

Greeds - Dokumentation

Grundlagen der Programmierung 1.

Ferenc András Kiss

1 Variablen, Strukturen und Macros

1.1 main.c

1.1.1 `int` i, j

Laufvariablen für for Schleifen

1.1.2 `char` name[30]

Name des Spielers

1.1.3 `int` pos[2]

Positionskoordinaten des '@' Symbols

1.1.4 `int` vorz [8][2]

Koordinaten für die Verschiebung des '@' in jeder 8 Richtungen

1.1.5 `int` countSelected

Anzahl der Elemente von Array 'selected'

1.1.6 `int` deleted

Anzahl der gelöschten Zahlen

1.1.7 `int` direct

Anzahl der verbleibenden Richtungen

1.1.8 `time_t` tBegin, tEnd

Zeit am Anfang und am Ende des Spiels

1.1.9 `int` endOfGame

Logischer Wert um zu wissen, ob das Spiel zu Ende ist

1.1.10 `COORD` c

Struktur für die Koordinaten des Cursors in der Kommandozeile

1.2 `greed_struct.h`

1.2.1 `struct` results

Struktur für die - aus dem file 'results.txt' - eingelesenen Dateien (Name, Punkte und Spielzeit)

1.2.2 `struct` strarray

Struktur von Arrays (Koordinaten und Werte der Nachbarzahlen von '@', Koordinaten der löschraren Zahlen, Zahlen des Spielbretts)

1.2.3 `struct` strarray

Struktur von Arrays (Koordinaten und Werte der Nachbarzahlen von '@', Koordinaten der löschraren Zahlen, Zahlen des Spielbretts)

1.2.4 `#define consoleWhite`
Macro für weißen Hintergrund in der Kommandozeile

1.2.5 `#define consoleBlack`
Macro für schwarzen Hintergrund in der Kommandozeile

2 Beschreibung der verwendeten Funktionen

2.1 `greed_display.h`

2.1.1 `void cursorPosition(int x, int y, COORD c)`
Stellt die Position des Cursors auf (x;y).

2.1.2 `void printFrame(COORD c)`
Zeichnet einen großen Rahmen von '#' in die Kommandozeile.

2.1.3 `void printRank(results* person, int n, COORD c, WORD saved_attributes)`
Schreibt die besten (max. 8) Ergebnisse in die Kommandozeile heraus.

2.1.4 `int menu(COORD c, WORD saved_attributes)`
Schreibt das Spielmenü in die Konsole heraus

2.1.5 `void printLogo(COORD c)`
Zeichnet das Logo des Spiels

2.1.6 `void printSmallFrame(COORD c)`
Zeichnet einen kleinen Rahmen

2.1.7 `void print(int n)`
Schreibt die farbigen Zahlen des Spielbretts heraus

2.1.8 `void printTime(time_t tEnd)`
Schreibt die Spielzeit (formatiert) heraus

2.1.9 `void printResult(time_t tEnd, int deleted, COORD c)`
Schreibt das Ergebnis am Ende des Spiels heraus

2.2 `greed_file.h`

2.2.1 `void addResult(char name[30], int deleted, time_t time)`
Fügt das Ergebnis (Name, Punkte, Spielzeit) zu 'results.txt' hinzu.

2.2.2 `void result(COORD c, WORD saved_attributes)`
Liest die Ergebnisse von 'results.txt' ein, ordnet die (max. 8) besten in abnehmende Reihenfolge.

2.2.3 `void saveGame(strarray* arr, int pos[2])`
Speichert den aktuellen Spielzustand, also die Zahlen des Spielbretts und die Koordinaten des Cursors in 'last.txt'.

2.2.4 `void openArray(strarray* arr, int pos[2])`
Öffnet das letzte gespeicherte Spielbrett ('last.txt').

2.3 `greed_game.h`

2.3.1 `void fillArray(strarray* arr)`
Füllt den Array 'row' mit Randomzahlen zwischen 1 und 9 auf.

2.3.2 `int` testSelected (strarray* arr, `int` y, `int` x, `int` element)

Testet jedes Element von 'row', und gibt 1 zurück, wenn die entsprechende Zahl (bzw. Koordinaten) gelöscht werden können. Diese Zahlen erscheinen mit weißem Hintergrund.

2.3.3 `int` testAround(strarray* arr, `int` y, `int` x)

Gibt 1 zurück, falls '@' einen Nachbarn mit den Koordinaten (x;y) hat. Dadurch wird die Anzahl der möglichen Schritten berechnet.

2.3.4 `int` isNull(strarray* arr, `int` n, `int`* vorz)

Überprüft, ob es in einer der 8 Richtungen die nötige Anzahl von Zahlen zusammenhängend existiert. Der Rückgabewert ist 1, falls ein Schritt nicht möglich ist, weil es unter den zu löschenden Zahlen eine 0 gibt.

2.3.5 `void` nextStep(strarray* arr, `int` pos[2], `int` n, `int`* deleted, `int`* vorz)

Stellt den Wert der besuchten Zahlen auf 0. Auf dem Brett erscheint keine 0, so werden die Zahlen gelöscht. Dann ändert sich die Position von '@'.

2.3.6 `void` endPosition(strarray* arr, `int`* k, `int`* vorz)

Entscheidet für jede Richtung, ob der nächste Schritt innerhalb des Spielbretts wäre.

2.3.7 `void` fillSelected (strarray* arr, `int`* k, `int` n, `int`* vorz)

Falls ein Schritt gültig ist, werden die Koordinaten der löschbaren Zahlen im Array 'selected' gespeichert.

2.3.8 `void` keyPressed(strarray* arr, `int` pos[2], `int`* deleted, `int`* vorz, `int`* endOfGame)

Wenn eine Taste während des Spiels gedrückt wird, wird entweder ein Schritt gemacht (Zifferntasten), der aktuelle Spielzustand gespeichert ('s'), oder mit dem Spiel aufgehört. Falls man 's' oder 'q' drückt, wird 'endOfGame' auf 1 gestellt.