Übungsblatt 2

Musterlösung

Aufgabe 1 Mock-up

Das Spielfeld ist in Abbildung 1 dargestellt. Die Spielfigur (unten links) muss das Ziel (oben rechts) erreichen. Es gibt drei andere Personen, die jeweils immer einen Schritt geradeaus gehen. Kommen sie dabei am Ende des Gitters an, drehen sie sich zusätzlich noch um 180°, wodurch sie ab dann in die entgegen gesetzte Richtung laufen. Erreichen sie bei ihrem Zug die Gitterstelle, auf der die Spielfigur gerade steht, kommen sie ihr zu nahe (Abstand halten!) und sie hat verloren.

Die Spielsituation ist so gestaltet, dass die Spielfigur erst durch hin- und herlaufen etwas abwarten muss, bis sie an allen Personen vorbei zum Ziel gelangen kann. Dies wird dadurch erreicht, dass die Person unten rechts einen etwas längeren Weg hat, auf dem sie immer patrouilliert. Dadurch verschieben sich die Bewegungsphasen zwischen dieser Person und den beiden anderen so, dass erst nach einer kurzen Weile ein Weg für die Spielfigur frei wird.

Da zum aktuellen Zeitpunkt nur die Spielfigur tatsächlich gesteuert werden soll, wird nur dieses Objekt tatsächlich einer Konstanten (*player*) zugewiesen, damit in Aufgabe 2 mit ihm interagiert werden kann.

Abbildung 1: Das Spielfeld

```
15          new GameObject(0, 0, 0, "path-e");
16          new GameObject(1, 0, 0, "path-i");
17          new GameObject(2, 0, 1, "path-t");
18          new GameObject(3, 0, 0, "path-i");
19          new GameObject(3, 0, 0, "goal");
20          new GameObject(4, 0, 0, "bridge");
21          new GameObject(0, 1, 0, "path-e");
22          new GameObject(1, 1, 0, "path-i");
23          new GameObject(2, 1, 0, "path-x");
24          new GameObject(3, 1, 2, "path-e");
25          new GameObject(4, 1, 3, "water-l");
26          new GameObject(0, 2, 0, "path-e");
27          new GameObject(1, 2, 0, "path-i");
28          new GameObject(2, 2, 0, "path-x");
29          new GameObject(3, 2, 0, "path-i");
```

```
30          new GameObject(4, 2, 2, "path-e");
31          new GameObject(0, 3, 0, "path-e");
32          new GameObject(1, 3, 0, "path-i");
33          new GameObject(2, 3, 2, "path-l");
34          new GameObject(3, 3, 0, "water-l");
35          new GameObject(4, 3, 0, "water-i");
36          new GameObject(1, 0, 2, "claudius");
37          new GameObject(0, 1, 0, "laila");
38          new GameObject(3, 2, 2, "child");
39          final GameObject player = new GameObject(0, 3, 0, "woman");
```

Aufgabe 2 Richtungweisend

Wie in der Vorlesung wird die Abfrage der Tastatur in einer Endlosschleife durchgeführt. Die gedrückte Taste wird in einer Konstanten gespeichert, damit diese mehrfach verglichen werden kann, ohne immer auf eine neue Taste zu warten. Danach wird der Reihe nach mit den vier Tasten für die Richtungen 0-3 verglichen. Wenn eine Taste passt, wird die Spielfigur in die entsprechende Richtung gedreht und dann einen Schritt in diese Richtung bewegt. Da sich die Bedingungen alle gegenseitig ausschließen, könnten die *else*-Schlüsselworte in dem abgedruckten Code auch weggelassen werden, ohne dass sich das Verhalten ändern würde. Für Aufgabe 3 werden sie aber benötigt.

```
40          while (true) {
41              final int key = getNextKey();
42              if (key == VK_RIGHT) {
43                  player.setRotation(0);
44                  player.setLocation(player.getX() + 1, player.getY());
46              }
47              else if (key == VK_DOWN) {
48                  player.setRotation(1);
49                  player.setLocation(player.getX(), player.getY() + 1);
51              }
52              else if (key == VK_LEFT) {
53                  player.setRotation(2);
54                  player.setLocation(player.getX() - 1, player.getY());
56              }
57              else if (key == VK_UP) {
58                  player.setRotation(3);
59                  player.setLocation(player.getX(), player.getY() - 1);
61              }
65          }
```

Aufgabe 3 Fehlerabweisend

Mit den *else*-Schlüsselworten in Aufgabe 2 schließt ein erfolgreicher Vergleich die Ausführung aller nachfolgenden Vergleiche aus. Dies bedeutet insbesondere, dass dem allerletzten Vergleich noch ein *else*-Zweig angehängt werden kann, der nur zur Ausführung kommt, wenn alle bisherigen Vergleiche fehlgeschlagen sind, d.h. eine andere Taste als die vier akzeptierten gedrückt wurde. In diesem Zweig wird also der Fehler-Sound abgespielt. In den anderen vier Zweigen wird jeweils der Schritt-Sound abgespielt.

```
45                  playSound("step");
50                  playSound("step");
55                  playSound("step");
60                  playSound("step");
62              else {
63                  playSound("error");
64              }
```