



## Horizontaler Sternenscroller und Parallax scrolling

### Description

Sternenscroller sind ein alter, aber heute noch sehr nützlicher Effekt. Es gibt sie in vielen verschiedenen Varianten. Die horizontale Version ist nicht nur für Menüs, Credits oder Intros gut geeignet, sondern auch für Weltraumshooter. Dieses Tutorial zeigt, wie man das leicht realisiert.

### Das Prinzip

Bei den Sternen ist es zunächst egal, ob sie als Kreise, Quadrate oder Sprites gezeichnet werden. Im Beispiel verwende ich Kreise und, bei den kleinsten, Punkte. Meine Version scrollt von links nach rechts, ihr könnt das aber gerne verdrehen, von oben nach unten oder umgekehrt scrollen.

Das Prinzip ist dabei denkbar einfach und wer sich mit [Arrays](#) und [for-Schleifen](#) auskennt, wird keine Probleme damit haben, es zu verstehen. Wir brauchen lediglich ein Objekt. Darin generieren wir Arrays, in denen wir die x- und y-Position eines jeden Sterns definieren.

Plural? Ja, im Beispiel wird es vier Ebenen von Sternen geben. Jede Ebene, von hinten nach vorne, hat größere Sterne, die schneller scrollen. Dadurch entsteht ein toller räumlicher Effekt. Dieses Prinzip nennt man auch „Parallax scrolling“. Dabei brauchen wir in GML im Raum keine verschiedenen Ebenen, wir zeichnen alles auf derselben Fläche.

Im **Draw-Event** haben wir für jede Ebene eine eigene Schleife, mit deren Hilfe wir die Sterne neu positionieren. Jede dieser Ebenen hat nicht nur seine eigene Größe und Geschwindigkeit, sondern auch Transparenz. Das bedeutet letztlich, dass die vorderen Sterne größer, schneller und heller sind, als die hinteren. Die Farbe könnt ihr natürlich selbst wählen und sogar für jede Ebene anpassen.

### Create-Event

```
randomize(); // Zufallsgenerator initiieren  
  
// Liste mit 40 Sternen  
stars_40 = [];
```

```
for (var i = 0; i < 40; i++)
{
    array_push(stars_40, {
        x: random(room_width),
        y: random_range(16, room_height-32)
    });
}

// Liste mit 60 Sternen
stars_60 = [];
for (var i = 0; i < 60; i++)
{
    array_push(stars_60, {
        x: random(room_width),
        y: random_range(16, room_height-32)
    });
}

// Liste mit 100 Sternen
stars_100 = [];
for (var i = 0; i < 100; i++)
{
    array_push(stars_100, {
        x: random(room_width),
        y: random_range(16, room_height-32)
    });
}

// Liste mit 200 Sternen
stars_200 = [];
for (var i=0; i<200; i++)
{
    array_push(stars_200, {
        x: random(room_width),
        y: random_range(16, room_height-32)
    });
}

// Geschwindigkeit der Sternengruppen
speed_40 = 1.2;
speed_60 = 0.8;
speed_100 = 0.4;
speed_200 = 0.2;

// Größe der Sternengruppen
size_40 = 0.7;
size_60 = 0.63;
size_100 = 0.57;
```

Dem oben beschriebenen Prinzip folgend ist es ziemlich einfach. Wir füllen 4 Arrays mit den Koordinaten. Dabei haben wir 40, 60, 100 und 200 Sterne, je nach Ebene. Am Ende definieren wir für jede Ebene eine Geschwindigkeit und eine Größe.

Warum wir nur drei Größen definieren? Nun, bei den 200 Sternen verwenden wir **draw\_point()** und ein

Punkt hat keine Größe, die wir definieren könnten. Ihr könnt das natürlich beliebig anpassen.

Wie man sieht, befinden sich alle Sterne zwischen 16 und *room\_height*-32. Das lässt oben und unten ein wenig Platz. Auch das lässt sich natürlich beliebig anpassen.

## Draw-Event

```
draw_set_color(c_white);

draw_set_alpha(1);
// Durchlaufe die Liste der 200 Sternen und zeichne sie
for (var i=0; i<array_length(stars_200)-1; i++)
{
    stars_200[i].x = stars_200[i].x+speed_200;
    draw_point(stars_200[i].x, stars_200[i].y)

    if (stars_200[i].x > room_width)
    {
        stars_200[i].x = 0;
    }
}

draw_set_alpha(.25);
// Durchlaufe die Liste der 60 Sternen und zeichne sie
for (var i=0; i<array_length(stars_100)-1; i++)
{
    stars_100[i].x = stars_100[i].x+speed_100;
    draw_circle(stars_100[i].x, stars_100[i].y, size_100, false);

    if (stars_100[i].x > room_width+size_100)
    {
        stars_100[i].x = -size_100;
    }
}

draw_set_alpha(.5);
// Durchlaufe die Liste der 60 Sternen und zeichne sie
for (var i=0; i<array_length(stars_60)-1; i++)
{
    stars_60[i].x = stars_60[i].x+speed_60;
    draw_circle(stars_60[i].x, stars_60[i].y, size_60, false);

    if (stars_60[i].x > room_width+size_60)
    {
        stars_60[i].x = -size_60;
    }
}

draw_set_alpha(.75);
// Durchlaufe die Liste der 40 Sternen und zeichne sie
for (var i=0; i<array_length(stars_40)-1; i++)
{
    stars_40[i].x = stars_40[i].x+speed_40;
```

```
draw_circle(stars_40[i].x, stars_40[i].y, size_40, false);

if (stars_40[i].x > room_width+size_40)
{
    stars_40[i].x = -size_40;
}
}
```

Nun werden die Sterne gezeichnet, was ebenfalls viermal identisch passiert. Die Schleifen werden so oft durchlaufen, wie es Sterne im Array gibt. Bei jedem Stern addieren wir die Geschwindigkeit zur x-Koordinate hinzu. Dann werden die Punkte bzw. Kreise gezeichnet. Zum Ende hin prüfen wir noch, ob sich der Stern außerhalb des Raums befindet. Wenn ja, setzen wir ihn wieder zurück.

Das war es auch schon!

**Date Created**

3. März 2023

**Author**

sven