



INFORMATIK II

TAFEL-ÜBUNGSBLATT 4

Ausgabe: Do, 22.05.2014 - **Abgabe: So, 01.06.2014 - 23:59 Uhr**

Informationen: **Die Abgabe muss in elektronischer Form erfolgen!**
d.h. Ihre Bearbeitung muss in den angegebenen Formaten bis vor Mitternacht ins CIS System hochgeladen werden.

Aufgaben [35 + 10 Bonus]

Notiz: Halten Sie sich beim Programmieren in Scheme an die eingeführte Konstruktionsanleitung:

- a) Kurzbeschreibung als Kommentar (mittels: "; ...");
- b) Signatur (der "Vertrag" mittels: "(: ...)");
- c) Testfälle (mittels "(check-within ...) " oder "(check-expect ...) " oder ...)
- d) Prozedur Gerüst und Rumpf

Benutzen Sie Hilfsprozeduren, wenn Teilprobleme gelöst werden müssen!

Notiz 2: Testfälle sind nicht notwendig in diesen Aufgaben! - sie Testen durch das Spielen.

4.1 SNAKE [15 + 5 Bonus]

Snake - eines der berühmten und ersten Computerspiele überhaupt existiert in Scheme! Das Programm liegt gut kommentiert im Downloadbereich und wurde in der VL besprochen. Erweitern Sie das Programm auf die folgende Art und Weise:

- a) Verhindern Sie, dass die Schlange rückwärts in sich selbst laufen kann. Dass es also nicht möglich ist, durch einen Tastendruck - oder eine Kombination von Tastendrücken - die Schlange rückwärts in sich selbst laufen zu lassen. Im Moment ist das leider noch möglich.
 - Stellen Sie sicher, dass einfaches Zurücklaufen nicht möglich ist (also, dass das Drücken der Taste gegen die aktuelle Laufrichtung keinen Effekt hat).
 - Stellen Sie auch sicher, dass das Zurücklaufen in den eigenen Körper auch durch das schnelle Drücken von mehreren Tasten nicht möglich ist (z.B. Im Moment ist es möglich, wenn die Schlange nach oben läuft, ganz schnell hintereinander "right" und dann "down" zu drücken, wodurch die Schlange direkt rückwärts in den eigenen Körper läuft und somit das Spiel zurückgesetzt wird).

Hinweis: Um letzteres zu verhindern, sollten sie den Game-Record mit einer Variable `next-dir` (oder ähnlich) erweitern. [5]

- b) Fügen Sie Hindernisse in das Spielfeld ein! Hindernisse sollen die Größe des Schlangenkopfs haben. - sie könnten zum Beispiel mit dem "#Symbol dargestellt werden – oder auch durch das Zeichnen eines Rechtecks mittels, zum Beispiel: `(rectangle 10 10 #solidblue)`.
 - Definieren Sie sich dafür eine Liste von Hindernissen (zum Beispiel als `(list-of segment)`), die Sie in den Game-Record integrieren sollten. Diese Liste sollte anfangs leer sein, sollte aber nach jedem gefressenem Stück Candy um ein Hindernis wachsen.

- Malen Sie diese Hindernisse auch auf den Bildschirm in das Spielfeld korrekt hinein. Benutzen Sie für das Malen eine weitere Faltung über die Liste der Hindernisse!
- Erlauben Sie nicht, dass die Schlange durch das Hindernis läuft. Insbesondere soll, falls der Schlangenkopf mit dem Hindernis zusammenstößt, das Spiel wieder einfach auf den Anfangszustand gesetzt werden.

[10]

- c) Bonus Aufgabe [5 Punkte]: Stellen Sie sicher, dass ein neues Stück Candy NICHT auf einem Hindernis landet.

Abgabe: Programm `snake-solution.rkt`

4.2 Star Wars Figher [20 + 10 Bonus]

Als zweites, finales Beispiel, was alles mit Scheme möglich ist, haben wir noch ein Star-Wars Fighter Spiel für Sie vorbereitet. Sie kontrollieren den Flieger der Rebellen und fliegen über den Todesstern. Gegnerische Tie-Fighter bedrohen Ihren X-Wing Fighter. Schießen Sie so viele wie möglich der Gegen ab - aber lassen Sie sich selbst nicht erwischen!

- a) Fügen Sie eine Score zum Spiel dazu in dem Sie im Game-Record eine ganzzahlige Score Komponente hinzufügen. Jeder abgeschossene Gegner soll Ihnen 42 Punkte einbringen. Jeder Schuss (2 Bullets) kostet einen Punkt. Jeder Gegner, der an Ihnen vorbeifliegt, kostet Sie 99 Punkte.
- Nutzen Sie eine helle Schrift und notieren Sie die aktuelle Score oben links (ähnlich wie bei dem Snake-Spiel)... sie brauchen dafür aber nicht einen extra Rahmen zu ziehen oder ähnliches.
 - Um zu zählen, wie viele Gegner im Moment getroffen wurden, können Sie einfach die Länge der Gegner Liste vor und nach dem Prüfen, ob ein Geschoss einen Tie-Fighter getroffen hat, bestimmen.
 - Ein Schuss wird durch einen Leertastendruck abgefeuert. Dabei können Sie direkt auch die Kosten mit berücksichtigen.
 - Die Kosten für einen Vorbeiflug sollten Sie entsprechend testen, in dem Sie prüfen ob ein Tie-Fighter den unteren Rand erreicht hat (das ist schon implementiert - nur im Moment wird der Tie-Fighter dann einfach aus der Liste genommen ohne weitere Konsequenz).

[10]

- b) Berücksichtigen Sie, dass unser X-wing Flugzeug nicht in einen Tie-Fighter fliegen darf. Wird eine Zusammenstoß signalisiert, soll der Spielstatus zunächst für 5 Sekunden angehalten werden und dann soll das Spiel wieder von neuen beginnen. Hinweis: Um das Spiel anzuhalten, nutzen sie einen zusätzlichen Timer, der das Ende des Spieles signalisiert - und eine Weile zählt - bis 5 Sekunden erreicht sind. [10]
- c) Fügen Sie eine weiter Schussvorrichtung hinzu - diese Schüsse sollen aus der Spitze des X-Wings abgefeuert werden - wobei ein Schuss aus einer Salve von 5 Schüssen bestehen soll, die sich radial nach vorne ausbreiten (z.B. -6, -3, 0, 3, 6 Grad). Fügen Sie die Schüsse einfach zu der Liste der übrigen Schüsse (`list-of bullet`) hinzu, damit diese nach dem Abfeuern wie die übrigen Schüsse behandelt werden. Diese Schüsse sollen mit der Taste "b" abgefeuert werden können und sollten 5 Punkte kosten [10 BONUS Punkte] .

Abgabe: Programm `sw-fighter-solution.rkt`