

## **Dokumentation**

### **Spielfunktionen:**

- Grafik mit AWT und Swing
- Einspieler-Modus
- Zweispieler-Modus
  - Mit einer Tastatur
  - Über Netzwerk
- Speichern und Laden von Spielständen
- Level-Editor mit Validitätsprüfung
- Extra Gegenstände
- Sound und Musik

### **Grafik**

Das Spiel benutzt AWT und Swing. Swing ist größtenteils nur für das Windowsfenster mit Menü zuständig. Mittels AWT Graphics werden Bilder im Speicher vorbereitet und dann auf den Bildschirm gezeichnet. Dieses Verfahren heißt Doublebuffering.

### **Spielfeld**

Das Spielfeld besteht aus einem char-Array. Wobei ein Feld im Array unterschiedliche Zustände annehmen kann:

'a' ist Wand

'p' ist Spieler 1

'l' ist Spieler 2

'g' ist Ziel

'b' ist Kiste

'x' ist Kiste hinter der sich ein Ziel verbirgt

'z' ist der Gegenstand mit dem Bomben-Extra

'u' ist der Gegenstand mit dem Bombenradius-Extra

Ansonsten leeres Feld

### **Levels**

Levels sind in Textdateien gespeichert und nutzen die Struktur des Spielfelds. Wobei jede Zeile in der Textdatei einer Stein-Reihe im Spielfeld entspricht.

### **Steuerung**

Das Spiel wartet auf Tastatur-Events und merkt sich die Tasten, die gerade gedrückt sind. So ist es möglich die Figur zu bewegen ohne immer wieder erneut auf eine Taste zu drücken.

## **Netzwerk-Modus**

Der Netzwerk-Modus wird mit Hilfe der einer TCP/IP Verbindung ermöglicht. Diese wartet auf Port „1234“ auf eine Verbindung des Gegenübers. Wenn eine Verbindung aufgebaut wird, werden die Bewegungsbefehle von Spieler 1 an Spieler 2 gesendet und umgekehrt.

## **Speichern und Laden von Spielständen**

Der Spielstand wird in der Datei „res/savegame“ gespeichert. Hier wird das aktuelle Spielfeldarray gespeichert und am Ende stehen folgende Informationen: Anzahl der Bomben, Bombenradius, Leben und Levelnummer.

## **Level-Editor**

Der Level-Editor ist ein reagiert auf Mausevents. Er kann Levels laden und speichern und man kann Felder manipulieren. Außerdem prüft er ob der Spieler das Ziel erreichen kann und sorgt dafür, dass es nur ein Ziel, ein Spieler 1 und ein Spieler 2 gibt.

## **Ressourcen-System**

Mit Hilfe von Maps kann man im Programm leicht auf Ressourcen zugreifen. Man benötigt lediglich den Namen einer Ressource und kann diese dann laden. Als Ressourcen gelten Bilder, Musikdateien und Levels.

## **Sound und Musik**

Für den Sound und die Musik wurden extra Klassen von Java genutzt. Für die Musik werden MIDI Dateien mit Hilfe von javax.sound.midi.MidiSystem geladen und für den Sound WAV Dateien mit Hilfe von javax.sound.sampled.AudioSystem.

## **Bomben**

Für jede Bombe wird eine Klasse geladen. Wenn eine Bombe aktiv wird zählt ein Counter eine Zahl hoch. Nach zwei Sekunden

explodiert die Bombe dann und erzeugt mögliche Kettenreaktionen.  
Nach 3 Sekunden verschwindet die Bombe wieder.