

GML-Skripte zur  
Textformatierung für  
Umbruch, Schatten und  
Kontur.

## Skripte für Textformatierung

### Description

Für Aufgaben, die immer wieder anfallen, schreibt man in GML Skripte. Das ist äußerst praktisch und spart viel Arbeit. Mittlerweile kann man sogar mehrere Funktionen in ein Skript stecken, was die Übersicht erhöht. Da Textformatierungen in fast allen Spielen vorkommen, zeige ich hier meine drei am häufigsten verwendeten Funktionen, die ich immer wieder (aktuell GMS 2.3) anpasse.

### Texte formatieren

Texte kommen in fast allen Spielen vor. Manchmal nur für Menüs, oft aber für Spielebeschreibungen, Statistiken, Dialoge etc. Dabei kann es sich bei Texten auch um Zahlen handeln. Im Grunde geht es um alles, was wir normalerweise mit `draw_text()` an den Spieler weitergeben.

Diese Texte wollen schön formatiert sein. GML bietet uns bereits einige Möglichkeiten. Wir können Schriften definieren, Farben, horizontale und vertikale Ausrichtung sowie Transparenz. Aber drei wichtige Formatierungen müssen wir selbst vornehmen:

- Text mit Schatten
- Text mit Umrandung
- Text, der an einer bestimmten Pixel-Breite umgebrochen wird

Genau darum kümmern wir uns in diesem Tutorial. Die Skripte kannst Du natürlich individuell anpassen.

### Ursprung der Skripte

Die nachfolgenden Skripte sind nicht von mir. Schatten und Umrandung verwende ich schon ewig und passe sie an die jeweilige GMS-Version an, sofern sich etwas in GML ändert. Wo der Ursprung dieser Skripte liegt, kann ich leider nicht mehr sagen.

Das Skript `draw_text_limit_width()` [kommt ursprünglich von gmlscripts.com](https://www.gmlscripts.com/). Eine tolle Seite, aber die

Skripte sind, soweit ich gesehen habe, schon etwas veraltet.

## Anpassungen an GMS 2.3

Wer die Version 2.3 von GameMaker Studio verwendet, kann die Skripte sofort verwenden. Im Vergleich zu älteren Versionen habe ich die Argumente geändert und eine ausführliche Beschreibung hinzugefügt. Dadurch taucht automatisch in der unteren Hilfe-Zeile das passende Argument auf. Es fühlt sich also an, als würde man eine normale GML-Funktion aufrufen.

Außerdem ist der ganze Code ausführlich kommentiert, weswegen ich auf weitere Erklärungen verzichte. Sollte es dennoch Fragen geben, könnt ihr diese gerne als Kommentar hinterlassen.

Im Titelbild sieht man übrigens alle drei Skripte im Einsatz.

### draw\_text\_shadow()

```

/// @function draw_text_shadow()
/// @param {xx} x-Koordinate
/// @param {yy} y-Koordinate
/// @param {scol} Schattenfarbe
/// @param {tcol} Textfarbe
/// @param {b} Schatten-Abstand
/// @param {alpha} Alpha-Schatten
/// @param {text} Text
/// @param {r} Richtung (1 = links, 0 = rechts)
/// @description Zeichnet einen Text mit Schatten
// Beispiel: draw_text_shadow(x, y, #000000, #FFFFFF, 5, 0.4, „Bytegame.de“, 0)
function draw_text_shadow(xx, yy, scol, tcol, b, alpha, text, r)
{
    var dgc = draw_get_color(); // Ermittelt die aktuelle Zeichenfarbe

    draw_set_color(scol); // Setzen die Farbe des Schattens
    draw_set_alpha(alpha); // Setzen Alpha für Schatten

    if (r = 0){draw_text(xx + b, yy + b, text);} // Zeichnet den Schatten nach re
    if (r = 1){draw_text(xx - b, yy + b, text);} // Zeichnet den Schatten nach li

    draw_set_alpha(1); // Alpha auf 1
    draw_set_color(tcol); // Textfarbe
    draw_text(xx, yy, text); // Text zeichnen

    draw_set_color(dgc); // Farbe zurücksetzen
}

```

### draw\_text\_outline()

```

/// @function draw_text_outline
/// @arg {xx} x-Koordinate
/// @arg {yy} y-Koordinate
/// @arg {text} Text

```

```

/// @arg {tcolor} Textfarbe
/// @arg {ocolor} Umrandungsfarbe
/// @arg {size} Größe
/// @description Zeichnet einen Text mit einer Umrandung;
// Beispiel: draw_text_outline(x, y, „Bytegame.de“, #FFFFFF, #000000, 2);
function draw_text_outline(xx, yy, text, tcolor, ocolor, size)
{
    var dgc = draw_get_color(); // Ermittelt die aktuelle Zeichenfarbe

    draw_set_color(ocolor); // Farbe der Umrandung
    draw_text(xx - size, yy, text); // Umrandung links
    draw_text(xx + size, yy, text); // Umrandung rechts
    draw_text(xx, yy - size, text); // Umrandung oben
    draw_text(xx, yy + size, text); // Umrandung unten

    draw_set_color(tcolor); // Textfarbe
    draw_text(xx, yy, text); // Text zeichnen

    draw_set_color(dgc); // Farbe zurücksetzen
}

```

## draw\_text\_limit\_width()

```

/// @function draw_text_limit_width
/// @arg {text} Text
/// @arg {width} Breite
/// @arg {split} Wort_teilen?
/// @description Bricht längere Texte nach vorgegebener Pixel-Breite um;
// Beispiel: draw_text_limit_width("Bytegame.de bietet viele interessante Arti
function draw_text_limit_width(text, width, split)
{
    var pos_space, pos_current, text_output, bbreak;
    pos_space = -1;
    pos_current = 1;
    bbreak = „\n“;

    text_output = „“;

    while (string_length(text) >= pos_current)
    {
        if (string_width(string_copy(text, 1, pos_current)) > width)
        {
            // Wenn in dieser Zeile ein Leerzeichen steht, können wir dort abb
            if (pos_space != -1)
            {
                text_output += string_copy(text, 1, pos_space) + string(bbreak

                // Den Text, den wir uns gerade angesehen haben, aus der aktue
                text = string_copy(text, pos_space+1, string_length(text)-(pos

            pos_current = 1;
            pos_space = -1;
        } else if (split) {
            // Wenn nicht, können wir Zeilenumbrüche erzwingen

```

```
        text_output += string_copy(text, 1, pos_current-1) + string(bb

        // Den Text, den wir uns gerade angesehen haben, aus der aktue
        text = string_copy(text, pos_current, string_length(text)-(pos_
        pos_current = 1;
        pos_space = -1;
    }
}

pos_current += 1;
if (string_char_at(text, pos_current) == " ") {pos_space = pos_current;
}

if (string_length(text) > 0) {text_output += text;}

return text_output;
}
```

Dieses Skript sollte man nicht im **Draw-Event** verwenden! Am besten im **Create-Event**, etwa so:

```
draw_set_font(f_test);
txt = draw_text_limit_width("GML-Skripte zur Textformatierung für Umruch, Scha
```

Die Schrift müssen wir schon hier definieren, damit das Skript die Breite berechnen kann.  
Anschließend kann die Variable *txt* im **Draw-Event** verwendet werden. Bspw so:

```
draw_set_font(f_test);
draw_set_halign(fa_center);
draw_set_valign(fa_middle);

var h_center = room_width/2;
var v_center = room_height/2;

draw_text_shadow(h_center, v_center, #000000, #FFFFFF, 5, 0.4, txt, 0);
draw_text_outline(h_center, v_center, txt, #FFFFFF, #000000, 2);
```

### **Date Created**

15. April 2022

### **Author**

sven