



Hausaufgabe 9 – Breakout

In der Vorlesung haben Sie ein Programm gesehen, das einen durch das Programmfenster fliegenden und von den Seiten des Fensters abprallenden Ball zeichnet. Ausgehend von dem Programm aus der Vorlesung entwickeln Sie in dieser Hausaufgabe in mehreren Schritten eine Version des klassischen Computerspiels "Breakout".

Das Spiel soll sich insgesamt wie folgt verhalten:

Ein Ball fliegt durch das Programmfenster und prallt von den Seiten ab. Der Spieler kontrolliert einen Schläger am unteren Bildschirmrand. Der Schläger befindet sich an der horizontalen Position des Mauszeigers. Der Ball prallt an der oberen Kante des Schlägers nach oben ab.

Am Anfang des Spieles erscheint ein Willkommensbildschirm und wenn der Spieler verliert erscheint ein Game-Over-Bildschirm. Wenn der Spieler im Willkommensbildschirm mit der Maus klickt, beginnt das eigentliche Spiel. Wenn der Spieler im Game-Over-Bildschirm mit der Maus klickt, erscheint wieder der Willkommensbildschirm. Der Spieler verliert, wenn der Ball den unteren Bildschirmrand schneidet.

Im oberen Teil des Bildschirms befinden sich bunte Rechtecke. Der Ball prallt von den Rechtecken ab. Die Rechtecke verschwinden, wenn der Ball von ihnen abprallt. Wenn alle Rechtecke verschwunden sind, gewinnt der Spieler und kommt wieder in den Willkommensbildschirm.

Es gibt zahlreiche online verfügbare Varianten von "Breakout". Am besten spielen Sie mal eine Runde, um das Verhalten des Spieles kennenzulernen. Es ist auch interessant, sich kurz über die Geschichte dieses Spieles zu informieren, zum Beispiel auf Wikipedia.

Bearbeiten Sie dieses Hausaufgabenblatt alleine. Stellen Sie dabei sicher, dass Sie die gesetzten Lernziele erreichen. Verwenden Sie wieder die Sprache BSL (How to Design Programs – Anfänger).

Bitte geben Sie eine **ausführbare** Version ihres Spiels ab. Stellen Sie sicher, dass keine Fehler erzeugt werden, wenn Sie in DrRacket auf "Start" klicken. Beachten Sie, dass Programme, die nicht ausführbar sind keine Punkte geben. Achten Sie darüber hinaus darauf, dass wenn man das Spiel vom Interaktionsbereich aus startet sich zumindest ein Fenster öffnet. Wenn Sie bereits anderen Code geschrieben haben, den Sie auch gerne abgeben würden, der aber nicht funktioniert, können Sie diesen "auskommentieren".

Noch ein Hinweis: Wenn Sie Probleme damit haben die Aufgabenstellung zu verstehen oder nicht ganz wissen wie Sie vorgehen sollen können Sie auch gerne im Forum fragen!

Viel Erfolg!

Lernziele

Durch diese Hausaufgabe üben Sie noch einmal

- ✓ mit Produkttypen, Summentypen und Listen zu programmieren
- ✓ interaktive Programme mit `big-bang` zu erstellen und
- ✓ größere Programme aus vielen Hilfsfunktionen zusammenzusetzen.

Nach dieser Übung sollen Sie in der Lage sein, selbstständig interaktive Programme mit komplexem oder beliebig großem Zustand zu programmieren.

Bewertungsschema (Gesamtpunktzahl 100 pt)

- ☐ 30 pt Das Spiel kann ausgeführt werden und die im ersten Absatz beschriebene Funktionalität ist weitestgehend umgesetzt.
- ☐ 30 pt Das Spiel kann ausgeführt werden und die im zweiten Absatz beschriebene Funktionalität ist weitestgehend umgesetzt.
- ☐ 15 pt Das Spiel kann ausgeführt werden und die im dritten Absatz beschriebene Funktionalität ist weitestgehend umgesetzt.
- ☐ 25 pt Die für das Spiel programmierten Struktur- und Datendefinitionen, sowie die Funktionen sind gut strukturiert; der Programmaufbau ist leicht nachvollziehbar; die Testabdeckung ist gut; (Wie es jeweils von der Anwendung des Entwurfsrezeptes und DRY zu erwarten ist).

Schritt 1: Interaktive Programme mit Produkttypen

In dieser Aufgabe programmieren Sie den ersten Teil Ihrer Variante von "Breakout", also den ersten Absatz der Spielbeschreibung oben auf dem Blatt. Dabei wiederholen Sie das Programmieren von interaktiven Programmen mit Produkttypen, wie Sie es bereits auf dem 6. Hausaufgabenblatt geübt haben. Gehen Sie von der bereitgestellten Vorlage aus.

Empfohlenes Vorgehen

Wir schlagen vor, dass Sie in derselben Reihenfolge vorgehen, in der wir das Spiel oben beschrieben haben.

1. *Ein Ball fliegt durch das Programmfenster und prallt von den Seiten ab.*

Dieser Schritt wurde in der Vorlesung bereits implementiert und ist in der Vorlage bereits fertig. Vollziehen Sie die Funktionsweise des Programms nach.

2. *Der Spieler kontrolliert einen Schläger am unteren Bildschirmrand.*

(Zur Erinnerung: Eine ähnliche Funktionalität haben Sie bereits in Aufgabe 1 des 6. Hausaufgabenblattes programmiert.)

Erweitern Sie den Zustand des Spieles um die Informationen, wo sich der Schläger gerade befindet. Zeigen Sie den Schläger an. Bewegen Sie den Schläger bei jeder Mausbewegung entsprechend.

Am besten programmieren Sie dazu Struktur- und Datendefinitionen für einen neuen WorldState, der als Komponenten den Zustand des Balls und den Zustand des Schlägers enthält. Vergessen Sie nicht alle Stellen anzupassen, an denen auf den Zustand des Spiels zugegriffen wird, insbesondere in den Tests.

3. *Der Ball prallt vom Schläger ab.*

Erweitern Sie Ihr Programm um den Fall, dass der Ball den Schläger berührt.

Nach der Bearbeitung dieser Aufgabe haben Sie schon ein kleines Spiel in dem der Spieler versuchen muss, den Ball so lange wie möglich in der Luft zu halten. Setzen Sie die Geschwindigkeit des Balles so, dass es nicht zu einfach aber auch nicht zu schwer ist.

Hinweis

Kopieren Sie sich am besten die Datei mit dieser Version, bevor Sie mit den nächsten Hausaufgaben weitermachen. So haben sie auf jeden Fall ein funktionierendes Programm, auch wenn Sie die anderen Aufgaben nicht ganz fertig stellen.

Denken Sie auch immer daran Tests für Ihre Funktionen zu schreiben. Es lohnt sich Änderungen in kleinen Schritten durchzuführen und zwischen diesen Änderungen immer wieder die Tests auszuführen.

Schritt 2: Interaktive Programme mit Summentypen

In dieser Aufgabe programmieren Sie den zweiten Teil Ihrer Variante von "Breakout", also den zweiten Absatz der Spielbeschreibung oben auf dem Blatt.

Empfohlenes Vorgehen

Wir schlagen vor, dass Sie in derselben Reihenfolge vorgehen, in der wir das Spiel oben beschrieben haben.

1. *Am Anfang des Spieles erscheint ein Willkommensbildschirm und wenn der Spieler verliert erscheint ein Game-Over-Bildschirm.*

Programmieren Sie eine Datendefinition für einen Summentyp mit drei Alternativen. Verwenden Sie als eine der Alternativen den bisherigen WorldState (aus Aufgabe 1, Aufgabenteil 3). Die anderen Alternativen sollen den Willkommensbildschirm und den Game-Over-Bildschirm repräsentieren. Ändern Sie dann Ihr Programm so, dass dieser neue Summentyp als WorldState verwendet wird, und je nach Alternative entweder das Spiel, ein Willkommensbildschirm oder der Game-Over-Bildschirm angezeigt wird. Ändern Sie den initialen Zustand so, dass am Anfang der Willkommensbildschirm angezeigt wird.

2. *Wenn der Spieler im Willkommensbildschirm mit der Maus klickt, beginnt das eigentliche Spiel. Wenn der Spieler im Game-Over-Bildschirm mit der Maus klickt, erscheint wieder der Willkommensbildschirm.*

Wechseln Sie bei einem Mausklick von der Alternative für den Willkommensbildschirm zur Alternative für das eigentliche Spiel. Wechseln Sie bei einem Mausklick von der Alternative für den Game-Over-Bildschirm zu der Alternative für den Willkommensbildschirm.

3. *Der Spieler verliert, wenn der Ball die untere Bildschirmseite berührt*

Ändern Sie ihr Programm, so dass bei Kollision mit dem unteren Bildschirmrand in die Alternative für den Game-Over-Bildschirm gewechselt wird.

Hinweis

Kopieren Sie sich am besten die Datei mit dieser Version, bevor Sie mit der letzten Aufgabe weitermachen. So haben sie auf jeden Fall ein funktionierendes Programm, auch wenn Sie die letzte Aufgabe nicht ganz fertig stellen.

Nach der Bearbeitung dieser Aufgabe haben Sie schon eine fast funktionsfähige Variante von Breakout. Setzen Sie die Geschwindigkeit des Balles so, dass es nicht zu einfach aber auch nicht zu schwer ist nicht in den Game-Over-Bildschirm zu kommen.

Schritt 3: Interaktive Programme mit Rekursiven Datentypen

In dieser Aufgabe programmieren Sie den dritten Teil Ihrer Variante von "Breakout", also den dritten Absatz der Spielbeschreibung oben auf dem Blatt. Dabei wiederholen Sie das Programmieren mit Listen, wie Sie es bereits auf dem 8. Hausaufgabenblatt geübt haben.

Empfohlenes Vorgehen

Wir schlagen vor, dass Sie in derselben Reihenfolge vorgehen, in der wir das Spiel oben beschrieben haben.

1. *Im oberen Teil des Bildschirms befinden sich bunte Rechtecke.*

Erweitern Sie den WorldState, um die Position und Farbe aller Rechtecke zu speichern. Zeichnen Sie die Rechtecke. Passen Sie das bestehende Programm an Ihre Änderungen an.

2. *Der Ball prallt von den Rechtecken ab.*

Erweitern Sie ihr Programm, sodass der Ball von den Rechtecken abprallt.

3. *Die Rechtecke verschwinden, wenn der Ball von ihnen abprallt.*

Ändern Sie ihr Programm, um alle Rechtecke mit denen der Ball kollidiert ist verschwinden zu lassen.

4. *Wenn alle Rechtecke verschwunden sind, gewinnt der Spieler.*

Wechseln Sie zurück in den initialen Zustand, wenn keine Rechtecke mehr übrig sind.

Nach der Bearbeitung dieser Aufgabe haben Sie ein vollständiges Computerspiel programmiert, Herzlichen Glückwunsch! Wenn Sie das Spiel noch ein bisschen aufpolieren wollen, können Sie ja noch weitere Level hinzufügen, oder eines der Spielelemente, die Sie vielleicht von andere Breakout-Varianten her kennen.