



## Textscroller: Wellen und einzelne Farben

### Description

Texte sind in vielen Spielen sehr wichtig, weshalb wir uns auch immer wieder in den Tutorials damit befassen. Heute geht es um einen speziellen Textscroller. Wir hatten bereits solche Tutorials, [etwa hier](#). Nun werden wir eine schönere Wellenbewegung realisieren und jedem Zeichen eine eigene Farbe geben.

### Farbe pro Zeichen

Da die Theorie bereits im oben verlinkten Tutorial erklärt wurde, beschränke ich mich auf die wesentlichen Änderungen.

Zunächst definieren wir im Create-Event, neben anderen Variablen, einen [Array](#), in dem wir 16 Farben mit ihren RGB-Werten vorgeben. Das lokale Array heißt *txt\_colors* und entspricht letztlich nur einer Liste. Anschließend erstellen wir das Array *char\_colors* und starten eine Schleife. In dieser Schleife gehen wir jedes Zeichen vom Text durch und weisen ihm eine Farbe zu, indem wir es in das neue Array schreiben. Als Hilfsvariable haben wir noch *count*. Hier zählen wir die Farben des Arrays *txt\_colors* hoch und setzen es auf 0 zurück, sobald wir die letzte Farbe erreicht haben.



Der Textscroller

Hier der entsprechende Code:

```
scroll_pos = 0;
```

```
text_size = 12;
scroll_speed = 1;
wave_amplitude = 18; // Amplitude der Welle
wave_frequency = 0.04; // Frequenz der Welle

text = „ Hallo GameMaker Freunde. Dies ist ein weiteres Tutorial fuer einen so

var txt_colors = [
  [0, 255, 0], // Grün
  [0, 255, 127], // Gelbgrün
  [127, 255, 0], // Gelb
  [255, 255, 0], // Gelbgelb
  [255, 127, 0], // Orange
  [255, 0, 0], // Rot
  [255, 0, 127], // Pink
  [255, 0, 255], // Magenta
  [127, 0, 255], // Lila
  [0, 0, 255], // Blau
  [0, 127, 255], // Himmelblau
  [0, 255, 255], // Türkis
  [0, 255, 127], // Grüngrün
  [127, 255, 0], // Grüngelb
  [255, 255, 0], // Gelbgelb
  [255, 127, 0] // Orange
];

char_colors = [];

var count = 0;
for (var i = 0; i < string_length(text); i++)
{
  char_colors[i] = txt_colors[count];
  count++;
  if (count > array_length(txt_colors)-1) {count = 0;}
}
```

## Wir zeichnen

Nachdem wir alles im Create-Event vorbereitet haben, geht es an das Draw-Event. Wie üblich kümmern wir uns zunächst um die Schrift (im Beispiel *f\_pixel\_12*) und Alpha.

Dann geht es in eine [for-Schleife](#), die so oft durchlaufen wird, wie der Text Zeichen hat. Bedenke, dass das 60mal pro Sekunde erfolgt (bei den GMS-Standardereinstellungen). D. h. bei einem Text von 500 Zeichen wird die Schleife 30.000 Mal durchlaufen.

In der Schleife suchen wir uns das aktuelle Zeichen raus (*var char...*) und bestimmen die y-Position. Mit **draw\_set\_color** setzen wir die Farbe. Diese wird aus dem Array *char\_colors* ausgelesen, was wir mit der GM-Funktion **make\_colour\_rgb** tun. Anschließend zeichnen wir den Text bzw. das jeweilige Zeichen.

Nach der Schleife wird *scroll\_pos* um den Wert *scroll\_speed* versetzt.

Am Ende schauen wir noch nach, ob der Text den Bildschirm verlassen hat. Wenn ja, setzen wir `scroll_pos` auf 0 und er beginnt wieder von vorne.

Und hier der Code:

```
draw_set_halign(fa_left);
draw_set_valign(fa_top);
draw_set_color(c_white);
draw_set_font(f_pixel_12);
draw_set_alpha(1);

var y_pos = room_height-40; // Position des Scrollers auf der Y-Achse

for (var i = 0; i < string_length(text); i++)
{
    var char = string_char_at(text, i);
    y_pos = room_height-40 + wave_amplitude * sin((scroll_pos + i * text_size)

    draw_set_color(make_colour_rgb(char_colors[i][0], char_colors[i][1], char_
    draw_text(room_width + i * text_size - scroll_pos, y_pos, char);
}

scroll_pos += scroll_speed;

if (scroll_pos > string_length(text) * text_size)
{
    scroll_pos = 0;
}
```

Das war es auch schon.

**Date Created**

18. Februar 2023

**Author**

sven