

Leitfaden zum Erstellen eines „Frogger“-Spiels: Teil 2

Ziel dieser Dokumentation ist es, Sie beim Erstellen eines Frogger-Spiels zu unterstützen. Am Ende dieses Leitfadens haben Sie ein funktionierendes Computerspiel erstellt, welches Abbildung 1 ähneln sollte. Es werden einige Begriffe aus der Computersprache vorkommen, die allerdings erklärt werden, bevor sie im weiteren Kontext wiederverwendet werden.



Inhalt

1. Was müssen wir zum Spiel hinzufügen?
2. Das „Fluss“-Objekt
3. Wir erstellen graphisch Baumstämme und ein weiteres Hilfsobjekt (Brücke)
4. Wir programmieren die Baumstämme
5. Wir erstellen graphisch Schildkröten und ein weiteres Hilfsobjekt (Palme)
6. Wir programmieren die Schildkröten
7. Wir programmieren das Ziel

1. Was müssen wir noch hinzufügen?

Wenn wir unser Spiel mit der Abbildung unten vergleichen; was müssen wir noch hinzufügen?

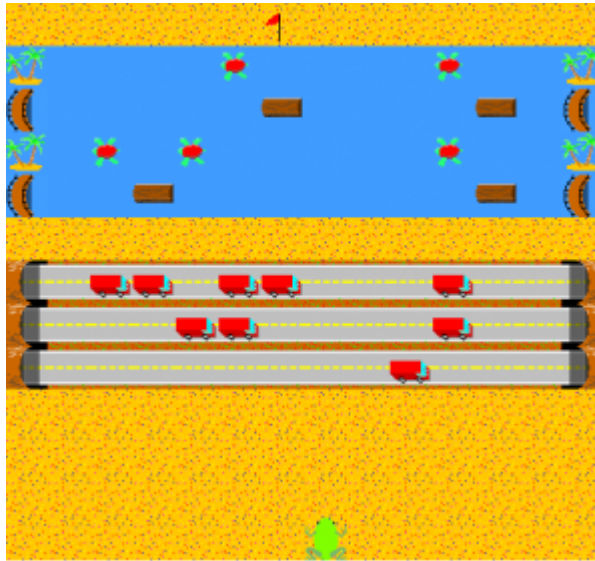


Abbildung 1. Fertiges „Frogger“-Spiel

- Fluss
- Baumstämme
- Hilfsobjekt (Brücken) für das Generieren von Baumstämmen Schildkröten -
- Hilfsobjekt (Palmen) für das Generieren von Schildkröten
- Ziel (Flagge)

Im Folgenden wollen wir diese neuen Objekte in unser Spiel einfügen.

- **Objekt: Fluss**
Der Frosch ertrinkt, wenn er in das Wasser hüpfet.
- **Objekt: Baumstamm**
Die Baumstämme treiben von links nach rechts auf dem Wasser.
Der Frosch kann auf die Baumstämme hüpfen.
Die Baumstämme verschwinden, wenn sie das Ende des Flusses erreichen.
- **Objekt: Hilfsobjekt** (Brücke), welches Baumstämme generiert
Die linken Brücken generieren die Baumstämme und die rechten Brücken löschen Baumstämme.
Es generiert Baumstämme, wenn rechts Wasser ist (Baumstämme schwimmen von links nach rechts).
- **Objekt: Schildkröten**
Die Schildkröten treiben von rechts nach links auf dem Wasser.
Der Frosch kann auf eine Schildkröte springen.
Schildkröten verschwinden, wenn sie das Ende des Flusses erreichen.
- **Objekt: Hilfsobjekt**, (Palme), das Schildkröten generiert.
Das rechte Hilfsobjekt generiert Schildkröten und das linke Hilfsobjekt löscht Schildkröten.
Es generiert Schildkröten, wenn links Wasser ist (Schildkröten schwimmen von rechts nach links).
- **Objekt: Frosch**
Wir passen den Frosch an.
Er kann auf Baumstämme sowie Schildkröten springen und wird von ihnen transportiert.
Der Spieler verliert, wenn der Frosch ins Wasser fällt (ertrinkt).

- **Ziel (Flagge)**

Wenn der Frosch die Zielflagge erreicht, gewinnt der Spieler.

2. Das „Fluss“-Objekt



Abbildung 1. Das „Fluss“-Objekt

In der Galerie (*gallery*), sprich unserem Graphikeditor, erstellen wir ein neues Objekt und nennen es Fluss. Dieser kann im einfachsten Fall ein grosses blaues Rechteck sein. Abbildung 2 veranschaulicht, wie das Fluss-Objekt aussehen kann.

Das Fluss-Objekt selbst hat kein Verhalten, es ist nur ein Bild. Andere Objekte, wie die Baumstämme oder Frösche, werden den Fluss als Hilfe für ihr Verhalten nutzen.

Anschliessend zeichnen wir die Fluss-Objekte auf das Worksheet (Spielfeld) als parallel zu der Strasse verlaufenden Fluss. Wir positionieren insgesamt 3 parallele Flussläufe.

Das Frosch- Verhalten anpassen:

Wenn sich der Frosch direkt auf einem Fluss-Objekt befindet, geht er unter und das Spiel ist beendet (ähnlich dem Vorgehen, wenn der Frosch von einem Fahrzeug getroffen wird und überfahren wird und das Spiel dadurch endet).

Wir ergänzen im Frosch-Programm eine Regel mit folgenden Befehlen:

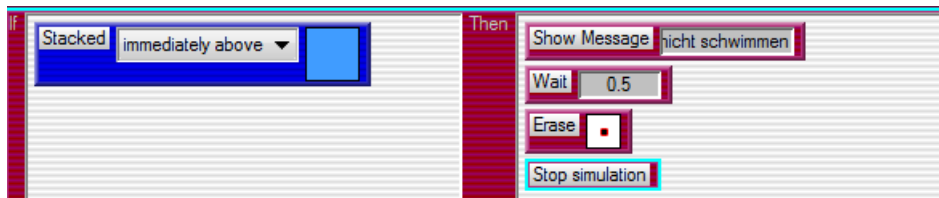
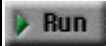


Abbildung 2. Der Frosch ertrinkt

Wenn sich der Frosch direkt über dem Fluss-Objekt befindet, wird das Spiel beendet und der Frosch wird gelöscht. Davor erscheint eine Textnachricht, die zum Beispiel lautet: „ich kann nicht schwimmen“. Durch die Zeitverzögerung von 0.5 Sek wird verhindert, dass die einzelnen Befehle so schnell abgespielt werden, dass die Textnachricht nur im Millisekundenbereich erscheint und gar nicht wahrgenommen werden kann. Durch den „erase“(löschen)-Befehl wird der Frosch gelöscht.



Programm-Test: Wir testen das Untergehen

Wir testen das Programm: Was passiert, wenn der Frosch ins Wasser springt?

- Bekommt der Spieler eine Nachricht, dass er etwas falsch gemacht hat (entweder durch einen Sound oder eine Nachricht)?
- Wird das Spiel beendet?
- Sollte etwas von diesen Möglichkeiten nicht funktionieren, prüfen Sie, wo der Fehler liegen könnte.
- Klappt alles? Perfekt gemacht! Nun erstellen wir die Objekte, die auf dem Fluss schwimmen und dem Frosch die Flussüberquerung ermöglichen.

3. Wir erstellen Baumstämme und ein weiteres Hilfsobjekt

Jetzt erstellen wir graphisch das „Baumstamm“-Objekt und das Hilfsobjekt, das Baumstämme generiert.



Abbildung 4. Das Baumstamm-Objekt



Abbildung 5. Hilfsobjekt

Das Hilfsobjekt soll Baumstämme generieren und löschen (genauso wie das Tunnel-Objekt im ersten Teil der Dokumentation, das Fahrzeuge generiert und wieder löscht). Hier im Beispiel wird dafür das Bild einer Brücke benutzt - es ist aber der Fantasie eines jeden überlassen, welche Darstellung sie/er für dieses Hilfsobjekt benutzt.

Wir programmieren das Verhalten des Hilfsobjektes, das Baumstämme generiert:

Baumstämme bewegen sich von rechts nach links; Das Hilfsobjekt generiert Baumstämme, sobald Wasser rechts vom Hilfsobjekt (Brücke) ist.

Im Folgenden wird die Regel für das Hilfsobjekt programmiert:

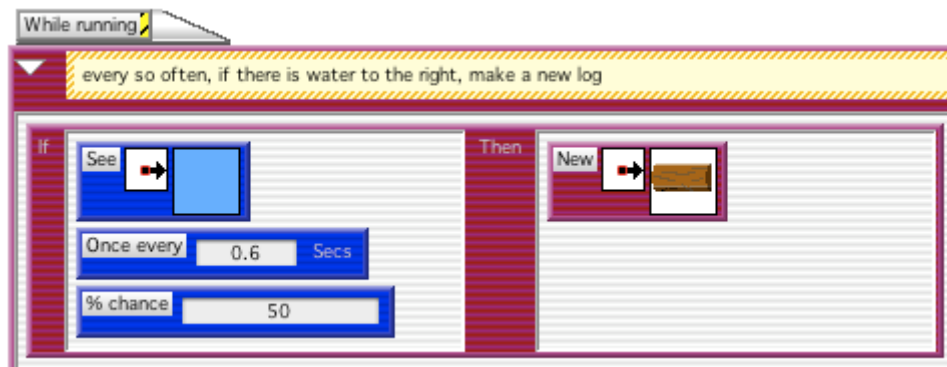


Abbildung 3. Die Brücke generiert einen Baumstamm.

Alle 0,6 Sekunden wird mit einer Wahrscheinlichkeit von 50% ein Baumstamm generiert. Durch die Wahrscheinlichkeit wird das Spiel spannender -- denn wenn Baumstämme in gleichen Abständen generiert würden, wäre es viel leichter, den Frosch über den Fluss zu navigieren.

An dieser Stelle können verschiedene Zeiteinheiten oder Wahrscheinlichkeiten ausprobiert werden und je nach Schwierigkeitsgrad für das Spiel angepasst werden.

4. Wir programmieren die Baumstämme

Das „Baumstamm“-Objekt bewegt sich von links nach rechts und es wird gelöscht, wenn ein Hilfsobjekt (Brücke) rechts von ihm ist.

Des Weiteren soll der Baumstamm den Frosch transportieren, wenn ein Frosch auf dem Baumstamm sitzt. Wir verwenden hierfür den Befehl „**transport**“, mit dem ein Objekt von einem anderen Objekt transportiert werden kann. Auch hier kann die Zeitsequenz an den Spiellevel angepasst werden. Je kürzer die Abstände, in denen neue Baumstämme generiert werden, desto einfacher das Spiel.

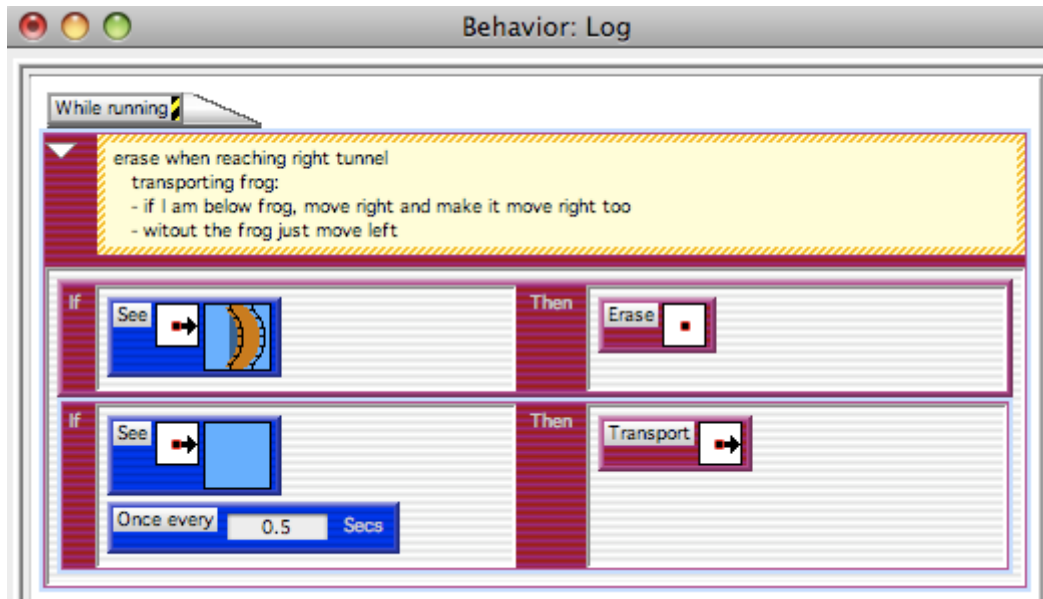
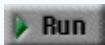


Abbildung 7. Verhalten des Baumstamms

Die erste Regel löscht den Baumstamm, wenn sich rechts vom Baumstamm das Hilfsobjekt (Brücke) befindet.

Die zweite Regel beschreibt, wie der Baumstamm sich selbst und Gegenstände, die sich auf ihm befinden, alle 0,5 Sek. einen Schritt nach rechts transportiert, solange rechts Wasser vorhanden ist.



Programm-Test: Wir testen die Baumstämme

Wir testen das Programm: Wie funktioniert das Hilfsobjekt, das Baumstämme generiert.

- Wir starten das Spiel mit dem grünen Pfeil für „play“. Entstehen Baumstämme?
- Bewegen sich die Baumstämme über den Fluss und verschwinden, wenn sie das Hilfsobjekt am Ende des Flusses (Brücke) erreichen?
- Wird der Frosch transportiert, wenn er auf einen Baumstamm springt?
- Wenn es funktioniert - SUPER - WEITER SO!

5. Wir erstellen graphisch Schildkröten und ein weiteres Hilfsobjekt



Abbildung 8 Das Schildkröten-Objekt



Abbildung 9 - Hilfsobjekt, welches Schildkröten generiert

Das Hilfsobjekt soll Schildkröten generieren und löschen. Hier im Beispiel werden dafür 2 Palmen gezeichnet - es ist aber der Fantasie eines jeden überlassen, welche Darstellung sie/er für dieses Hilfsobjekt benutzt.

Wir programmieren das Verhalten des Hilfsobjektes, welches Schildkröten generiert:

Schildkröten und die Hilfsobjekte zum Generieren von Schildkröten funktionieren analog zu den Baumstämmen. Das Hilfsobjekt generiert und löscht Schildkröten. Wählen Sie für das Hilfsobjekt eine geeignete graphische Darstellung (wie zum Beispiel Palmen auf einer Sandinsel). Die Schildkröten bewegen sich von rechts nach links (im Gegensatz zu den Baumstämmen, die sich von links nach rechts bewegen). Daher generiert das Hilfsobjekt die neuen Schildkröten so, dass sie sich nach rechts bewegen.

Im Folgenden wird die Regel für das Hilfsobjekt programmiert:

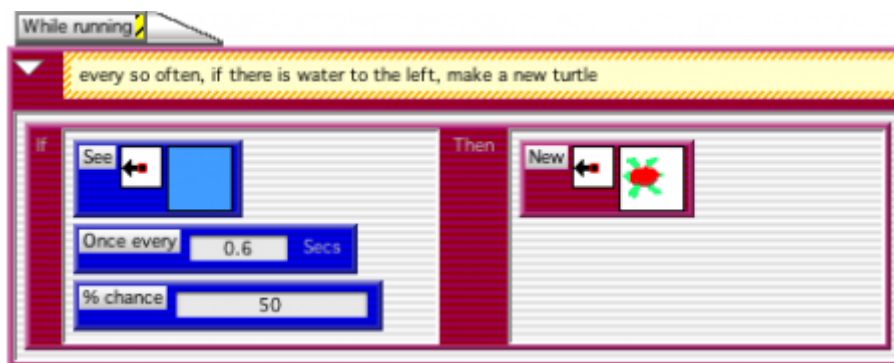


Abbildung 10. Die Palmen generieren eine Schildkröte.

Alle 0,6 Sekunden wird mit einer Wahrscheinlichkeit von 50% eine Schildkröte generiert. An dieser Stelle können verschiedene Zeiteinheiten oder Wahrscheinlichkeiten ausprobiert werden und je nach Schwierigkeitsgrad für das Spiel angepasst werden.

6. Wir programmieren die Schildkröten

Die Schildkröte bewegt sich von rechts nach links auf dem Fluss und wird gelöscht, wenn links ein Hilfsobjekt (Brücke) ist.

Des Weiteren soll die Schildkröte analog zum Baumstamm den Frosch transportieren. Wir verwenden hierfür wieder den Befehl „**transport**“. Auch hier kann die Zeitsequenz an den Spiellevel angepasst werden. Je kürzer die Abstände, in denen neue Schildkröten generiert werden, desto einfacher das Spiel.

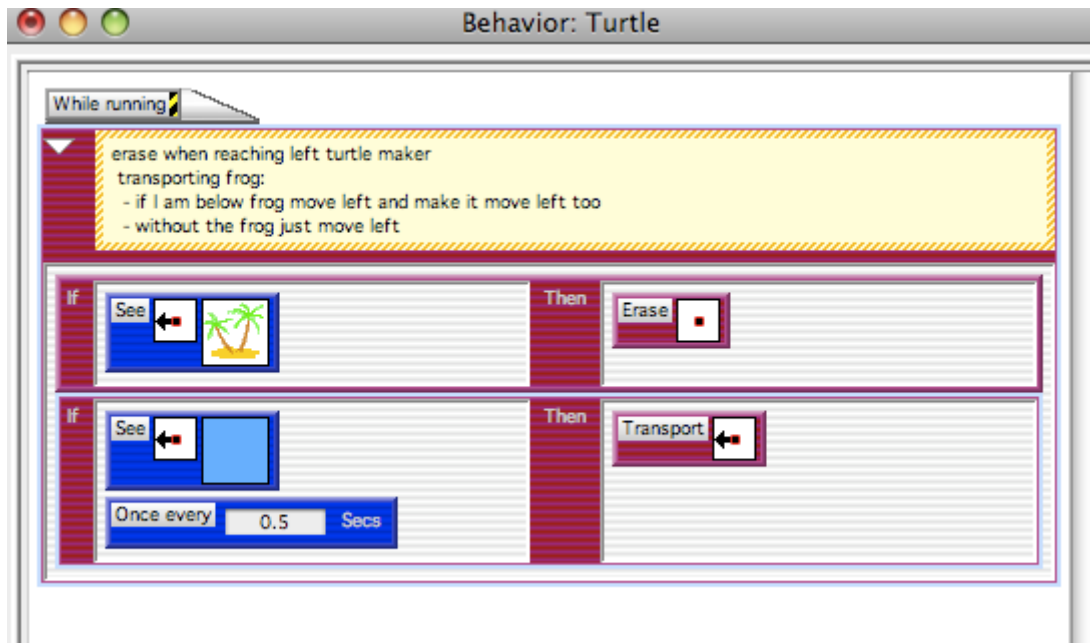
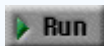


Abbildung 11. Verhalten der Schildkröte



Programmier-Test: Wir testen die Schildkröten

Wir testen das Programm: Wie funktioniert das Hilfsobjekt, welches Schildkröten generiert.

- Wir starten das Spiel mit dem grünen Pfeil für "play". Entstehen Schildkröten?
- Bewegen sich die Schildkröten über den Fluss und verschwinden, wenn sie das Hilfsobjekt (Palmen) erreichen?
- Wird der Frosch transportiert, wenn er auf eine Schildkröte springt?
- Wenn es funktioniert - SUPER - WEITER SO!

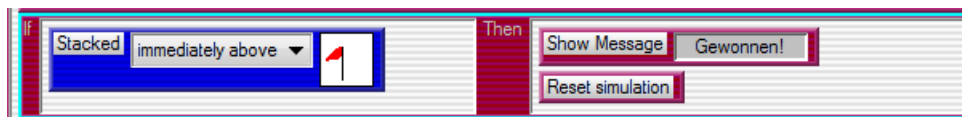
7. Wir programmieren das Ziel

Das letzte, was wir noch programmieren müssen, ist das Verhalten bei Spielgewinn. Der Spieler soll den Frosch zum Ziel (Flagge) navigieren, um das Spiel zu gewinnen und eventuell zum nächsten Spiellevel zu gelangen. Wenn der Spieler es schafft, den Frosch heil über die Straße und über den Fluss zu bewegen, erreicht er die Zielflagge und eine Textnachricht erscheint: „Du hast gewonnen!“.



Abbildung 12 Die Zielflagge

Wir erstellen graphisch die Zielflagge oder ein beliebiges Zielobjekt und programmieren dann die letzte Regel für den Frosch. Das Zielobjekt (z.B. die Flagge) wird auf dem Spielfeld oben platziert.



Wenn sich nun der Frosch auf Position von der Zielflagge befindet, erscheint eine Siegesnachricht, und da das Level damit beendet wird, wird Simulation gestoppt oder direkt ein neues Spiel initiiert („reset simulation“).

Run

Der letzte Programmier-Test

Wir testen unser Spiel ein letztes Mal.

- Wird bei Erreichen des Ziels die Nachricht angezeigt?
- Wird die Simulation gestoppt oder ein neues Spiel initiiert?
- Wenn es funktioniert - SUPER -!
- Es kann nun ein weiterer Level programmiert werden mit höherem Schwierigkeitsgrad.