Praktische Informatik I
Tutor:in: Rodrigue Wete Nguempnang

WiSe 2021/22
Bearbeiter:in: Moritz Salger [611554]

Übungsblatt 3

Lösungsvorschlag

Aufgabe 1 Einmal mit Klasse

"NPC" ist die Klasse mit den drei Attributen für diese Aufgabe

```
6  class NPC {
7      private final GameObject avatar;
8      private final int patrolDistance;
9      private int progress;
```

avatar	die Referenz zum *GameObject*.
patrolDistance	die Länge des Laufweges vom *NPC*.
progress	die Anzahl der Schritte des Laufweges, die der *NPC* beim Start schon zurück gelegt hat.

Da *patrolDistance* und *avatar* des *NPCs* nach der initialisierung nicht verändert werden, sind diese *final*.

Aufgabe 2 Konstruktivismus

```
12     public NPC(GameObject avatar, int patrolDistance, int progress){
13         this.avatar = avatar;
14         this.patrolDistance = patrolDistance;
15         this.progress = progress;
16 
17         if (progress>patrolDistance || progress < 1) {
18             this.progress = 1;
19         }
20     }
```

Im Konstrukter werden die drei Attribute aus Aufgabe 1. initialisiert,
die Werte hierfür werden als Parameter entgegengenommen

Zusätzlich wird geprüft ob auserhalb 1..patrolDistance liegt und wenn dies der Fall ist auf 1 gesetzt

Aufgabe 3 Schritt für Schritt

```
22      public void act(){
23          progress++;
24          if(progress>=patrolDistance){
25              returning = !returning;
26              progress = 1;
27          }
28          int _direction = returning?2:0;
29          Move(_direction);
30      }
```

In der Funktion *act* wird die *progress* Variable um 1 erhöt, darauf hin wird überprüft ob das Ende der Laufdistanze ereicht wurde. Falls dies der Fall ist wird progress auf 1 gesetzt und die *returning* Variable wird invertiert.
Am Ende wir noch die *Move* Function mit der korrekten Richtung aufgerufen um die Bewegung durch zu führen.

```
32      private void Move(int direction){
33          avatar.setRotation(direction);
34          int _x = avatar.getX();
35          int _y = avatar.getY();
36          switch(direction){
37              case 0:
38                  _x++;
39                  break;
40              case 1:
41                  _y++;
42                  break;
43              case 2:
44                  _x--;
45                  break;
46              case 3:
47                  _y--;
48                  break;
49          }
50          avatar.setLocation(MinMax(0, RPGGame.FIELD_X-1, _x), MinMax(0, RPGGame.FIELD_Y-1,
                _y));
51      }
```

Hier werden locale x und y Variablen erstellt und auf die aktuelle *NPC* position gesetzt. Darauf hin werden diese Abhängig von der *direction* Variable verändert, worauf hin *avatar.setLocation* mit diesen Werten aufgerufen wird um den *NPC* zu bewegen.
Die Werte werden durch eine *MinMax* Funktion auf das Spielfeld beschränkt damit *NPCs* nie aus dem Spielfeld laufen können.

Aufgabe 4 Und Action!

```
74          NPCs[0] = new NPC(new GameObject(1, 0, "claudius"), 3, 1);
75          NPCs[1] = new NPC(new GameObject(4, 1, "claudius"), 5, 4);
```

Als erstes werden zwei *NPCs* erstellt, und deren *act* Methode wird nach jedem Spieler Zug aufgerufen.

```
113             for(NPC npc: NPCs){
114                 npc.act();
```

Aufgabe 5 Bonusaufgabe

```
114                 npc.act();
115                 if(checkCollision(player.getX(), player.getY(), npc.getX(), npc.getY())){
116                     player.setVisible(false);
117                     System.exit(0);
118                 }
119             }
```

Nachdem sich ein *NPC* bewegt hat, wird überprüft ob die neue Position mit der des Spieler übereinstimmt. Falls dies der Fall sein sollte, wird die Spielfigur unsichtbar gemacht und das Spiel beendet.