

Technisches Wahlpflichtfach „MineSweeper®“-Clone

als Umsetzung des Entwurfskomplexes „Dynamische Anzeige zur
Entscheidungsunterstützung“

Studenten : Martin Amelsberg
Daniel Finger

Professor : Rolf Socher-Ambrosius

Projektname : MindSweeper

Datum : 01. März 2003

Inhaltsverzeichnis

1. Studentendaten	3
2. Projektbeschreibung	
2.1. Allgemein.....	4
2.2. Das Spiel „MineSweeper®“	4
2.3. Änderungen / Unterstützungssysteme.....	5
2.3.1. Unterstützungssystem 1: „Aufmerksamkeitslenkung“	6
2.3.2. Unterstützungssystem 2: „Risikovisualisierung“	6
2.3.3. Unterstützungssystem 3: „Repräsentanzdarstellung“	7
3. Wiederverwendung von Quellcode	8
3.1. Was ist Wiederverwendung?.....	8
3.2. Vor- und Nachteile der Wiederverwendung.....	8
4. Vom „Open Source“-Programm zum Clone	
4.1. Das Spiel „Starsweeper“	9
4.2. Erforderliche und relevante Anpassungen.....	9
4.3. Unterschiede zu „MineSweeper“	11
5. Aufbau der Statistik-Datei	
5.1. Allgemein.....	12
5.2. Der Aufbau.....	12

1. Studentendaten

An diesem Projekt sind die Studenten¹

Martin Amelsberg, Matrikel-Nr. 2083918, Amel@AmelFin.de

Daniel Finger, Matrikel-Nr. 2084610, Fin@AmelFin.de

der Fachhochschule OOW Emden beteiligt.

Die Betreuung übernahm Professor Rolf Socher-Ambrosius.

Zur Vereinfachung wurde eine eMail-Adresse eingerichtet, die alle eingehenden Nachrichten an uns weiterleitet:

MindSweeper@AmelFin.de

Auf unserer Homepage

<http://www.AmelFin.de>

finden Sie weitere Informationen und Downloads zu unseren Projekten².

¹ Zu Beginn der Arbeiten war noch der Student Lars Raap am Projekt beteiligt, der jedoch aus persönlichen Gründen ausschied.

² Dieses Projekt wird in Absprache mit Dipl. Des. Philip Zerweck erst nach Fertigstellung des DFG-Projektes auf unserer Homepage veröffentlicht werden.

2. Projektbeschreibung

2.1. Allgemein

„Dynamische Anzeige zur Entscheidungsunterstützung am Beispiel ‚Minesweeper‘ “ ist ein Abschnitt der Arbeit „Beitrag eines Designers zu Benutzerschnittstellen von Leitwarten, Dokumentation des Arbeitsanteils von Dipl. Des. Philip Zerweck am DFG-Projekt ‚Dynamisierte Abbildung technischer Systeme in Leitwarten-SW unter dem Aspekt einer ergonomisch komfortablen Prozessvisualisierung‘ “.

Das DFG-Projekt und der Arbeitsanteil von Dipl. Des. Philip Zerweck werden unter Prof. Dr.-Ing. Gunnar Johannsen im Fachgebiet Systemtechnik und Mensch-Maschine-Systeme, Fachbereich Maschinenbau, Universität Kassel erarbeitet.

Aufgabe der Studenten war es, das bekannte Spiel „MineSweeper“ nachzuprogrammieren und mit einigen Änderungen zu versehen.

Das Wahl des Betriebssystems und die eingesetzte Programmiersprache wurde den Studenten überlassen.

2.2. Das Spiel „MineSweeper“

Bei diesem Spiel geht es darum, in einer möglichst kurzen Zeit auf einem $n*m$ -Spielfeld eine gewisse Anzahl von Minen aufzuspüren.

Hierfür werden Quadrate im Spielfeld aufgedeckt:

- Handelt es sich um ein vermintes Quadrat, so ist das Spiel verloren.



Abbildung 1: Aufgedeckte Minen; das Spiel ist verloren

- Handelt es sich um ein nicht-vermintes Quadrat, so wird die Anzahl von Minen in den benachbarten acht Quadraten angezeigt.

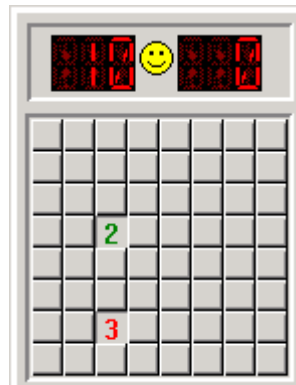


Abbildung 2: Aufgedeckte Zahlen geben die Anzahl der Minen wieder. Bei der 3 befinden sich 3 Minen in den umliegenden 8 Quadraten.

Weitere Anzeigen sind ein Counter für die Minen und eine Uhr:



Abbildung 3: LED-Anzeigen für die Anzahl der Minen und die Spielzeit

2.3. Änderungen / Unterstützungssysteme

Die genauen Definitionen zu den Unterstützungssystemen entnehmen Sie bitte der Ausarbeitung „Anschauliche Beispiele der Unterstützungssysteme“.

An dieser Stelle sollen die Unterstützungssysteme nur kurz erläutert werden.

2.3.1. Unterstützungssystem 1: „Aufmerksamkeitslenkung“

Irrelevante Informationen werden durch „Verblässung“ in den Hintergrund gestellt, so dass sich der Benutzer auf die am Spielgeschehen beteiligten Felder konzentrieren kann.



Abbildung 4: Für den Spielverlauf unerhebliche Felder werden in den Hintergrund gesetzt

2.3.2. Unterstützungssystem 2: „Risikovisualisierung“

Durch einen Farbverlauf von Rot bzw. einem Grün-Ton wird die Wahrscheinlichkeit eines Feldes für das Vorhandensein einer Mine dargestellt.

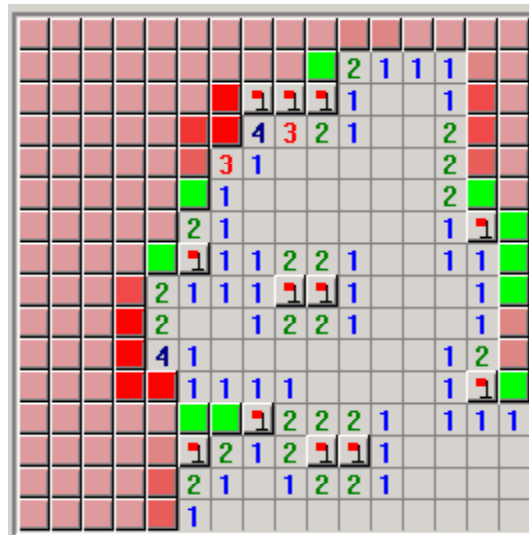


Abbildung 5: Unaufgedeckte Felder werden mit einer Rotfärbung bzw. einem Grün-Ton versehen, um so das Risiko zu visualisieren.

Nicht gekennzeichnete Felder treten auf, wenn der Benutzer offensichtlich falsche Eingaben gemacht hat.

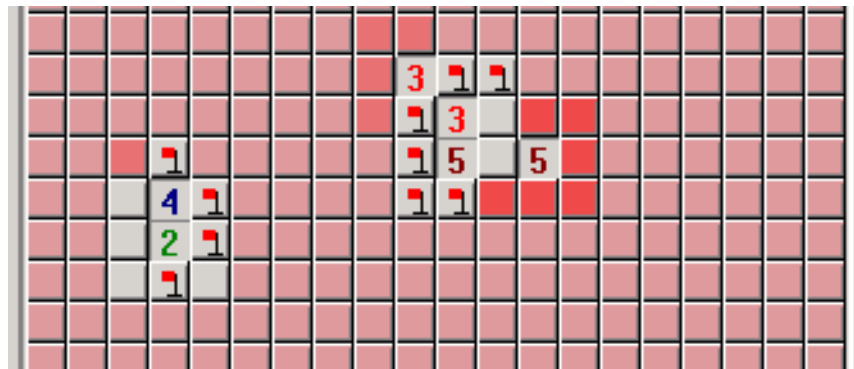


Abbildung 6: Da der Benutzer bspw. bei der aufgedeckten „2“ mehr Flaggen gesetzt hat als möglich sind, werden die übrigen Felder grau dargestellt.

Berechnet werden die Prozentwerte für jedes einzelne Feld nach folgendem Schema:

- ist ein Feld bspw. mit einer „3“ beschriftet und sind die umliegenden 8 Felder verdeckt, so befindet sich unter jedem dieser Felder mit einer Wahrscheinlichkeit von

$$\frac{3 \text{ Minen}}{8 \text{ verdeckte Felder}} = 0,375 \approx 38\%$$

eine Mine.

- werden für ein Feld unterschiedliche Prozentzahlen berechnet, so gilt der höhere Wert
- die Werte 0% und 100% gelten als „unantastbar“ und können auch durch andere Berechnungen nicht mehr verändert werden

2.3.3. Unterstützungssystem 3: „Repräsentanzdarstellung“

Der Benutzer kann seine Aktionen zusätzlich durch verschiedene Cursor (ändern sich bei bestimmten Tasten) verdeutlichen:

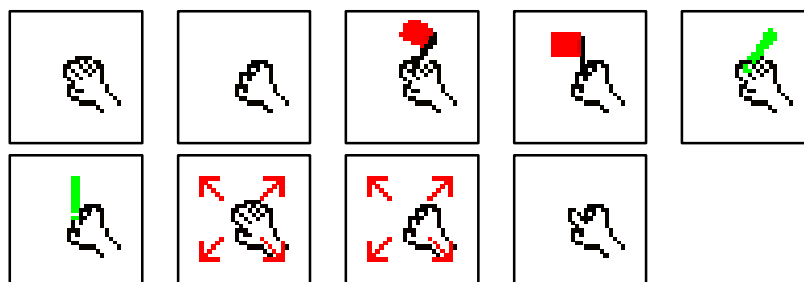


Abbildung 6: Alle möglichen Cursor der Repräsentanzdarstellung

3. Wiederverwendung von Quellcode

An dieser Stelle soll erklärt werden, was „Wiederverwendung von Quellcode“ ist, wo die Vor- und Nachteile liegen und wie wir dieses Mittel genutzt haben.

3.1. Was ist Wiederverwendung?

Hierbei geht es darum, auf einem bestehenden Quellcode aufzubauen und nicht „von Grund auf“ neu zu beginnen.

Es kann sich dabei um eigene Routinen (Bibliotheken) handeln, aber auch Programme von fremden Programmierern genutzt werden.

3.2. Vor- und Nachteile der Wiederverwendung

Ein großer Vorteil ist die Zeit- und damit die Kostenersparnis, da ggf. komplizierte Algorithmen nicht neu entwickelt werden müssen.

In vielen Fällen werden solche „Open Source“-Projekte von vielen Programmierern entwickelt und getestet, so dass oft davon ausgegangen werden kann, dass ein Quellcode keine Fehler enthält und auf Schnelligkeit optimiert ist.

Aus dem „Open Source“-Konzept ergeben sich aber auch Nachteile, da nie eine 100%ige Garantie vorliegt, dass ein bestehender Quellcode einwandfrei läuft. Dementsprechend müssen zur Sicherheit Tests durchgeführt werden.

Außerdem sind Projekte in den seltensten Fällen identisch, so dass trotz „Grundgerüst“ noch einige Änderungen vorgenommen werden müssen, um einen Quellcode effektiv nutzen zu können.

4. Vom „Open Source“-Programm zum Clone

4.1. Das Spiel „Starsweeper“

In der „Open Source“-Gemeinde wurden viele „MineSweeper“-Clones gefunden, doch war das Programm „Starsweeper“³ den Anforderungen am nächsten.

Diese Programmversion wurde in der Programmiersprache Borland Delphi® unter Microsoft Windows® entwickelt.

4.2. Erforderliche und relevante Anpassungen

Zwar waren die grundsätzlichen Funktionen des Original-Spiels bereits implementiert, doch mussten noch einige Änderungen vorgenommen werden, damit die Eigenschaften eines „Clones“ in den Augen der Autoren erfüllt waren:

a) Das Programm „MineSweeper“ schreibt die Dimensionen des Spielfeldes für die jeweiligen Optionen vor:

- Anfänger
 - * Breite : 9 Felder
 - * Höhe : 9 Felder
 - * Minen : 10

- Fortgeschrittener
 - * Breite : 16 Felder
 - * Höhe : 16 Felder
 - * Minen : 40

- Profi
 - * Breite : 30 Felder
 - * Höhe : 16 Felder
 - * Minen : 99

³ Der Autor des Programms ist den Studenten nicht bekannt; alle im Internet verfügbaren Informationen sind auf der CDROM zum Projekt in den Verzeichnissen „\Quellcode“ bzw. „\Binary“ zu finden.

- b) Die Anzahl der jeweils max. verteilten Minen berechnet sich in „Benutzerdefiniert“ nach der Formel

$$(H\ddot{o}he-1) * (Breite-1)$$

- c) Die minimal verteilbare Anzahl von Minen betragt 10.
- d) Die max. Werte fur die Spielfeld-Dimension (nur wahlbar in „Benutzerdefiniert“) betragen
- Breite : 30 Felder
 - Hohe : 24 Felder

Dementsprechend ergibt sich die max. Anzahl von 667 Minen. (29*23)

- e) Die Funktion „Bestzeiten“ (=Highscores) war gar nicht implementiert.
Da die zukunftigen Assistenzen bei den Zeiten berucksichtigt werden sollten, wurden die Werte entsprechend der eingesetzten Hilfe unterteilt:
- ohne Hilfe
 - US 1
 - US 2
 - US 3
 - US 1 und 2
 - US 1 und 3
 - US 2 und 3
 - US 1, 2 und 3

Gespeichert werden die Daten in „Highscores.ini“.

- f) Das Design passte nicht ganz zum Originalspiel. Aus diesem Grunde wurden die Minen, Smiley usw. durch eigene Grafiken ersetzt.
- g) Die Darstellung der Minenanzahl und die Sekundenanzeige werden in „MindSweeper“ durch LED-Grafiken angezeigt, wobei „Starsweeper“ die Zeitmessung noch gar nicht implementiert hatte.
- h) Bei einem verlorenen Spiel wurden die „?“-Felder nicht aufgedeckt, auch wenn sich darunter eine Mine befand.

- i) Die Programmeinstellungen für „Anfänger“, „Fortgeschrittener“, „Profi“ und „Benutzerdefiniert“ wurden nach Programmende gelöscht.
- j) Um die Zahlen deutlicher erscheinen zu lassen, wurde die Schriftgröße geändert.

Dies sind nur einige Änderungen, doch die Auflistung aller Änderungen würde den Rahmen des Dokuments sprengen.

4.3. Unterschiede zu „MineSweeper“

Abgesehen von den Unterstützungssystemen und den damit notwendigen Abwandlungen (z. B. „Bestzeiten“) unterscheidet sich der Clone „MindSweeper“ nur noch in wenigen Punkten vom Originalspiel „MineSweeper“:

- a) Kein s/w-Modus
Gründe:
 - Die Autoren sind der Meinung, dass dieser Modus lediglich bei s/w-Monitoren eingesetzt wird und diese fast gänzlich vom Markt verschwunden sein sollten.
 - Bei der „Risikovisualisierung“ werden explizit Farben verwendet; eine Implementierung mittels Grautönen wäre nicht nur sinnfrei, sondern auch zu aufwendig.
- b) Kein Sound
Grund:
 - Grundlage für dieses Projekt ist kein 100%iger „MineSweeper“-Clone, sondern die Umsetzung der von Dipl. Des. Philip Zerweck visuellen Änderungs-Wünsche; eine fehlende Soundunterstützung beeinträchtigt die an die Studenten gestellten Forderungen nicht.
- c) „Smiley“ (für Programmstart) ist kein Button
Grund:
 - In der bei „Starsweeper“ vorliegenden Version war kein Button implementiert und die Einbindung wäre zu aufwendig und ohne großen Nutzen für das Gesamtprojekt gewesen.

d) Keine Programmhilfe

Grund:

- Das Spiel sollte bekannt sein, wenn ein Benutzer einen Clone nutzen möchte.
Von den Studenten wurde lediglich eine separate Dokumentation bezüglich den Unterstützungssystemen verfasst.

5. Aufbau der Statistik-Datei

5.1. Allgemein

Für statistische Auswertungen war es erforderlich, Spieldaten zu speichern. Unter 5.2 wird nun erklärt, welche Daten gespeichert werden und wie diese zu bewerten sind.

5.2. Der Aufbau

Die Datei wird im ASCII-Format als *.CSV-Datei gespeichert, so dass sie ohne großen Aufwand z.B. in Microsoft EXCEL® gelesen und ausgewertet werden kann.

In der ersten Zeile der Datei befindet sich eine „Beschreibung“ der einzelnen Felder, um die Werte der folgenden Spieldaten entsprechend zuordnen zu können:

- **Modus**

Welcher Spielmodus (Anfänger, Fortgeschrittener, Profi oder Benutzerdefiniert) wurde eingestellt?

Mögliche Werte für dieses Feld: „A“, „F“, „P“ und „B“

- **US**

Beschreibt das gewählte Unterstützungssystem.

Mögliche Werte für dieses Feld:

0	(ohne Hilfe)
1	US 1 aktiviert
2	US 2 aktiviert
3	US 3 aktiviert
12	US 1 und 2 aktiviert
13	US 1 und 3 aktiviert
23	US 2 und 3 aktiviert
123	US 1, 2 und 3 aktiviert

- **Spielstatus**

Zeigt, ob das jeweilige Spiel verloren oder gewonnen wurde.
Mögliche Werte für dieses Feld:

- 0 Spiel verloren
- 1 Spiel gewonnen

- **Zugzahl**

zeigt Gesamtanzahl für aufgedeckte Felder und das Setzen/Entfernen von Fahnen und Merker (also im Prinzip die Anzahl der Maus-Klicks)

Grund hierfür ist, dass bei diesem Spiel zumindest die ersten Züge „Glückstreffer“ sind und ggf. eine Statistik verfälschen können. Somit könnten (müssen nicht unbedingt, das liegt nicht an den Autoren) bei den späteren Berechnungen Spiele mit weniger als 3 Züge aus der Statistik herausgenommen werden.

Mögliche Werte für dieses Feld: Zahl von 1 bis X

- **falsche Flaggen**

Bei Spielende wird die Anzahl der gesetzten Fahnen gezählt.
Mögliche Werte für dieses Feld: Zahl von 0 bis X

- **richtige Flaggen**

Bei Spielende wird die Anzahl der Fahnen gezählt und geprüft, ob sich auch wirklich eine Mine unter den markierten Feldern befunden hat.

Mögliche Werte für dieses Feld: Zahl von 0 bis X

Anmerkungen: Sind die Zahlen von „falsche Flaggen“ „richtige Flaggen“ identisch, heißt dies nicht, dass das Spiel auch gewonnen wurde.

- **Anzahl Minen**

Je nach Spielmodus (bei „Benutzerdefiniert“ ist es sogar variabel) wird eine unterschiedliche Anzahl von Minen versteckt.

Mögliche Werte für dieses Feld: Zahl von 10 bis 667

Anmerkungen: Sind die Zahlen von „falsche Flaggen“ bzw. „richtige Flaggen“ mit „Anzahl Minen“ identisch, bedeutet auch dies nicht zwangsweise, dass das Spiel gewonnen wurde.

- **Spielzeit**

Gesamtzeit (in Sekunden) für ein Spiel; im Gegensatz zur Anzeige im Spiel kann der Wert auch größer als 999 Sekunden sein

Mögliche Werte für dieses Feld: Zahl von 1 bis X