcher Ausgangssituation – in die gewünschte Richtung dreht.
- Zum Entwickeln des Programms kann es nützlich sein, die Drehung der Bilder der Figuren zu erlauben, auch wenn man dies im Endprogramm ausstellen möchte. Wenn das Programm fertig entwickelt ist, kann man diese Drehung per Mausklick ganz einfach wieder verbieten.

