1/5

Aufgabe 3: Wortsuche-Dokument

Team-ID: 00070

Team: Gaußgamz

Bearbeiter/-innen dieser Aufgabe:  
Max Humbert, Nils Stäcker

1. Oktober 2021

Lösungsidee 1

Umsetzung 1

Beispiele 2

Quellcode 3

Der Code zum Einlesen der Datei: 3

Hier ein Beispiel zum Einfügen der Wörter: 3

Und noch der Code, der unser Level3 ausführt: 4

# Lösungsidee

Für die Aufgabe haben wir uns drei verschiedene Schwierigkeitsstufen ausgedacht.

In der einfachsten Schwierigkeitsstufe können die Wörter entweder vertikal oder horizontal im Wortgitter verteilt werden. Die Ausrichtung der Wörter ist hierbei nur von links nach rechts bzw. von oben nach unten möglich.

In der zweiten Schwierigkeitsstufe werden die Wörter dann noch gespiegelt, so dass die Wörter nun auch von rechts nach links bzw. von unten nach oben vorkommen können.

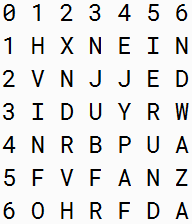
In unserer schwersten Schwierigkeitsstufe werden nun die Wörter auch noch diagonal ins Gitter eingefügt. Hierbei ist es möglich, dass die Wörter von links oben nach rechts unten oder von rechts oben nach links unten vorkommen. Auch können die Wörter dabei zusätzlich gespiegelt sein.

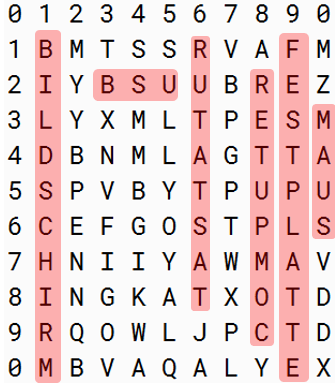
# Umsetzung

Um das Wortgitter dazustellen, wird ein zweidimensionales Char-Array erstellt, welches zunächst mit Platzhaltern gefüllt wird. Bevor wir mit dem Einfügen der Wörter beginnen, werden erstmal die Wörter der Länge nach sortiert und mit dem längsten Wort angefangen. Die Wörter werden völlig zufällig verteilt und horizontal, vertikal und diagonal in das Wortgitter eingefügt. Die Zufälle erzeugen wir mithilfe der „java.util.Random“-Klasse, die eine Zufallszahl generiert. Im schwersten Schwierigkeitsgrad wird zuerst zufällig entschieden, ob wir das Wort spiegeln wollen (fällt bei Level 1 weg, da diese Spiegelung, dafür zuständig ist, dass es auch Wörter von rechts nach links oder von unten nach oben gibt, die es bei Level 1 nicht geben soll). Danach wird wieder mit einer Zufallszahl bestimmt, ob das Wort horizontal, vertikal oder diagonal eingefügt wird. Wo es eingefügt wird, hängt wiederum auch vom Zufall ab. Nachdem alle Wörter eingefügt wurden, werden die restlichen Platzhalter in unserem Array durch zufällige Buchstaben aus dem Alphabet ersetzt.

# Beispiele

Hier ist ein Beispiel einer Lösung unseres Algorithmus von „worte1.txt“ mit der Schwierigkeitsstufe 1. Die Wörter sind nur von links nach rechts oder von oben nach unten verteilt.





Das ist ein Beispiel von „worte2.txt“ mit der Schwierigkeitsstufe 2. Hierbei sind die Wörter jetzt zusätzlich auch von rechts nach links und von unten nach oben angeordnet.



Das letzte Beispiel ist ein Beispiel mit „worte3.txt“ mit der 3. Schwierigkeitsstufe. Hierbei werden die Wörter auch diagonal im Feld verteilt. In diesem Beispiel ist es Zufall, dass 8 von 12 Wörtern diagonal sind.

# Quellcode

## Der Code zum Einlesen der Datei:

private void einlesen(String input){

File file = new File(input);

lines = new ArrayList<String>();

try{

Scanner scanner = new Scanner(file);

while(scanner.hasNext()){

lines.add(scanner.nextLine());

}

} catch(FileNotFoundException e){

e.printStackTrace();

}

parts = lines.get(0).split(" ");

## Hier ein Beispiel zum Einfügen der Wörter:

private boolean horiEinfg(String wort){

boolean passt = true;

char bst[] = wort.toCharArray();

Random random = new Random();

row = random.nextInt((zeile - bst.length + 1));

coloum = random.nextInt(spalte);

while(passt){

for(int i = 0; i < bst.length; i++){

if(Gitter[row+i][coloum] == '#' || Gitter[row+i][coloum] == bst[i]){

passt = true;

}else{

passt = false;

break;

}

}

break;

}

if(passt == true){

for(int i = 0;i < bst.length; i++){

Gitter[row+i][coloum] = bst[i];

}

return true;

} else{

return false;

}

}

## Und noch der Code, der unser Level3 ausführt:

public void level3(){

reset();

Random random = new Random();

for(int i = 2; i < (Integer.parseInt(lines.get(1)) + 2); i++){

for(int j = 0; j < 50; j++){

int zufall2 = random.nextInt(2);

if(zufall2==0){

int zufall = random.nextInt(4);

if(zufall == 0){

if(vertEinfg(lines.get(i)) == true){

break;

}

} else if(zufall == 1){

if(horiEinfg(lines.get(i)) == true){

break;

}

}else{

if(diagonalEinfg(lines.get(i)) == true){

break;

}

}

}else{

drehen(i);

int zufall = random.nextInt(4);

if(zufall == 0){

if(vertEinfg(lines.get(i)) == true){

break;

}

} else if(zufall == 1){

if(horiEinfg(lines.get(i)) == true){

break;

}

}else{

if(diagonalEinfg(lines.get(i)) == true){

break;

}

}

}

}

}

randomAuffüllen();

zeigen();

}