

Ziel des Codes

Der Code beschreibt ein **Konsolenspiel**, in dem der Spieler die Positionen von drei versteckten Gegnern erraten muss. Die Gegner befinden sich auf aufeinanderfolgenden Positionen in einem Bereich (0–6), und der Spieler hat eine begrenzte Anzahl von Schüssen (5), um die Gegner zu treffen.

Das Ziel des Spiels ist es:

- **Alle Gegner zu eliminieren**, bevor die Anzahl der verfügbaren Schüsse aufgebraucht ist.
- Die Trefferanzahl und die verbleibenden Schüsse zu verfolgen.
- Dem Spieler Rückmeldung zu geben, ob seine Eingaben korrekt oder bereits geprüft wurden.

Programme

Das EnemykillGame muss diese Dinge tun:

1. Erstelle ein EnemykillObjekt.
2. Erstelle eine Position (drei aufeinanderfolgende Zellen auf einer einzelnen Zeile aus sieben virtuellen Zellen).
3. Benutzer zur Eingabe eines Rateversuchs auffordern.
4. Rateversuch überprüfen. Der User kann die selber Position nicht mehr verwenden.

Vorcode

EnemykillGame	Enemykill
	<pre>private ArrayList<Integer> enemyLocation private int choice; private int leftShoot=5; private int numberHit; private ArrayList<Integer> choiceList= new ArrayList<Integer>();</pre>
Void main()	<pre>playGame() setLocation() setChoice() controlChoice() compareChoice()</pre>

Spielablauf

1. Start des Spiels:

Das Spiel wird durch die Methode `playGame` in der Klasse `EnemyKillGame` gestartet, welche die Hauptklasse enthält.

2. Erstellung der Gegner-Liste:

Eine Liste der Gegnerpositionen (`enemyLocation`) wird generiert und als `ArrayList` gespeichert.

3. Erstellung der Wahl-Liste:

Eine Liste für die bisherigen Spieler-Eingaben (`choiceList`) wird ebenfalls als `ArrayList` erstellt.

4. Eingabe des Spielers:

Der Spieler wählt eine Position (einen Zielbereich) aus und gibt diese über die Konsole ein.

5. Speichern der Position:

6. Die gewählte Position wird in der Liste `choiceList` gespeichert.

7. Analyse der Position:

Die eingegebene Position wird durch die Methode `controlChoice` überprüft:

- **Wenn die Position bereits in `choiceList` enthalten ist:**

Der Spieler wird informiert, dass die Position bereits geprüft wurde. Das Programm fordert ihn auf, eine neue Position einzugeben.

- **Wenn die Position nicht in `choiceList` enthalten ist:**

Die Position wird der Liste hinzugefügt und die nächste Prüfung erfolgt.

7. Überprüfung auf Treffer:

- Wenn die Position in der Liste `enemyLocation` enthalten ist:

- Der Spieler erzielt einen Treffer und erhält einen Punkt.
- Die verbleibende Anzahl der Schüsse (`leftShoot`) wird um 1 reduziert.
- Die Position wird aus der Liste `enemyLocation` entfernt.

- Wenn die Position nicht in der Liste `enemyLocation` enthalten ist:

- Der Spieler wird informiert, dass kein Gegner auf dieser Position ist.
- Die verbleibende Anzahl der Schüsse wird ebenfalls um 1 reduziert.

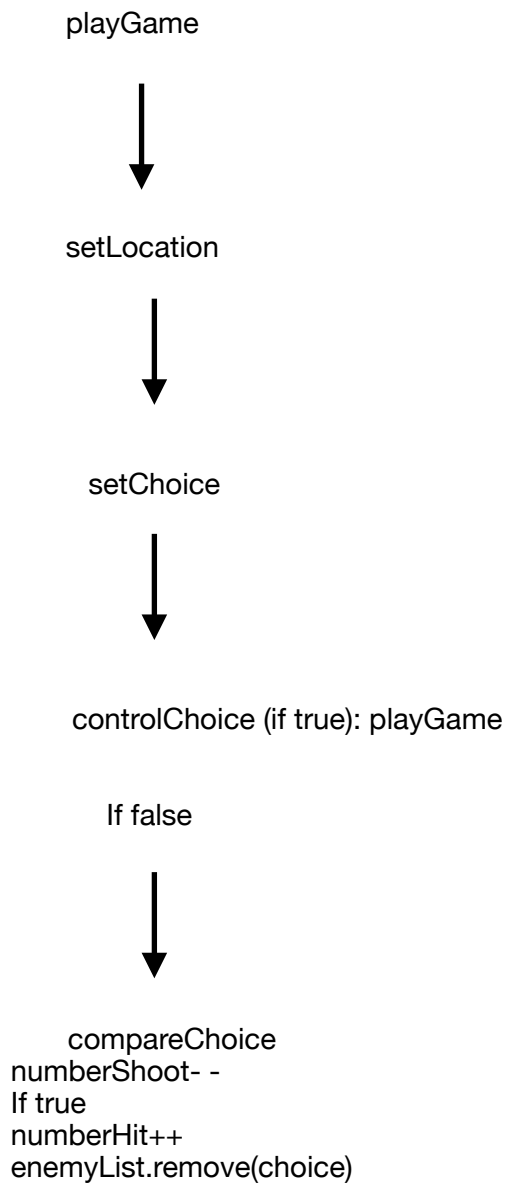
8. Ende des Spiels:

Das Spiel endet in zwei Fällen:

- **Alle Gegner wurden gefunden:**

Der Spieler gewinnt und wird entsprechend informiert.

- **Keine Munition mehr übrig:**
Der Spieler verliert, da die Anzahl der Schüsse aufgebraucht ist.



Beispielsausgabe

```
gautiernoumfie@host10-21-138-56 EnemieKill % /usr/bin/env /Library/Java/JavaVirtualMachines/  
jdk-23.jdk/Contents/Home/bin/java --enable-preview -XX:+ShowCodeDetailsInExceptionMessages -c  
p /Users/gautiernoumfie/Library/Application\ Support/Code/User/workspaceStorage/07484f7099f64  
27b7aaf0bd2b1ed1424/redhat.java/jdt_ws/EnemieKill_3c7aaeef/bin EnemykillGame  
Wellcome to enemy kill my general  
Enemys base are between us zone 0 to 6 but we are not sure  
Give the coordinates of the shoot  
0  
Nothing here  
Try aigain  
Give the coordinates of the shoot  
0  
We have already check this position  
Give the coordinates of the shoot  
1  
Nothing here  
Try aigain  
Give the coordinates of the shoot  
3  
Bingo  
Give the coordinates of the shoot  
4  
Bingo  
Give the coordinates of the shoot  
5  
Bingo  
We have killed all the enemy  
We don't have Missile anymore my general  
gautiernoumfie@host10-21-138-56 EnemieKill %
```