



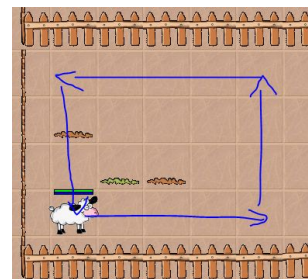
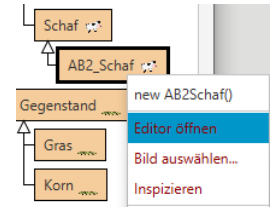
Du hast die ersten Schritte gemeistert. Allerdings hast du die Schafe bisher von Hand gesteuert. Wer kann schon einem Schaf den lieben langen Tag sagen, was es tun soll? Ok, ein Schäferhund vielleicht... Die Figuren sollen nun lernen, sich alleine zu bewegen.

Die Figuren lernen dazu ...

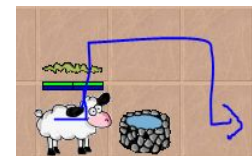
ZIEL: Wissen, dass alles, was die Figuren ausführen können, im Quelltext notiert ist. Vorhandene Quelltexte ergänzen und erweitern können.

Aufgaben:

1. Welche Befehle bietet dir ein **AB2-Schaf** in seinem Kontextmenü direkt an (also nicht „geerbt von“)? Öffne nun den Quelltext; diesen kannst du dir mit einem Rechtsklick auf die AB2-Klasse im rechten Fenster mit „Editor öffnen“ (oder alternativ Doppelklick auf die Klasse AB2Schaf) anzeigen lassen. Jede Fähigkeit ist in einer sogenannten Methode im Quelltext beschrieben.
2. Steuere das Schaf unten links noch ein letztes Mal von Hand durch einzelne Befehle so, dass es eine Runde dreht wie im Bild (siehe rechts). Die bekannten Befehle findest du jetzt im Kontextmenu unter „geerbt von Schaf“ bzw. „geerbt von Figur“. Welche Befehle hast du ihm dazu gegeben?



3. **Drehe Runde im Gatter:** Öffne den Quelltext der Klasse **AB2_Schaf**. Ergänze die Anweisungen in `dreheRunde()`, damit es eine vollständige Runde wird. Es soll danach auch möglich sein, mehrmals nacheinander den Befehl "dreheRunde" aufzurufen. Nach jedem Befehl musst du einen Strichpunkt setzen. (Hinweis: in rosa und grau findest du sogenannte Kommentare. Das sind Hinweise für dich und haben für das Schaf keine Bedeutung) Übersetze und erprobe die veränderte Methode `dreheRunde()`.
4. **Drehe um:** Schreibe im Quelltext die Anweisungen für `dreheUm()`.
5. **Fresse dich satt:** Bringe dem **AB2-Schaf** bei, drei Grasstücke abzufressen, die direkt hintereinander liegen. Du kannst davon ausgehen, dass sie schon genügend gewachsen sind. Teste die neue Methode am Schaf rechts unten. Welche Änderung am Code sind nötig, um zu beeinflussen, ob das Schaf mit dem Grasbüschel direkt unter ihm oder mit dem vor ihm beginnt.
6. **Haken schlagen:** Sorge dafür, dass das **AB2-Schaf** frei stehende Brunnen umlaufen kann (s. Bild rechts). Wie nennst du diese Fähigkeit eines **AB2-Schafs**? Das wird auch der Name der Methode, die du im Quelltext beschreibst. Der Methodenname sollte mit einem Kleinbuchstaben beginnen. Neue Methoden müssen immer mit `public void methodenname()` beginnen. Die Befehle der Methode werden dann in `{}`-Klammern eingeschlossen. Schaue dieses Konzept bei den bestehenden Methoden ab. Wenn ein **AB2-Schaf** genau vor einem Brunnen steht und einen Haken schlagen möchte, muss er sich z.B. nach links drehen, einen Schritt vor gehen, nach... Das kriegst du selbst raus. Erprobe deine neue Methode.





7. **Drehe Runde Variante 2:** Man kann auch die gerade erst selbst programmierten Methoden in eigenen Methoden nutzen. Implementiere dazu nochmal die Methode `dreheRunde`. Kopiere zunächst den ganzen Quelltext der Methode. Ersetze dann, wo immer möglich mehrere Befehle durch die Methode `dreiSchritte()`. Dadurch werden Programme kürzer und besser lesbar.
8. **Falsche Namensgebung:**
Genau zwei Namen sind unzulässig. Welche vermutest du? Warum?
`linksUm(); vor(); vierVor(); legeAb(); legeSpur(); linksum();`
`links um(); hebeAuf(); einsVor(); rechtsUm(); dreheUm(); lege3Ab(); sammle3Blaetter();`
`schiebeBaum(); zickzack(); 1Vor();`

Teste abschließend deine Implementierungen mit dem Befehl `checkup02()`, den du auf einem freien Feld mit der rechten Maustaste starten kannst. Wenn alles richtig ist, kannst du direkt zum nächsten Level gehen (`geheZuLevel()`) oder dich noch am Leveltest 2 versuchen.

Leveltest 2: Bring das Schaf in den Stall

Das Schaf muss abends in den Stall. Auf ihrer kargen Weide gab es nicht viel zu fressen und auch keinen Brunnen zum Trinken. Daher wird es sich auf dem Weg zu Stall stärken müssen.
Es kommt ohnehin an einem Brunnen vorbei und überquert eine saftige Wiese, die schon lange nicht mehr abgegrast wurde.



Implementiere für diesen Einsatz die Methode `leveltest2()` im Quelltext. Rufe dazu die zuvor erstellten Methoden in der richtigen Reihenfolge auf. Für den Methodenaufruf musst du nur ihren Methodennamen mit der `()`-Klammer dahinter hinschreiben (z.B. `dreheUm();`) Ergänze ggf. weitere Befehle (z.B. `einsVor();`). (Hinweis: du kannst mit unter 10 Befehlen auskommen!)

Um den Einsatz durchzuführen, musst du in der Bauernhof-Welt mit der rechten Maustaste auf einem beliebigen bräunlichen Hintergrundfeld die Methode `leveltest_02()` aufrufen (nicht direkt beim Schaf!).

Ich bin mal gespannt, ob du deine erste richtige Aufgabe bewältigst!

Falls du mit dem Einsatz Schwierigkeiten hat, kann dir dein Lehrer weiter helfen.

Zusammenfassung: Du kannst nun Programmieren – d.h. Methoden mit Anweisungen füllen. Dadurch kannst du Figuren Befehle geben, die sie dann selbständig ausführen!

Bildquellen: Die verwendeten Bilder des Bauernhof-Szenarios sind alle ohne Bildnachweis verwendbar (selbst gezeichnet, Pixabay Lizenz oder Public Domain). Genaue Nachweise: siehe [bildquellen.html](#).