

Snake Programmieren

Um Snake zu programmieren bekommt ihr ein Rahmenprogramm, welches einfach nur aus Sachen besteht, die wir bisher noch nicht hatten. Ihr schreibt die Funktionen **moveIt()**, **hitDot()**, **hitTail()**, **gameLost()**, **gameWon()**.

Aufgabe 1: **moveIt()**

```
fun moveIt(evt: KeyEvent?) {
```

Für moveIt() bekommen wir in unserer Funktion ein komisches **evt** mit. Was genau das ist müsst ihr nicht wissen. Ihr bekommt jetzt folgende Befehle, die für euch genügen:

```
KeyEvent.VK_LEFT
```

Dieser Befehl ist einfach nur der Tastendruck der Pfeiltaste links.

```
KeyEvent.VK_RIGHT
```

Dieser Befehl ist einfach nur der Tastendruck der Pfeiltaste rechts.

```
KeyEvent.VK_UP
```

Dieser Befehl ist einfach nur der Tastendruck der Pfeiltaste oben.

```
KeyEvent.VK_DOWN
```

Dieser Befehl ist einfach nur der Tastendruck der Pfeiltaste unten.

Die dazu passende if Anweisung lautet wie folgt:

```
if (evt?.keyCode == KeyEvent.VK_DOWN)
```

Sie fragt ob wir die obere Pfeiltaste gedrückt haben und macht dann darauf hin etwas.

Zudem haben wir eine Variable namens **myDirection**, die uns immer ausgibt in welche Richtung wir uns bewegen. Die Richtungen heißen wie folgt **lastDirection.Up** (für oben), **lastDirection.Down** (für unten), **lastDirection.Left** (für links) und **lastDirection.Right** (für rechts).

Um zu fragen in welche Richtung ihr wollt schreibt ihr einfach:

```
if (myDirection == lastDirection.Right)
```

Damit habt ihr erstmal alle Werkzeuge.

Was soll unsere Funktion aber genau machen ?

Die Funktion soll die Richtung wechseln wenn eine Pfeiltaste gedrückt wird. Wenn also die Pfeiltaste „oben“ gedrückt wird soll auch die **myDirection** nun oben sein. (dabei ist oben immer „lastDirection.Up“). Es soll aber auch gelten, dass wir uns nicht um 180° drehen können sollen. Wenn also die Richtung oben ist darf sie im nächsten Schritt nicht unten sein.

Snake Programmieren

Aufgabe 2: **hitDot()**, **hitTail()**

Unsere Schlange bewegt sich auf einem Koordinatenfeld. Wenn also die Schlange die selben Koordinaten wie etwas hat, so sind beide auf dem selben Feld.

Die Funktion `hitDot()` soll erkennen ob die Schlange den Punkt trifft.

Die Funktion `hitTail()` soll erkennen ob die Schlange sich selbst trifft.

Beide Funktionen geben einen Boolean zurück. „Wahr“ wenn das gilt und sonst „false“

Für diese Aufgabe habt ihr folgende Funktionen:

`getSnakeX()` `getSnakeY()` `getdotY()` `getdotX()`

Sie geben euch die X und Y Koordinate vom Punkt und von der Schlange.

Hinweis: `hitTail` ist besonders schwierig. Lasst die am Anfang einfach weg.

Aufgabe 3: **gameLost()**, **gameWon()**

In dieser Aufgabe sollt ihr die Funktionen `gameLost()` und `gameWon()` implementieren. Beide Funktionen geben einen Boolean zurück. Der Wert dieses boolean ist „true“ wenn gewonnen oder eben verloren gilt und sonst „false“.

Gewonnen: Gewonnen hat der Spieler, wenn seine Schlange die Länge 30 erreicht (also wenn der Array der Schlange an der 29. Stelle != null ist)

Verloren: Der Spieler hat verloren wenn `hitBorder()` oder `hitTail()` gilt.