

Übungsblatt 3

Lösungsvorschlag

Aufgabe 1 Aufgabe 1 Einmal mit Klasse

Der folgende Text und Code wurden von mir erstellt. Da ich kein Muttersprachler bin, habe ich den Text in ChatGPT eingefügt, um Wörter und Sätze zu korrigieren. ChatGPT hat den Inhalt inhaltlich nicht verändert, sondern nur sprachlich verbessert. Ob der Text nun etwas „roboterhaft“ klingt, kann ich nicht genau beurteilen, aber die Anzahl der grammatischen und sprachlichen Fehler ist deutlich geringer

In dieser Aufgabe wurde eine neue Klasse NPC erstellt, die ein nicht steuerbares Spielobjekt beschreibt. Sie enthält Attribute für die Figur, die maximale Anzahl der Schritte und den aktuellen Schritt.

```

1 /**
2  * Diese Klasse definiert <Zusammenfassung ergänzen>
3 *
4 * @author Ramtin
5 */
6 class NPC
7 {
8     // Aufgabe 1: Attribute
9     private GameObject character;
10    private int maxStep;

```

Beschreibung: Diese Attribute speichern, welche Figur der NPC steuert und wie weit sie sich auf ihrem Weg bewegen darf.

Aufgabe 2 Aufgabe 2 Konstruktivismus

Es wurde ein Konstruktor definiert, der die drei Attribute initialisiert.

```

1
2     public NPC(GameObject character, int maxStep, int startStep){
3         this.character = character;
4         this.maxStep = maxStep;
5         this.currentStep = startStep;

```

Beschreibung: Der Konstruktor weist jedem NPC sein Spielfigur-Objekt, die maximale Schritt-länge und die aktuelle Position zu.

Aufgabe 3 Aufgabe 3 Schritt für Schritt

Die Klasse wurde um eine Methode act() erweitert, die einen einzelnen Bewegungsschritt aus-führt. Die Figur läuft einen Schritt vorwärts, zählt diesen, und wenn die maximale Schrittzahl erreicht ist, dreht sie sich um und setzt den Zähler zurück.

```

1
2     public void act(){

```

```

3
4     if(character.getRotation() == 2){
5         // Schritt vorwärts machen
6         int newX = character.getX() - 1;
7         character.setLocation(newX, character.getY());
8     } else {
9         int newX = character.getX() + 1;
10        character.setLocation(newX, character.getY());
11    }
12
13    // Schrittzähler erhöhen
14    currentStep++;
15
16    // Prüfen, ob Maximalanzahl erreicht ist
17    if (currentStep >= maxStep){
18        currentStep = 0; // Schrittzähler zurücksetzen
19        character.setRotation(character.getRotation() + 2); //// Richtung umdrehen
20    }
21
22 }
```

Beschreibung: Der NPC bewegt seine Figur abwechselnd hin und her, indem er die Rotation um 2 erhöht, sobald die maximale Schrittlänge erreicht wurde.

Aufgabe 4 Aufgabe 4 Und Action!

In der Hauptmethode des Spiels (`NPCGame`) wurden mehrere NPCs erstellt, und deren `act()`-Methoden werden nach jeder Spielerbewegung aufgerufen.

```

1     final int key = getNextKey();
2     if (key == VK_RIGHT || key == VK_LEFT || key == VK_UP || key == VK_DOWN) {
3         if(key == VK_RIGHT) {
4             player.setLocation(player.getX()+1, player.getY());
5             player.setRotation(0);
6         }
7         if(key == VK_DOWN) {
8             player.setLocation(player.getX(), player.getY()+1);
9             player.setRotation(1);
10        }
11        if(key == VK_LEFT) {
12            player.setLocation(player.getX()-1, player.getY());
13            player.setRotation(2);
14        }
15        if(key == VK_UP) {
16            player.setLocation(player.getX(), player.getY()-1);
17            player.setRotation(3);
18        }
19
20        playSound("step");
21    }
22    else if (key == VK_SPACE){
23        isOn = false;
24        break;
25    }
26    else {
27        playSound("error");
28    }
29    sleep(200);
30
31    child.act();
32    laila.act();
33    laila.checkCollision(player);
34    child.checkCollision(player);
35
36    }
37
38
39    }
40 }
```

```
41     static void turn(final int direction){  
42         player.setRotation(player.getRotation() + direction);  
43         player.playSound("step");  
44     }  
45  
46 }
```

Beschreibung: Nach jedem Tastendruck bewegt sich der Spieler, danach führen die NPCs einen Schritt aus. Die Figuren reagieren also unabhängig vom Spieler.

Aufgabe 5 Aufgabe 5 Bonusaufgabe

Die NPC-Objekte überprüfen in ihrer Methode `checkCollision()`, ob sie die gleiche Position wie der Spieler erreicht haben. Falls ja, wird die Spielfigur unsichtbar und das Spiel endet.

```
1 //prufen NPC position mit Player position:  
2 if (character.getX() == player.getX() && character.getY() == player.getY()){  
3     player.setVisible(false);  
4 }  
5 }  
6  
7 }
```

Beschreibung: Sobald der NPC die gleiche Position wie der Spieler hat, wird der Spieler mit `setVisible(false)` entfernt und die Hauptschleife beendet sich automatisch.