

Highscore++

Highscore++ ist ein eigenes Highscore-System, basierend auf den `ds_grid` Funktionen. Es dient zum Ausgleich der Defizite des internen Highscore-Systems bzw. als kompletter Ersatz.

Funktionsweise

Beim Erstellen oder Laden einer Highscore wird ein Handle zurückgegeben, das für alle anderen Funktionen gebraucht wird, um auf die Highscore zugreifen zu können.

Um mehrere Daten gleichzeitig zu übergeben, z.B. beim Hinzufügen eines neuen Eintrags, werden alle Daten mit `|` getrennt in einen String geschrieben.

Beispiel:

```
hs = hs_create_highscore("Name|Score");  
hs_add(hs, "Spieler|45");
```

Außerdem fangen die Highscoreeinträge nicht beim index 0 an, sondern bei 1. Das liegt daran, dass die erste Reihe für die Spaltennamen reserviert ist.

Funktionsreferenz

```
hs_create_highscore(cols)
```

Erstellt eine neue Highscore und gibt dessen Handle zurück.

`cols` gibt die Spalten (getrennt mit `|`) an. Auch leere Spalten sind möglich.

```
hs_destroy(handle)
```

Löscht die Highscore `handle` und gibt den Speicherplatz wieder frei.

Alternativ kann auch einfach `ds_grid_destroy()` aufgerufen werden.

```
hs_add(handle, data)
```

Fügt den Datensatz `data` am Ende der Highscore `handle` hinzu.

`data` muss ein String sein, der die Daten in der richtigen Reihenfolge und mit `|` getrennt enthält.

`hs_insert(handle, pos, data)`

Fügt den Datensatz `data` an Position `pos` in der Highscore `handle` ein.

`hs_set(handle, pos, data)`

Überschreibt den Wert an Position `pos` in der Highscore `handle` mit dem Datensatz `data`.

`hs_get(handle, pos)`

Gibt einen String mit den Werten an Position `pos` in der Highscore `handle` zurück. Dieser String kann gegeben Falls mit `hs_dataset_split()` in eine `ds_list` umgewandelt werden.

`hs_get_value(handle, col, pos)`

Gibt den Wert an Position `pos` in der Spalte `col` in der Highscore `handle` zurück. Alternativ kann auch `ds_grid_get()` verwendet werden.

`hs_get_colwidth(handle, col)`

Gibt die Breite in Pixel der Spalte `col` in der Highscore `handle` zurück.

`hs_get_length(handle)`

Gibt die Anzahl der Einträge der Highscore `handle` zurück.

`hs_find_column(handle, col)`

Gibt die Position des ersten Vorkommens der Spalte `col` in der Highscore `handle` zurück. Falls `col` nicht in der Highscore enthalten ist, wird `-1` zurückgegeben.

`hs_find_entry(handle, col, val)`

Gibt die Position des ersten Vorkommens des Wertes `val` in der Spalte `col` in der Highscore `handle` zurück. Wenn `val` nicht in der Highscore enthalten ist, wird `-1` zurückgegeben.

`hs_remove(handle)`

Löscht den letzten Eintrag in der Highscore `handle`.

`hs_remove_at(handle, pos)`

Löscht den Eintrag an Position `pos` in der Highscore `handle`.

`hs_clear(handle)`

Leert die Highscore `handle`.

`hs_sort(handle, col, desc)`

Sortiert die Highscore `handle` nach den Werten in der Spalte `col`.
`desc` gibt an, ob die Werte absteigend (`true`) oder aufsteigend (`false`) geordnet werden sollen.

`hs_draw(handle, x, y, len, num, distH, distV)`

Zeichnet die Highscore `handle` an Position (`x` | `y`) mit dem horizontalen Zellenabstand `distH` und dem vertikalen Zellenabstand `distV`.
`len` gibt die Anzahl an Einträgen an. Wenn `len -1` ist, werden alle Einträge angezeigt.
`num` gibt an, ob die Zeilen durchnummeriert werden sollen (`true`).

`hs_save(handle, fname, shift)`

Speichert die Highscore `handle` in der Datei `fname`.
Um die Highscore leicht zu verschlüsseln, wird eine simple Zeichenverschiebung verwendet. `shift` gibt dabei die Verschiebung an (0 für keine Verschiebung).

`hs_load(fname, shift)`

Lädt eine Highscore aus der Datei `fname` und gibt dessen Handle zurück.
`shift` gibt die Zeichenverschiebung an, die beim Speichern verwendet wurde.

`hs_dataset_split(data)`

Spaltet den String `data` auf und schreibt alle darin enthaltenen Werte in eine `ds_list`, deren Handle anschließend zurückgegeben wird.
Diese Liste sollte nach der Benutzung wieder mit `ds_list_destroy()` freigegeben werden.

Neben diesen Funktionen kann natürlich auch direkt mit den `ds_grid` Funktionen auf die Highscores zugegriffen werden.

Beispiel

```
var hs, date;

//Highscore erstellen
hs = hs_create_highscore („Name|Score“);

//Einträge hinzufügen
hs_add(hs, „Spieler1|45“);
hs_add(hs, „Spieler2|65“);
hs_add(hs, „Spieler3|26“);

//Einträge absteigend sortieren
hs_sort(hs, „Score“, true);

//Prüfen ob der Eintrag mit der höchsten Score ganz oben ist
if (hs_get_value(hs, „Name“, 1) == „Spieler2“)
{
    show_message(„Richtig sortiert!“);
}

//Highscore löschen
hs_destroy(hs);
```