Umgang mit Texten

Description

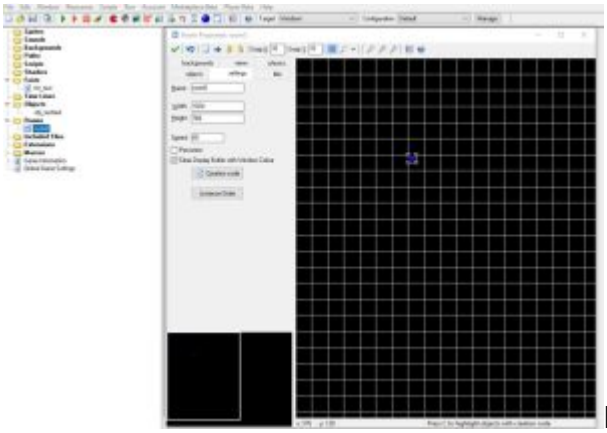
Früher oder später kommt man nicht drum herum, Texte in einem Spiel anzuzeigen. Seien es Beschriftungen von Buttons, ein Vorspann, Dialoge oder Beschriftungen im Userinterface. Heute schauen wir uns an, welche Möglichkeiten der GameMaker bietet.

Abgesehen von einem Raum und einem Objekt, brauchen wir drei Dinge, um einen Text korrekt anzeigen zu können. Eine Schrift, eine Farbe und den Befehl zur Textausgabe.

Die Schrift

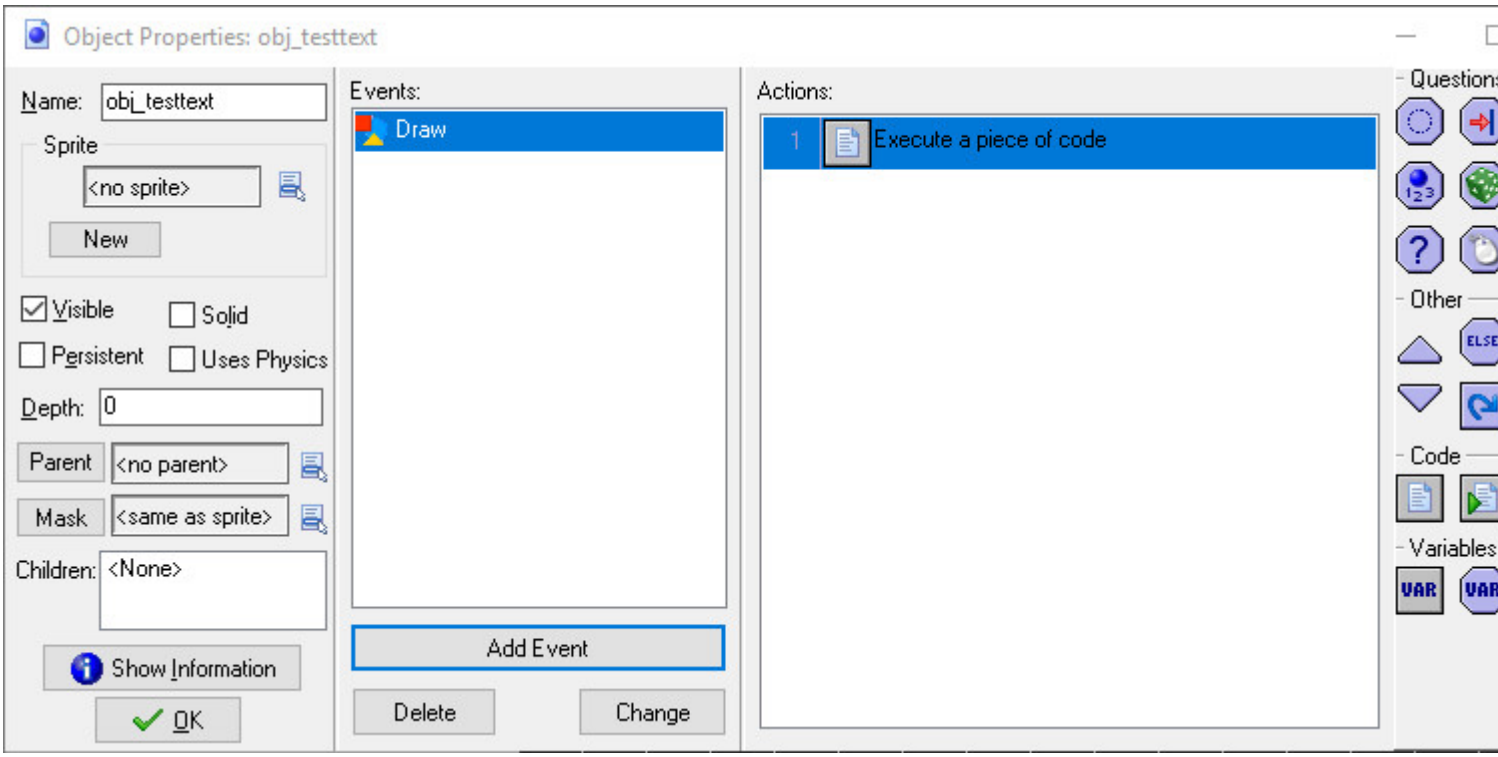
Eine Schrift erstellt man bei den **Fonts**. GameMaker kann nicht einfach irgend eine Schrift aus dem Windows-Ordner des Spielers verwenden, die Schrift muss im GameMaker-Projekt eingebettet werden. Schließlich wissen wir nicht, ob jeder Spieler die gleichen Schriften besitzt wie wir und letztlich soll ja das Spiel bei allen Spielern gleich aussehen.

Mit einem Klick auf das Icon **Create a Font** kann man eine Schrift als Ressoruce erstellen. Im Fenster wählt man eine Schriftart, eine Größe und zusätzliche Optionen wie Fett, Kursiv (wenn das die Schrift anbietet) Anti-Aliasing, also die Kantenglättung. Wenn man eine möglichst glatte Schrift haben möchte, ist die voreingestellte 3 eine gute Wahl. Wenn man ein Retro-Spiel mit entsprechender Schrift macht, sollte man es ausschalten. Wichtig ist der sogenannte Range. Wenn man alle deutschen Zeichen, also auch Umlaute und ein scharfes S haben möchte, muss man, wenn man auf das Pluszeichen klickt, ASCII einstellen. Damit werden die Zeichen 32 bis 255 eingeladen und man hat damit alles, was manin der deutschen Sprache braucht. Die Range-Option dient dazu, möglichst optimiert zu arbeiten.Wenn man eine Schrift nur benötigt, um Zahlen anzuzeigen, müssen die Buchstaben nicht mitgeliefertund im Spiel geladen werden.


Erst einmal erstellen wir den oben genannten Raum, ein

Testobjekt und setzen das Objekt irgendwo in den Raum. Da wir nachher ein wenig mit den Koordinaten experimentieren werden, ist es völlig egal, wo im Raum sich das Objekt befindet.

Um einen Text anzeigen zu können, müssen die Befehle in einem der **Draw-Events** liegen. Für den ersten Test reicht der normale Draw, die anderen sollen uns an dieser Stelle nicht weiter beschäftigen.



Nachdem wir das Event erstellt haben, müssen wir nun die drei oben genannten Dinge tun. Die Schrift aktivieren, die Farbe definieren und dann unseren Text angeben.

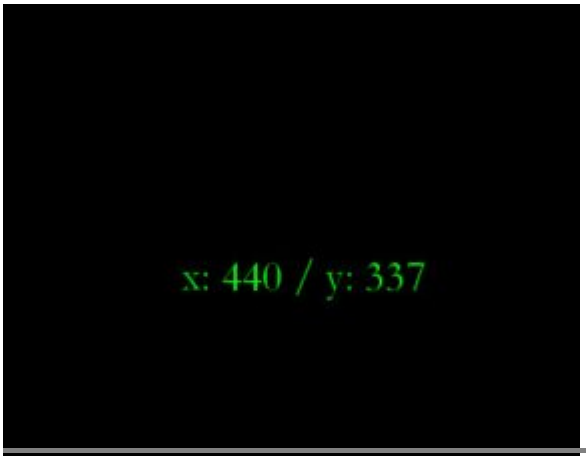
Event Draw

```
draw_set_font(fnt_test);
draw_set_color(c_lime);
draw_text(20, 20, 'Hallo du schöne Welt!');
```

In der ersten Zeile wird die Schrift definiert, die ich *fnt_test* genannt habe. Eine Zeile weiter definieren wir eine Farbe. Diese Farbe würde sich nicht nur auf den Text auswirken, sondern auf alles, was darunter gezeichnet wird.

In der letzten Zeile schreiben wir den Text auf den Bildschirm. Die ersten beiden Zahlen, also 20 und 20, geben die Koordinaten *x* und *y* auf dem Bildschirm an. Am Ende kommt, in Anführungszeichen, ein Text. Damit wir auch einen Umlaut sehen, habe ich das berühmte „Hallo Welt!“ ein wenig abgeändert.


GameMaker: Studio
Hallo du schöne Welt!


x: 440 / y: 337

kurz die Koordinaten an. In anderen Tutorials wurde schon mehrfach darauf hingewiesen, dass es hier mehrere Möglichkeiten gibt. Mit festen Zahlen sprechen wir immer einen genauen Punkt auf dem Bildschirm an. Wenn wir stattdessen nur *draw_text(x, y, 'Hallo du schöne Welt!');* angeben, landet der Text dort, wo das Objekt auf dem Bildschirm platziert wurde. So kann man beispielsweise den Namen des Spielers mit der Spielfigur mitbewegen. *draw_text(x+20, y+20, 'Hallo du schöne Welt!');* bedeutet, dass der Text 20 Pixel rechts und 20 Pixel unterhalb des Mittelpunkts des Objekts befindet.

Mit Befehlen wie **room_width** und **room_height** kann man sich auf die Raumgröße beziehen, mit **view_xview** und **view_yview** bezieht man sich auf eine Position relativ zum View.

Wir versuchen nun folgenden Code:

```
draw_set_font(fnt_test);
draw_set_color(c_lime);
draw_text(mouse_x + 20, mouse_y + 20, 'Hallo du schöne Welt!');
```

Jetzt bewegt sich der Text mit der Maus mit. Man kann also nicht nur die Koordinaten des eigenen Objektes, des Raumes oder des Views verwenden, sondern auch die Maus und andere Objekte.

Zeilenumbruch

Nun wissen wir, wie wir einen Text optimal auf dem Bildschirm platzieren können. Wenn wir einen Text in mehreren Zeilen angezeigt haben möchten, gibt es in GameMaker mehrere Möglichkeiten. Im Text dient das Raute-Zeichen # als Zeilenumbruch. Stattdessen können wir auch mitten im Befehl einen Zeilenumbruch schreiben, der von GameMaker ebenso als solcher erkannt wird. Zu guter Letzt können wir auch jede Zeile einzeln ausgeben. Das kann sinnvoll sein, wenn wir den Zeilenabstand manuell beeinflussen wollen. Hier die drei Beispiele:

```
draw_set_font(fnt_test);
draw_set_color(c_lime);
draw_text(mouse_x + 20, mouse_y + 20, 'Zeile 1 #Zeile 2 #Zeile 3');

draw_set_font(fnt_test);
draw_set_color(c_lime);
draw_text(mouse_x + 20, mouse_y + 20, 'Zeile 1
Zeile 2
Zeile 3');

draw_set_font(fnt_test);
draw_set_color(c_lime);
draw_text(mouse_x + 20, mouse_y + 20, 'Zeile 1');
draw_text(mouse_x + 20, mouse_y + 50, 'Zeile 2');
draw_text(mouse_x + 20, mouse_y + 80, 'Zeile 3');
```

Die Beispiele 1 und 2 sehen auf dem Bildschirm identisch aus. In Beispiel 3 werden, je nach Schriftart und Schriftgröße, die Abstände größer oder kleiner sein als bei den ersten beiden Beispielen.

Die dritte Möglichkeit sollte nur in Ausnahmefällen verwendet werden, da sie deutlich aufwändiger ist. Methode 1 hat den Nachteil, dass sie bei längeren Texten unübersichtlich wird. Außerdem bemerken wir schnell, dass wir nicht ohne weiteres eine Raute zeichnen können. Die gelingt, indem wir einen

Slash davor stellen, also \# schreiben. Dann wird die Raute nicht als Zeilenumbruch, sondern als ASCII-Zeichen interpretiert.

Zahlen als Text

In der Programmierung arbeitet man viel mit Zahlen. Egal ob Koordinaten, Energiewerte oder Punkte, im Hintergrund werden vom Computer viele Zahlen verarbeitet und manchmal wollen wir sie auf dem Bildschirm anzeigen. Das ist mit einem kleinen Hindernis verbunden. Wenn wir einfach eine Variable statt eines Textes angeben, belohnt uns der GameMaker mit einer Fehlermeldung. Deswegen müssen wir eine Zahl erst in einen String, also eine Zeichenkette umwandeln. Wir bleiben bei dem Beispiel mit dem Mauszeiger und wollen uns nun die Koordinaten des Zeigers anzeigen lassen.

```
draw_set_font(fnt_test);
draw_set_color(c_lime);
draw_text(mouse_x + 20, mouse_y + 20, string(mouse_x));
```

Wenn wir das Programm starten, wird uns die x-Koordinate der Maus angezeigt. Sie verändert sich mit der Bewegung der Maus. Mit der gleichen Methode können wir auch die y-Koordinate anzeigen lassen. *string()* wandelt jede Zahlenvariable in einen String um.

Variablen und Texte mischen

Wenn wir das bisher gelernte anwenden, ist es mühsam, eine Anzeige wie „x: 100 / y: 120“ für die Mausanzeige zu generieren, weil wir zwei Variablen und zwei Textpassagen haben. So eine Anzeige würde vier Zeilen voraussetzen und wir müssten die mögliche Zeichenlänge der Koordinate berücksichtigen, weil sie in den meisten Fällen 1 bis 4 Stellen haben kann. Zum Glück lässt sich das in GameMaker kombinieren:

```
draw_set_font(fnt_test);
draw_set_color(c_lime);
draw_text(mouse_x + 20, mouse_y + 20, 'x: ' + string(mouse_x) + ' / y: ' + str
```

Mit dem Pluszeichen können wir mehrere Variablen und Texte miteinander mischen. Das Großartige daran ist, dass die Abstände nun immer passen, ganz abgesehen davon, dass wir das Problem innerhalb einer Teile elegant lösen konnten.

Texte auslagern

Je umfangreicher ein Projekt wird, umso sinnvoller ist es, Texte zentral auszulagern, um bei Änderungen nicht laufend zahllose Objekte und Events editieren zu müssen. Hier bietet es sich an, Texte in ein Script auszulagern und diese als globale Variablen zu definieren. Das ist zwar nicht sehr Ressourcenschonend, da sich alle Texte immer im Speicher befinden, aber es ist sehr Wartungsfreundlich. Außerdem ist es ab hier nur noch ein kleiner Schritt, um mehrsprachige Texte zu erzeugen.

Um das Prinzip zu zeigen, erzeugen wir zwei Variablen im Draw-Event, die wir bei der Textausgabe verwenden.

```
Weitere Möglichkeiten
Mit dem bisher gezeigten lassen sich die gängigsten Aufgaben im Alltag der Spi


Allerdings kann man mit Texten noch viele andere Dinge tun. Man könnte zählen,
erfährst Du in diesem Tutorial.
```

Date Created
16. November 2016
Author
sven