technische universität
dortmund

Fakultät für Informatik
Lehrstuhl 14
Lars Hildebrand

Praktikum zu

Einführung in die Informatik für LogWiIngs und WiMas

Wintersemester 2018/19

Tower-Defense-Übungsblatt

Aufgabe 4: Türme zählen – um eine Koordinate herum

Ein Turm, den die Spieler in unserem Spiel bauen können, ist der Verstärkerturm. Dieser Turm greift Gegner nicht direkt an, sondern sorgt dafür, dass die Türme um ihm herum mehr Schaden verursachen. Damit der Turm jedoch nicht einfach alle acht Türme, die um ihm herum stehen, um den selben Wert verstärkt, verteilt er seinen Bonus auf alle Türme in seiner Reichweite. Steht ein einzelner Turm neben einem Verstärkerturm, wird er doppelt so stark, zwei Türme werden $1\frac{1}{2}$ mal so stark, drei Türme $1\frac{1}{3}$ mal so stark, etc.

Damit dies funktioniert, muss die Funktion **countTowersAround** implementiert werden. Sie können davon ausgehen, dass auf dem Feld in der Mitte ein Turm steht, der **nicht** mitgezählt werden soll.

Wie in Aufgabe 2 sollen Sie Türme zählen, diesmal erhalten Sie jedoch die Koordinaten x und y um die Sie das 3×3-Raster um die Position herum abzählen sollen. Die Hilfsmethoden der Vorgabe sind die selben. Verwenden Sie also erneut eine verschachtelte Schleife, aber zählen Sie nicht von 0 bis zur Grenze, sondern versuchen Sie mithilfe geschickter Addition die 9 Felder um die gegebene Koordinaten abzufragen.

Beispiel: rufen wir die Funktion für die Koordinaten $(4, 3)$ auf, sind die folgenden Koordinaten abzufragen.

$(3, 2)$	$(4, 2)$	$(5, 2)$
$(3, 3)$	$(4, 3)$	$(5, 3)$
$(3, 4)$	$(4, 4)$	$(5, 4)$

Achten Sie darauf, dass sich ein Verstärkerturm am Spielfeldrand befinden kann. In dem Fall dürfen Sie **towerOn** nur mit Koordinaten verwenden, die sich nicht außerhalb des Spielfeldes befinden. Im Folgenden sehen Sie zwei Beispiele für die Koordinaten $(0, 3)$ und $(3, 11)$ in einem 12×12 großen Spielfeld:

$(-1, 2)$	$(0, 2)$	$(1, 2)$	$(2, 10)$	$(3, 10)$	$(4, 10)$
$(-1, 3)$	$(0, 3)$	$(1, 3)$	$(2, 11)$	$(3, 11)$	$(4, 11)$
$(-1, 4)$	$(0, 4)$	$(1, 4)$	$(2, 12)$	$(3, 12)$	$(4, 12)$

Ein Aufruf mit den rot markierten Koordinaten als Parameter würde zu einem Fehler im Programm führen. Bauen Sie also entsprechende Tests ein, die die Grenzen des Spielfeldes überprüfen, bevor Sie **towerOn** aufrufen.

Lösung:

```
package aufgaben;

import model.Battlefield;

public class Aufgabe4 {
    public static int countTowersAround(int x, int y) {
        int towersAround = 0;

        for(int i = -1; i <= 1; ++i) {
            for(int j = -1; j <= 1; ++j) {
                if(x+i >= 0
                && x+i < battlefieldWidth()
                && y+j >= 0
                && y+j < battlefieldHeight()) {
                    if(towerOn(x+i, y+j)) {
                        towersAround++;
                    }
                }
            }
        }

        towersAround--; // wir ziehen den Turm in der Mitte ab

        return towersAround;
    }

    // folgende Hilfsmethoden duerfen gern verwendet werden
    /* ... */
}
```