

Tutorial: Dein erstes Spiel

Copyright 2006, www.gm-d.de (engl. Original Copyright 2003-2004, Mark Overmars)

Übersetzer: Windapple

Letzte Änderung: 8. Januar 2006

Verwendet: Version 6.x, einfacher Modus (simple mode)

Niveau: Anfänger

Dieser Text steht unter einer Creative Commons Lizenz:

<http://creativecommons.org/licenses/by-nd/2.0/de/>

Dieses Tutorial hilft denen die den *Game Maker* noch nie zuvor genutzt haben und blutige Anfänger sind. Es führt dich Schritt für Schritt durch das Erstellen deines ersten Spieles, dies ist übrigens der schwierigste Teil. Um ein erstes Spiel machen zu können müssen einige grundlegende Arbeitsweisen mit dem *Game Maker* erlernt werden. Lese dieses Tutorial also sorgfältig und versuche jeden Schritt zu verstehen und nicht nur motorisch nachzubauen. Wenn du fertig bist wird dir dein zweites Spiel schon viel leichter fallen, da du die Grundkenntnisse dann besitzt.

Die Spielidee

Es ist immer wichtig erst mal eine genaue Beschreibung des Spieles anzufertigen, das man machen möchte. Weil das dein erstes Spiel ist wagen wir mal etwas ganz einfaches, es soll den Spieler nur für eine kurze Zeit bei Laune halten. Unser Spiel wird ein Actionspiel mit dem Titel *Fang den Clown!* (Versuche immer einen kurzen, einprägsamen, beschreibenden und schönen Titel für dein Spiel zu finden, kryptische Kürzel und unbekannte Wörter schrecken Spieler nur ab.) Hier ist die Beschreibung unseres Spieles:

Fang den Clown!

Fang den Clown! ist ein kleines Actionspiel, in dem sich ein Clown in einem Spielfeld herumbewegt. Das Ziel des Spielers ist den Clown zu fangen indem er mit der Maus auf ihn klickt. Je öfter der Spieler den Clown gefangen hat umso schneller bewegt er sich, dadurch wird es schwieriger ihn zu erwischen. Für jeden Fang gibt es Punkte und das Spielziel ist es so viele Punkte wie möglich zu bekommen. Die erwartete Spielzeit beträgt nur ein paar Minuten.

Natürlich haben Spiele wie dieses nur sehr begrenzte Anziehungskraft. Aber wir müssen einfach anfangen. Später können wir noch neue Features einbauen um das Spiel interessanter zu gestalten.

Ein Designdokument

Der zweite Schritt beim Erstellen des Spieles ist das schreiben eines genaueren Designdokumentes. Ich rate dir immer dies zu tun, auch wenn es ein sehr einfaches Spiel wird. Hier ist das Designdokument für unser *Fang den Clown!*. (Ich hab die oben schon gegebene Beschreibung weggelassen.)

Fang den Clown! *Designdokument*

Spielobjekte

Es wird nur zwei Objekte geben: Clown und Wand. Das *Wand*-Objekt hat ein quadratisches Bild. Die Wand, die den Spielbereich umschließt, besteht aus diesem Objekt. Sie tut nichts und ist nur dazu da den Clown am Verlassen des Spielfeldes zu hindern. Das *Clown*-Objekt hat das Bild eines Clowngesichtes. Es bewegt sich mit fixer Geschwindigkeit. Immer wenn es ein Wand-Objekt berührt wechselt es die Richtung. Wenn der Spieler auf den Clown klickt bekommt er 10 Punkte, der Clown wird an eine zufällige Position gesetzt und die Geschwindigkeit wird etwas erhöht.

Sounds

In diesem Spiel verwenden wir zwei Sounds. Ein Abprall-Sound wenn der Clown die Wand berührt und ein Klick-Sound wenn der Spieler geschafft hat auf den Clown zu klicken.

Steuerung

Die einzige Steuermöglichkeit des Spielers ist die Maus. Das Klicken auf den Clown mit der linken Maustaste fängt ihn.

Spielfluss

Am Anfang des Spieles wird die Punktzahl auf 0 gestellt. Der Raum mit dem sich herumbewegenden Clown wird gezeigt. Das Spiel beginnt sofort. Wenn der Spieler die <Esc> Taste betätigt beendet sich das Spiel.

Level

Es gibt nur ein Level. Die Schwierigkeit wird dadurch erhöht, indem bei jedem erfolgreichen Fang die Geschwindigkeit des Clowns erhöht wird.

Das sollte für den Augenblick genügen. Nun können wir mit dem Erstellen des Spieles anfangen. Starte also den *Game Maker* und lege los. Beachte das dieses Tutorial Version 6.x des *Game Makers* nutzt. Wenn du eine andere Version hast können die Bilder etwas anders aussehen. Es geht auch davon aus das Game Maker im einfachen Modus läuft. Du kannst zwischen einfachem und erweitertem Modus im **File** Menü mit dem Menüpunkt **Advanced Mode** wechseln. Im erweiterten Modus gibt es viel mehr Möglichkeiten und erweiterte Menüs, aber wir machen ein einfaches Spiel und da genügt uns der einfache Modus.

Das Spiel das wir erstellen werden liegt im Ordner `example`, zusammen mit allen Sprites und Sounds die benötigt werden. Du kannst es von dort laden, aber ich empfehle das Spiel selbst zu erstellen um den maximalen Lerneffekt zu erzielen. Auf diese Weise verstehst du besser wie ein Spiel mit dem *Game Maker* gemacht wird.

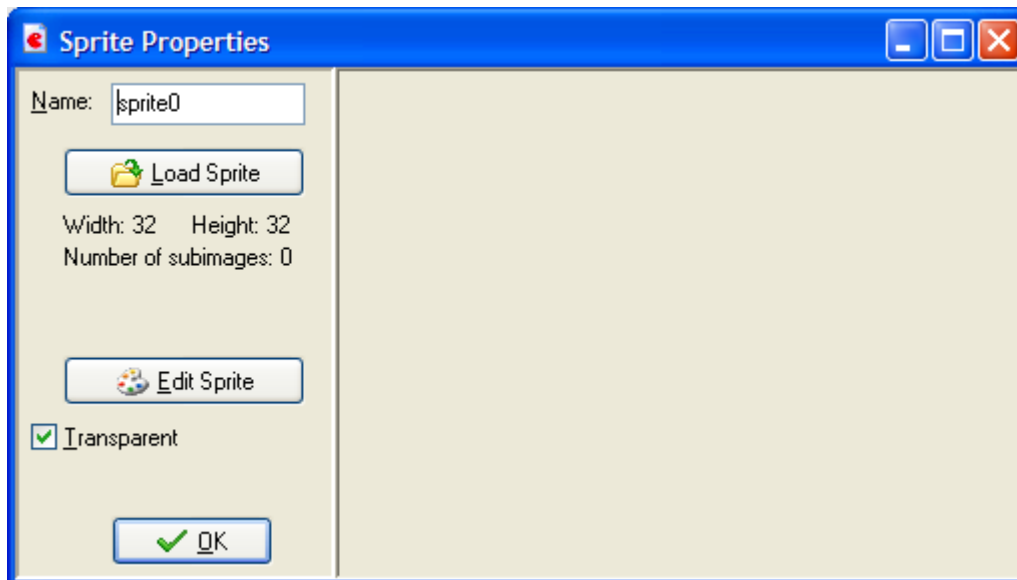
Sprites und Sounds Hinzufügen

Wie schon im Designdokument beschrieben benötigen wir zwei Bilder für zwei Objekte. Solche Bilder werden im *Game Maker* Sprites genannt. Es gibt vieles was man über Sprites wissen sollte, aber im Moment reicht es sich diese als kleine Bildchen vorzustellen. Wir müssen also solche Bilder machen oder finden. Um solche selber zu machen kannst du ein beliebiges Zeichenprogramm verwenden, z.B. das in Windows integrierte Paint. Aber der *Game Maker* besitzt auch ein eigenes kleines Tool für diesen Zweck. Gutmachende Sprites machen zu können ist eine Kunst die viel Übung und Talent benötigt. Glücklicherweise finden sich im Internet große Sammlungen von freien Sprites aller Arten. *Game Maker* bringt schon einige mit, und auf der *Game Maker* Webseite (www.gamemaker.nl) kannst du viele mehr herunterladen.

Alternativ kannst du die Suchmaschine deines Vertrauens nutzen und eines der vielen freien Archive für solche Sprites finden und besuchen. Für unser kleines Spiel nehmen wir zwei Sprites von der mitgelieferten Sammlung.

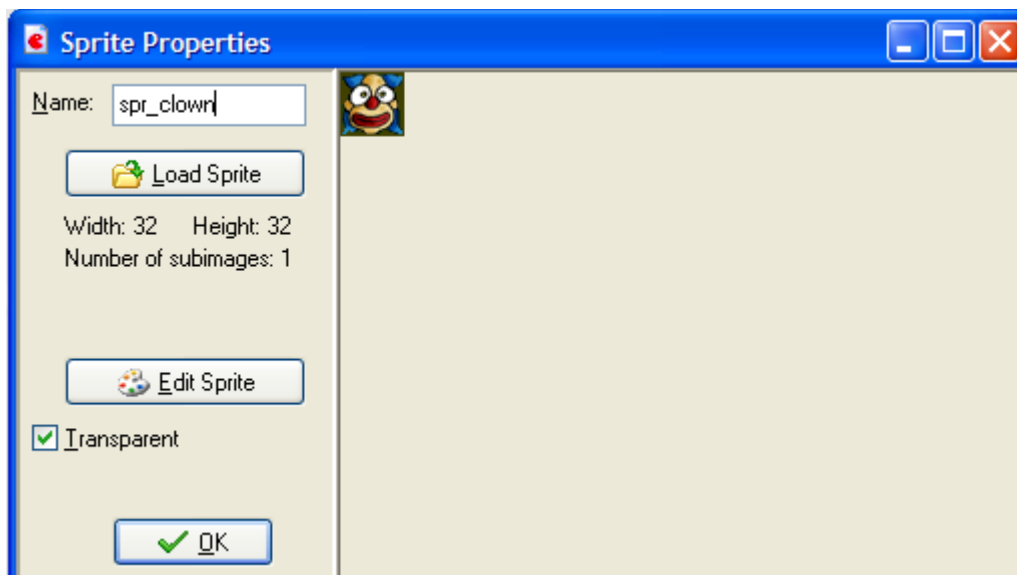
Der Clown:  Die Wand: 

Um unserem Spiel diese Sprites hinzuzufügen klicke im Menü **Add** auf **Add Sprite**. Der folgende Dialog erscheint:



Betätige den **Load Sprite** Knopf und wähle die Datei aus, die das Clown Sprite enthält. (Du findest es im Ordner `resources` im `example` Ordner.) Das Sprite wird im Dialog angezeigt. Fülle das **Name** Feld mit einem guten Namen für das Sprite, wie `spr_clown`. (Ich beginne meine Spritenamen immer mit `spr_` damit ich immer weiß das es Namen von Sprites sind. Solche Benennungsregeln sind gut im sich auch später immer noch im Spiel orientieren zu können.)

Beachte, die Eigenschaft **Transparent** ist aktiviert. Diese Eigenschaft macht die Hintergrundfarbe des Sprites transparent (du siehst den Hintergrund an dieser Stelle). Allgemein wirst du wohl in jedem Sprite einen transparenten Hintergrund haben wollen.

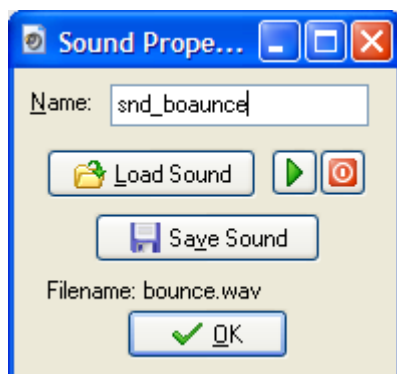


Betätige den **OK** Knopf und das Sprite wird hinzugefügt. Dasselbe tust du mit dem Sprite der Wand, nenne es `spr_wall`.

Wie du vielleicht schon bemerkt hast sind auf der linken Seite des *Game Maker* Fensters die Namen der beiden Sprites erschienen. Hier findest du immer alle Sprites, Sounds, Objekte, Räume usw. die du definiert hast. Alle zusammen heißen *Ressourcen* des Spieles. Du kannst eine Ressource auswählen indem du auf ihren Namen klickst. Nun kannst du das **Edit** Menü zum Ändern, Verdoppeln, oder Löschen der Ressource verwenden. Ein Rechtsklick auf eine Ressource zeigt das selbe Menü. Dieser Überblick über die Ressourcen ist wichtig, vor allem später wenn du komplexere Spiele erstellst.

Nun, nachdem wir die beiden Sprites hinzugefügt haben machen wir dies mit den beiden Soundeffekten. Der eine wird abgespielt wenn der Clown die Wand berührt, der andere wenn der Spieler den Clown gefangen hat. Wir werden zwei Wavedateien hierfür verwenden. Wavedateien sind für kurze Soundeffekte hervorragend geeignet. Einige Soundeffekte sind Teil des *Game Makers* und viele mehr können im Internet gefunden werden. Du kannst sie natürlich auch selber machen, das fordert aber einiges an Können.

Um den ersten Abprallsound hinzuzufügen wähle **Add Sound** im **Add** Menü. Es erscheint ein Dialog in dem du die Eigenschaften des Sounds ändern kannst. Betätige den **Load Sound** Knopf um den korrekten Sound zu laden. Wir nennen den Sound `snd_bounce`. Der Dialog sieht nun wie folgt aus:



Du kannst den Play Knopf, der mit dem nach rechts zeigenden Dreieck, verwenden um den Sound abzuspielen. Drücke **OK** um den Sound in dein Spiel hinzuzufügen. Du kannst den Klicksound auf die selbe Art und Weise hinzufügen, benenne ihn `snd_click`. Und siehe da, die beiden Sounds werden auch in der Auflistung der Ressourcen angezeigt.

Objekte und Aktionen

Das Hinzufügen von Sprites und Sounds heißt nicht das sich im Spiel irgendetwas tut. Sprites sind nur Bilder für Spielobjekte, und diese haben wir bisher nicht erstellt. Auch die Sounds werden nur abgespielt wenn man es ihnen mitteilt. Also definieren wir als nächstes unsere beiden Spielobjekte.

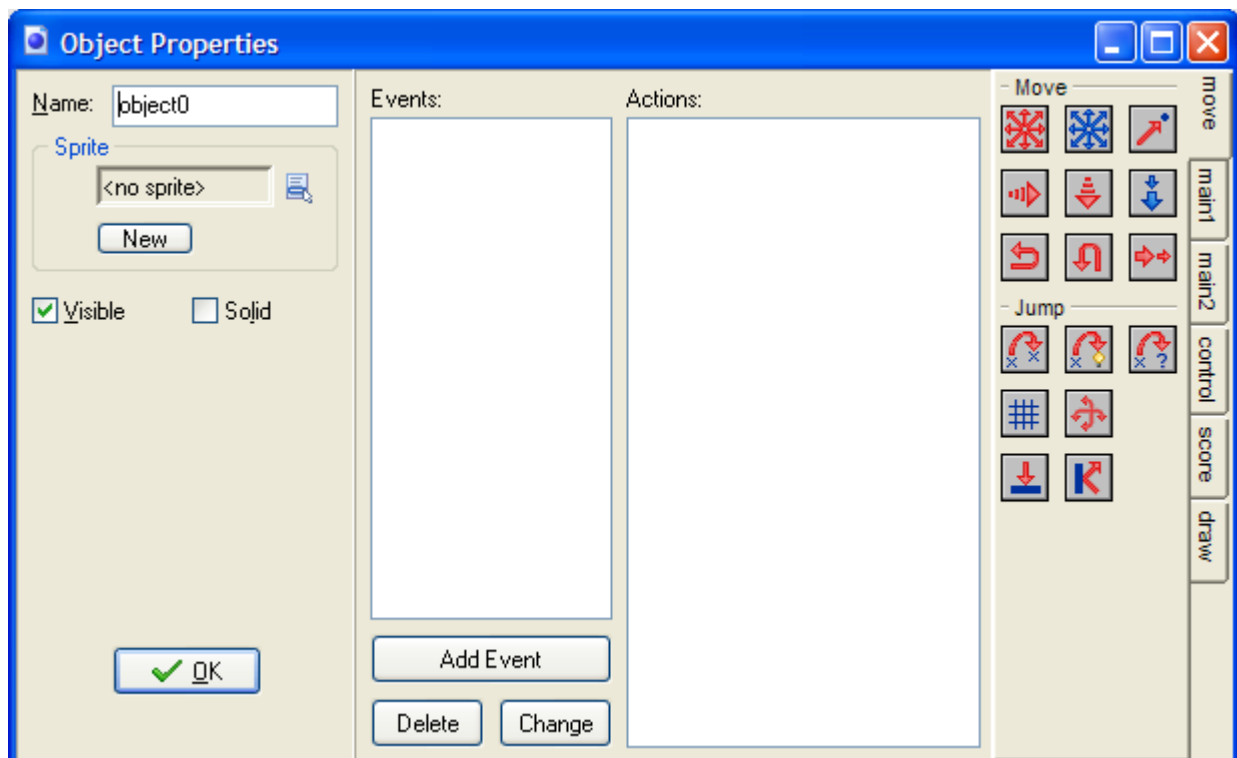
Bevor wir das tun müssen wir aber verstehen wie der *Game Maker* und die meisten anderen Systeme zum Erstellen von Spielen funktionieren. Wie bereits erwähnt gibt es im Spiel eine Anzahl von verschiedenen *Spielobjekten*. Während des Spiels wird eine oder mehrere *Instanzen* dieser Spielobjekte auf dem Bildschirm präsent, oder allgemeiner gesagt, in der Spielwelt existieren. Wissenswert ist es hierbei, das es mehrere Instanzen eines Objektes geben kann. In unserem *Fang den Clown!* gibt es beispielsweise viele Instanzen des Wandobjektes rund um das Spielfeld aber nur eine Instanz des Clownobjektes.

Instanzen tun eigentlich gar nichts solange man ihnen nicht beibringt etwas zu tun. Dies erreichst du indem du den Instanzen der verschiedenen Objekte sagst wie sie auf bestimmte *Ereignisse* reagieren sollen. Es gibt viele Ereignisse die auftreten können. Das erste wichtige ist jenes wenn die Instanz erstellt wird. Dies ist das **Creation Ereignis**. Möglicherweise wird hier schon eine Reaktion benötigt. Dem Clownobjekt muss mitgeteilt werden das es sich in eine bestimmte Richtung bewegen soll. Ein anderes wichtiges Ereignis ist die Kollision; das sogenannte **Collision Ereignis**. Wenn z.B. der Clown mit der Wand zusammenstößt muss der Clown reagieren und seine Bewegungsrichtung ändern. Andere Ereignisse treten auf, wenn der Spieler eine Taste drückt oder mit der Maus auf eine Instanz klickt. Für den Clown benutzen wir ein **Mouse Ereignis** um den Mausklick auf ihn zu registrieren.

Um festzulegen was passiert wenn ein Ereignis auftritt gibt es *Aktionen*. Du kannst aus einer großen Auswahl an Aktionen wählen. Beispielsweise gibt es ein ganzes Set das die Bewegung eines Objektes beeinflusst, es gibt eine Aktion um die Punktzahl zu ändern und eine um Sounds abzuspielen. Ein Objekt zu definieren besteht also aus folgenden Bestandteilen: wir geben jedem Objekt einen Namen und ein Bild, wir stellen einige Eigenschaften ein, und wir bestimmen auf welche Ereignisse es reagiert und welche Aktionen dabei stattfinden sollen.

Beachte, das ich zwischen *Objekten* und *Instanzen* dieser Objekte unterscheide. Ein Objekt definiert ein bestimmtes Spielobjekt mit seinen Eigenschaften (das sind Reaktionen auf die Ereignisse). Von diesem Objekt können dann im Spiel eine oder mehrere Instanzen existieren. Diese Instanzen verhalten sich so wie es die Ereignisse und Aktionen vorschreiben. Anders definiert ist ein Objekt ein abstraktes Ding. Im echten Leben kann man auch über einen Stuhl als Abstraktum, als allgemeines Möbelstück reden, oder über einen bestimmten Stuhl welcher bei uns zuhause am Tisch steht.

Wie funktioniert das also in dem Spiel das wir erstellen wollen? Wir brauchen zwei Objekte. Beginnen wir mit dem sehr einfachen Wandobjekt. Dieses Objekt braucht kein Verhalten. Es reagiert auf keinerlei Ereignisse. Um das Objekt hinzuzufügen wähle **Add Object** im **Add Menü**. (Wie du möglicherweise schon herausgefunden hast gibt es dafür auch Knöpfe in der Toolbar.) Der folgende, vielleicht kompliziert wirkende Dialog, erscheint:



Auf der linken Seite kannst du einige Objekteinstellungen vornehmen. Benenne es `obj_wall`. Als nächstes müssen wir das korrekte Sprite auswählen. Um dies zu tun musst du das Icon im **Sprite** Bereich anklicken. Es erscheint ein Menü das alle vorhandenen Sprites auflistet (in diesem Fall also Clown und Wand). Wähle für dieses Objekt das Wandsprite. Zum schluss wollen wir natürlich das die Wand fest ist und kein Objekt hindurch kann. Markiere also die **Solid** Box indem du mit der Maus darauf klickst. Klicke auf **OK**, wir sind mit diesem Objekt fertig.

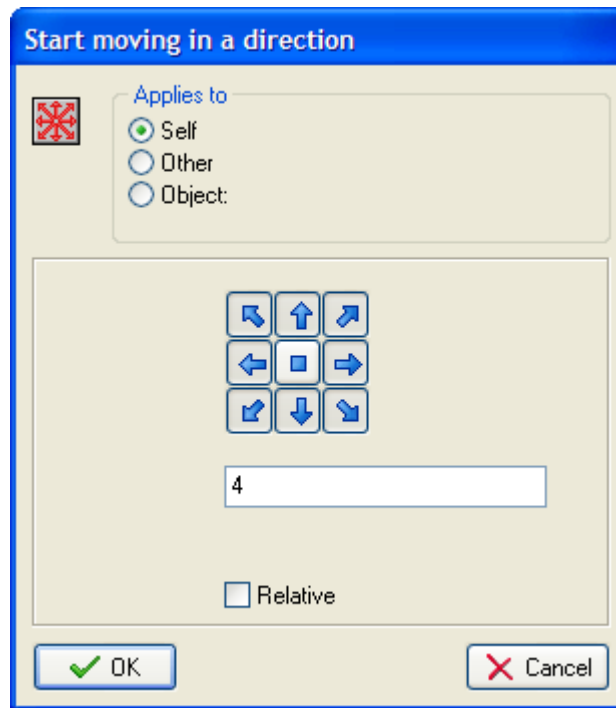
Mit dem Clownobjekt fangen wir genauso an. Füge ein neues Objekt hinzu, benenne es `obj_clown`, und wähle das korrekte Sprite. Der Clown muss nicht solid sein. Für ihn muss aber noch einiges getan werden; Wir müssen sein Verhalten festlegen. Hierfür verwenden wir den Rest des Dialogs. In der Mitte siehst du ein leeres Feld mit drei Knöpfen darunter, die Liste enthält später die Ereignisse auf die das Objekt reagieren muss. Mit den Knöpfen darunter kannst du Ereignisse hinzufügen, ändern oder löschen. Es gibt eine ganze Menge an Ereignisse, aber in diesem Spiel brauchen wir nur einige wenige.

Neben der Ereignisliste ist die leere Liste mit den Aktionen die für das (sofern vorhandene) gewählte Ereignis ausgeführt werden. Ganz rechts befindet sich eine Registerkartei, in der die verschiedenen Aktionen geordnet aufgelistet sind. Die Icons stellen die Aktionen dar. Du kannst aus nahezu 100 unterschiedlichen Aktionen wählen. Wenn du mit dem Mauszeiger über eine Aktion fährst erscheint eine kurze Beschreibung, ziehst du dann das Icon mit gedrückter linker Maustaste in die Aktionsliste wird die Aktion dort eingefügt.

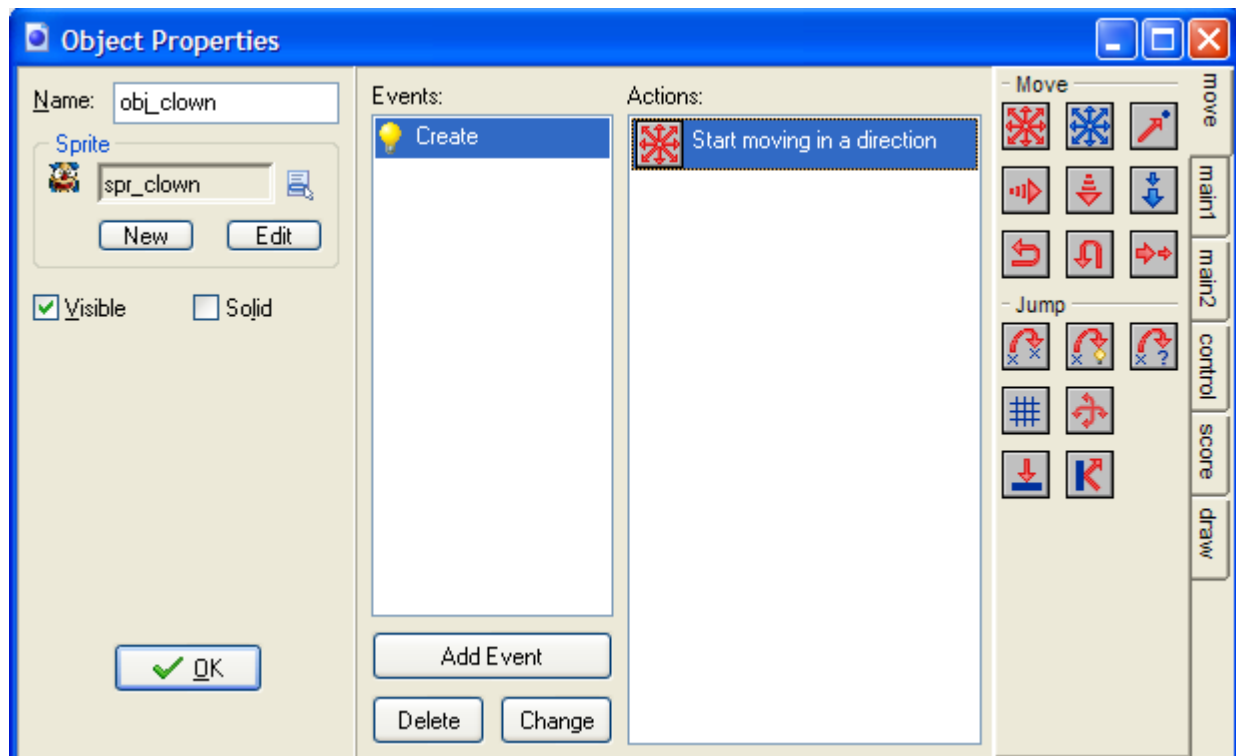
Zuerst definieren wir was passieren soll wenn eine Instanz des Clowns erstellt wird. Wir wollen ihn in diesem Fall in eine zufällige Richtung sich losbewegen lassen. Betätige den **Add Event** Knopf. Der folgende Dialog erscheint:



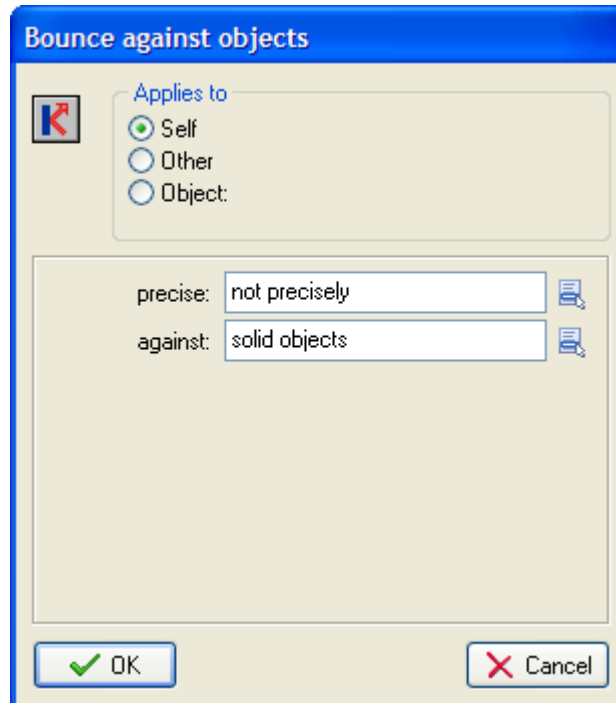
Klicke auf den **Create** Knopf und das Create Ereignis wird der Ereignisliste hinzugefügt und markiert. Nun ziehe die Aktion mit den acht roten Pfeilen in die Aktionsliste. Dieses Icon repräsentiert die Aktion, die das Objekt in eine bestimmte Richtung losbewegt. Der folgende Dialog erscheint:



Die meisten Aktionen besitzen einen Dialog in dem die Einstellungen vorgenommen werden können. Den Bereich, der mit **Applies to** beschriftet ist, lassen wir erst einmal ausser Acht. Wir können zwei Dinge bestimmen: Die Bewegungsrichtung und die Geschwindigkeit. Weil wir eine zufällige Richtung haben wollen musst du auf alle acht Pfeile klicken (nicht das Viereck in der Mitte, das stoppt eine Bewegung!). Das bedeutet nun, dass aus allen acht möglichen Richtungen zufällig eine gewählt wird. Die Geschwindigkeit 4 ist in Ordnung, eine höhere macht den Clown zu schnell. Klicke auf **OK** um die Einstellungen dieser Aktion zu bestätigen. Der Objekteigenschaftsdialog sollte nun so aussehen:

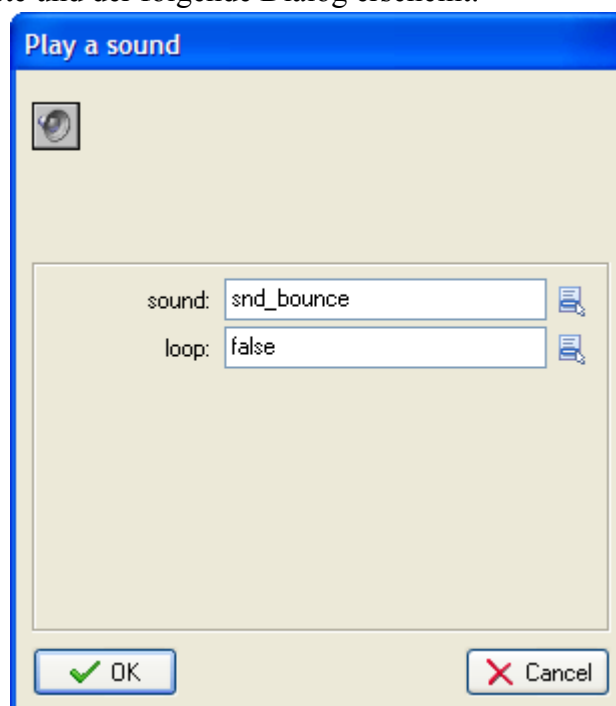


Das nächste zu definierende Ereignis ist die Kollision mit der Wand. Klicke abermals auf **Add Event**, wähle aber diesmal **Collision**. Es erscheint ein Menü mit allen verfügbaren Objekten (die Wand und der Clown). Wähle das Wandobjekt aus, bei diesem wollen wir ja die Kollision überprüfen. Für dieses Ereignis benutzen wir die Abprall Aktion, jene bei der ein Pfeil von einer Wand abprallt (rechtes unterstes Icon auf der ersten Registerkarte). Ziehe es in die Aktionsliste und der folgende Dialog erscheint:

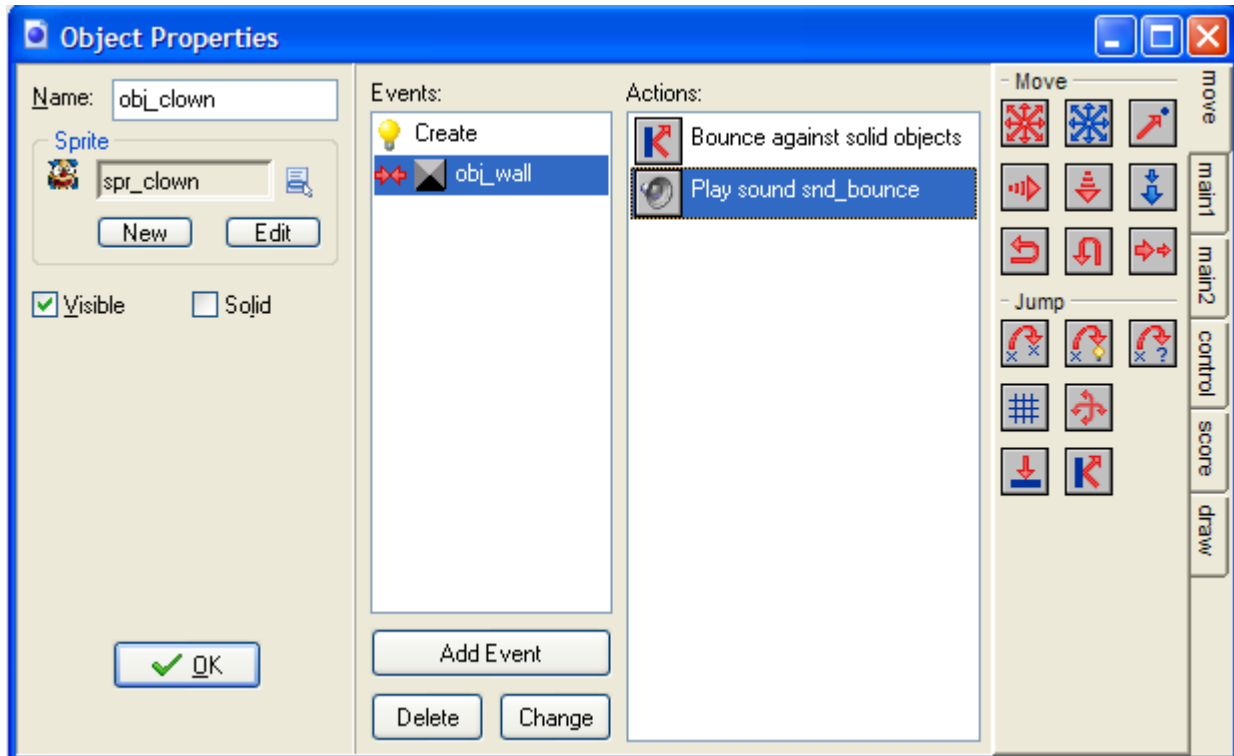


Hier gibt es zwei Eigenschaften, ihre Standartwerte sind in Ordnung. Wir benötigen kein genaues Abprallen und wollen dies nur bei soliden Objekten. (Erinnere dich, wir haben die Wand solide gemacht.) Klicke auf **OK** und wir sind fertig.

Wir müssen auch eine Aktion zum Abspielen des Soundeffektes beim Abprallen hinzufügen. Die Soundaktion kann auf dem register **main1** gefunden werden und sieht wie ein Lautsprecher aus. Ziehe es in die Aktionsliste und der folgende Dialog erscheint:

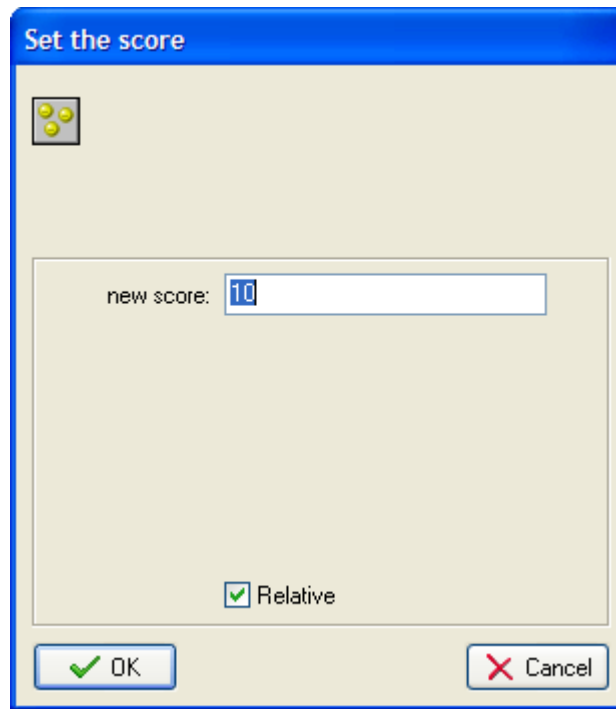


Auch hier können zwei Einstellungen vorgenommen werden. Zuerst muss der Sound zum Abspielen ausgewählt werden. Wir brauchen `snd_bounce`. (Langsam beginnst du zu verstehen warum man seinen Ressourcen einprägsame Namen geben sollte.) Die zweite Einstellung bestimmt darüber, ob der Sound in einer Endlosschleife wiederholt wird oder nur einmal gespielt wird. Es ist ja offensichtlich das wir das nicht wollen, also belassen wir es bei **false**. Klicke auf **OK**. Der Objekteigenschaftsdialog sollte nun folgendermaßen aussehen:



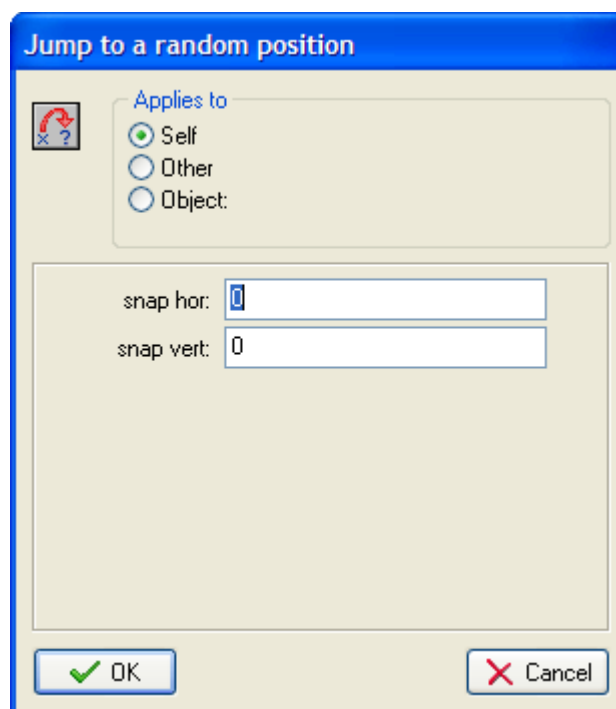
Es sind zwei Aktionen die beide (in deren Reihenfolge) ausgeführt werden wenn eine Kollision auftritt. Wenn du einen Fehler gemacht haben solltest kannst du mit der rechten Maustaste auf eine Aktion klicken und **Delete** zum entfernen wählen (oder die Entf Taste auf deiner Tastatur benutzen). Du kannst auch **Edit Values** zum ändern der Aktionseinstellungen wählen. (Auch per Doppelklick möglich.) And you can drag them up and down to change the order in which they are executed.

So, nun müssen wir noch festlegen was passiert wenn der Spieler mit dem Mauszeiger auf den Clown klickt. Klicke wiederum auf den **Add Event** Knopf und wähle nun die **Mouse** Ereignisse. Es öffnet sich wieder ein Menü mit vielen Auswahlmöglichkeiten. Diejenige die wir brauchen heißt **Left pressed**. Dieses Ereignis tritt auf wenn der Spieler mit der linken Maustaste auf die Instanz klickt. Andere Mausereignisse treten immer auf wenn eine Maustaste betätigt wird, oder auch wenn es nicht über einer bestimmten Instanz stattfindet. Hier brauchen wir vier Aktionen. Zuerst müssen wir die Punktzahl erhöhen. *Game Maker* hat einen automatischen Mechanismus um die Punktzahl zu speichern und anzuzeigen. Um die Punktzahl zu erhöhen musst du die Aktion mit den drei gelben Münzen vom **score** Register in die Aktionsliste ziehen. Der folgende Dialog erscheint:

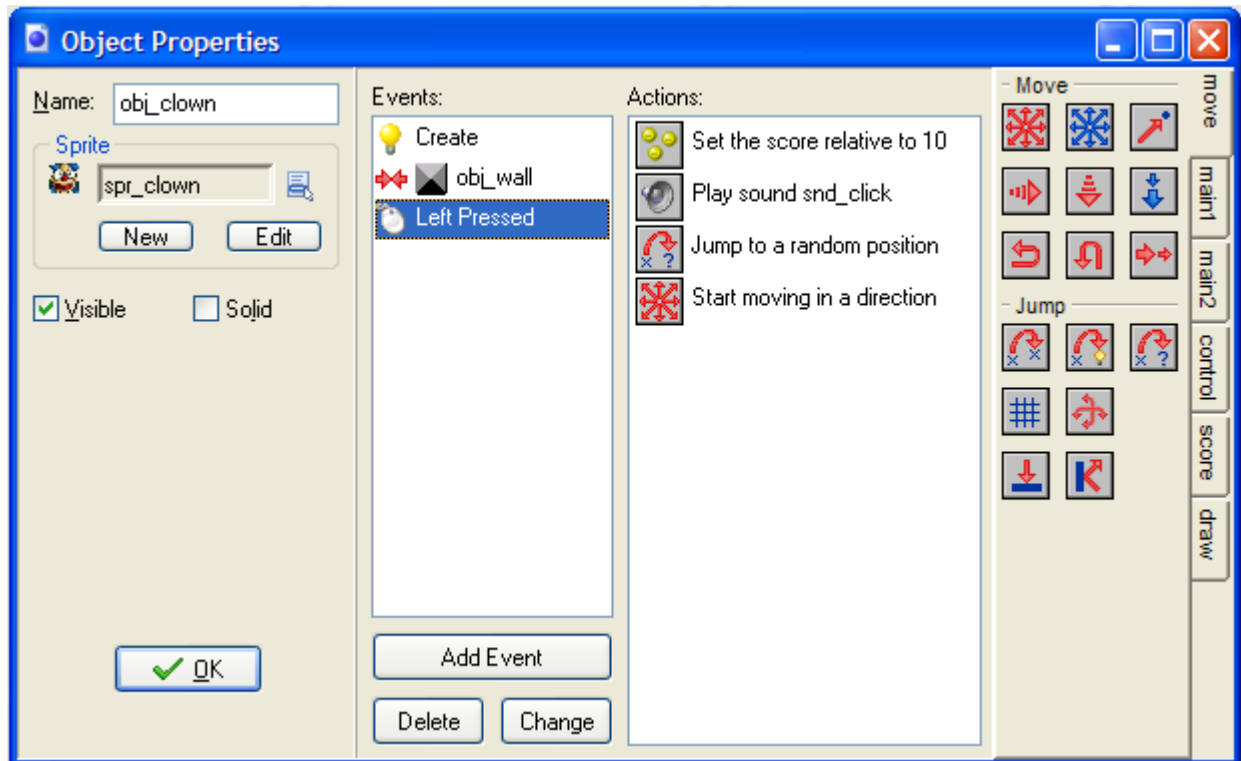


Gebe 10 in das Feld ein, so wie hier gezeigt, damit die Punktzahl um 10 erhöht wird. Dies ist aber nicht alles. Wir wollen die Punktzahl nicht auf 10 stellen sondern 10 Punkte hinzufügen. Dies erreicht man mit einem Häkchen bei **Relative**. Viele Aktionen haben solche Relative Auswahlboxen um einen Wert zu erhöhen statt ihn festzulegen. Klicke auf **OK** um den Dialog zu schließen.

Die nächste Aktion soll den Sound abspielen. Hier gehen wir genauso wie beim Abprallsound vor. Dann soll der Clown noch an eine zufällige Stelle springen. Das macht das Spiel schwerer. Im Register **Move** findest du dafür eine Aktion. Ziehe die Aktion mit dem Fragezeichen in die Aktionsliste. Der folgende Dialog erscheint:



Die Standartwerte sind in Ordnung, klicke **OK** um den Dialog zu schließen. Als letztes wollen wir dem Clown noch etwas Beine machen und die Geschwindigkeit erhöhen, so wie im Designdokument erwähnt. Dafür benutzen wir wieder die Aktion mit den roten Pfeilen. Klicke erneut auf alle acht Pfeile und geben bei speed 0.5 ein, machen ein Häckchen bei **Relative**. Die Objekteigenschaften sollten nun folgendermaßen aussehen:

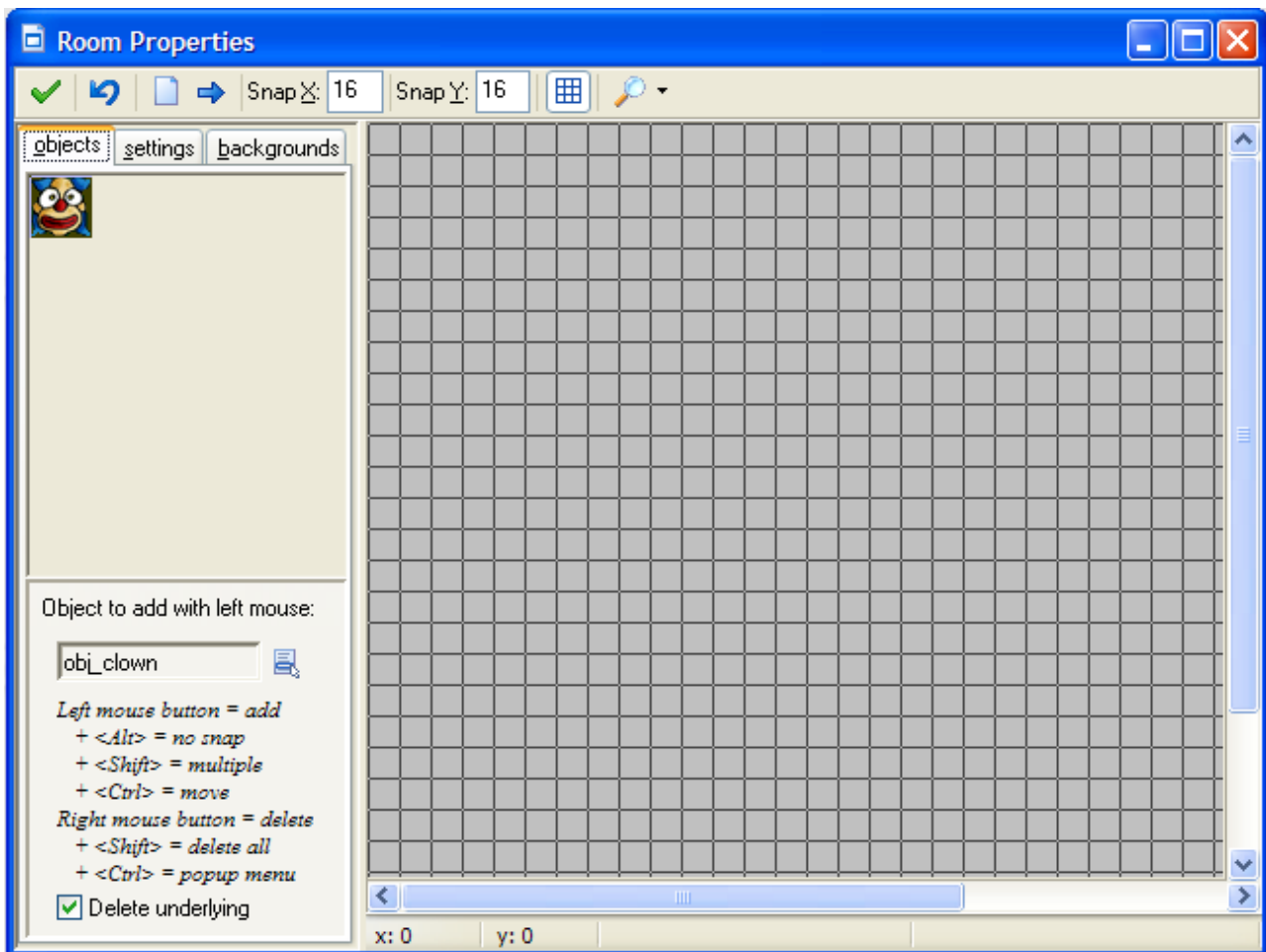


Wir sind nun mit dem Clownobjekt fertig und haben Aktionen für die drei wichtigen Ereignisse definiert. Klicke auf **OK** um den Dialog zu schließen.

Erstellen des Raumes

Nun sind die Spielobjekte einsatzbereit und wir müssen uns noch um den Raum kümmern, in dem das Spielgeschehen stattfindet. Für die meisten Spiele ist das designen von Räumen (auch Level genannt) ein langer Prozess da immer die richtige Balance aus Schwierigkeit und Fortkommen gefunden werden muss. Ein Raum für *Fang den Clown!* ist aber ganz einfach gestrickt: eine von einer Wand umgebenen Fläche mit dem Clown darin.

Um einen Raum zu erstellen musst du **Add Room** im **Add** Menü anklicken. Der folgende Dialog erscheint:



Auf der linken Seite haben wir drei Registerkarten. Wir benutzen hier nur das **objects** Register. Auf der rechten Seite siehst du den aktuellen Raum welche nur eine graue Fläche mit einem Raster darauf ist, das Raster hilft dabei Objekte zu platzieren. Weil unsere Spielobjekte die Größe 32x32 haben sollten wir das Raster etwas ändern. Füge einfach oben wo zwei mal 16 steht jeweils 32 ein. Die Zellen des Rasters sind nun größer.

Um ein bestimmtes Objekt auszuwählen klickst du auf das Menüicon. Wähle `obj_wall` aus. Nun kannst du die Wand in die Zellen am Rand des Raumes platzieren. Um den ganzen Raum zu sehen kannst du das Fenster vergrößern oder die Scrollleisten benutzen. Wenn du die `<Shift>` Taste während des Bewegens der Maus gedrückt hältst werden mehrere Instanzen in den Raum gesetzt. Sollte dir ein Fehler passiert sein kannst du die Instanz mit einem Klick der rechten Maustaste entfernen. Zum Schluss wählst du `obj_clown` und setzt eine Instanz mitten in den Raum. Das war alles, der Raum ist fertig. Schließe das Fenster mit einem Klick auf den **OK** Knopf; Das ist der Knopf mit dem grünen V links oben.

Speichern und Testen

Vielleicht hast du es noch nicht bemerkt, aber unser Spiel ist fertig. Sounds und Sprites wurden hinzugefügt, Spielobjekte kreist und der erste Raum erzeugt, in dem das Spiel stattfindet. Nun speichern und testen wir das Spiel. Speichern funktioniert so wie in jedem anderen Windows Programm. Wähle **Save** im **File** Menü, suche einen Pfad aus und drücke auf Speichern. *Game Maker* Spiele haben die Dateierweiterung `.gm6`. (Steht für Game Maker version 6.) Solche Spiele kann man aber nicht direkt ausführen, nur in den *Game Maker* laden. Wie man ein eigenständiges Spiel erstellt wird weiter unten erklärt.

Als nächstes testen wir das Spiel. Testen ist immer notwendig! Du kannst es selber ausprobieren, solltest aber auch andere bitten es zu tun. Testen (oder das Spiel überhaupt ausführen) ist einfach; Wähle **Run normally** im **Run** Menü. Das Designfenster verschwindet, das Spiel wird geladen und der Raum erscheint, wenn du keine Fehler gemacht hast, mit dem sich darin bewegenden Clown. Klicke auf ihn um zu schauen ob das Spiel sich wie erwartet verhält. Der Sound sollte zu hören sein und der Clown sich schneller bewegen. Zum Beenden des Spieles drückst du die <Esc> Taste oder klickst auf das X im Fenstertitel. Das Designfenster erscheint wieder.



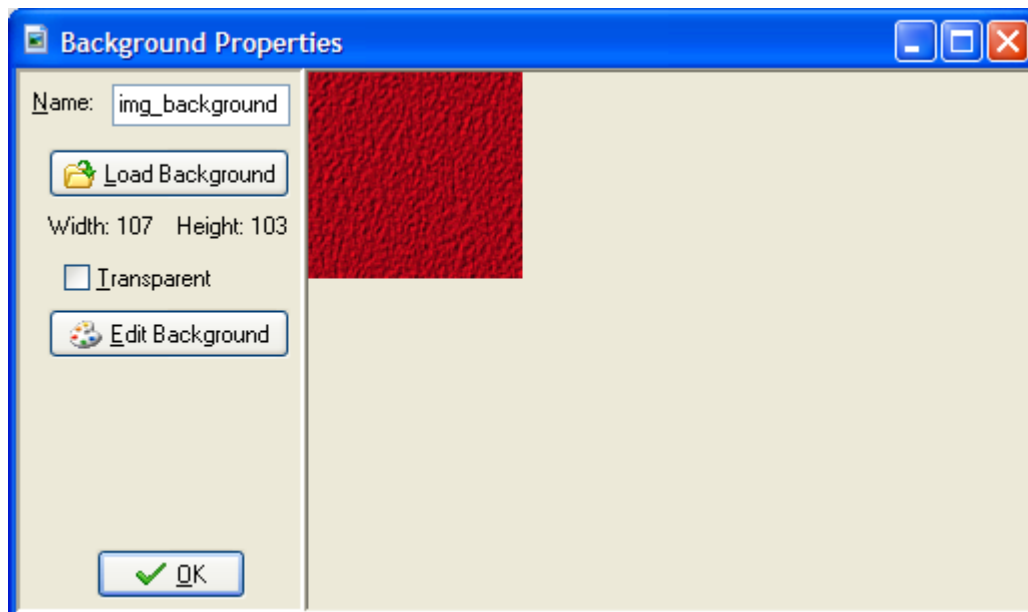
Nun ist Zeit für den Feinschliff. Du solltest dir z. B. folgende Fragen stellen: Ist die Anfangsgeschwindigkeit korrekt? Ist der Geschwindigkeitszuwachs korrekt? Ist die Raumgröße korrekt? Sind das die richtigen Sprites und Sounds für dieses Spiel? Wenn die etwas nicht passt ändere es und teste erneut. Vergesse nicht es auch mal jemand anderen testen zu lassen, für dich als Autor des Spieles mag es einfacher sein als für unbeteiligte Spieler.

Wenn alles zu deiner Zufriedenheit ist kannst du eine Exe aus dem Spiel machen. Dies ist eine Version deines Spieles die ohne den *Game Maker* funktioniert. Das geht ganz einfach. Im **File** Menü wählst du **Create executable**. Du musst nun ein Verzeichnis für die Exe angeben und speichern. Nun kannst du den *Game Maker* schließen und die Exe starten. Diese kannst du auch weitergeben, beispielsweise an Freunde oder auf deiner Webseite.

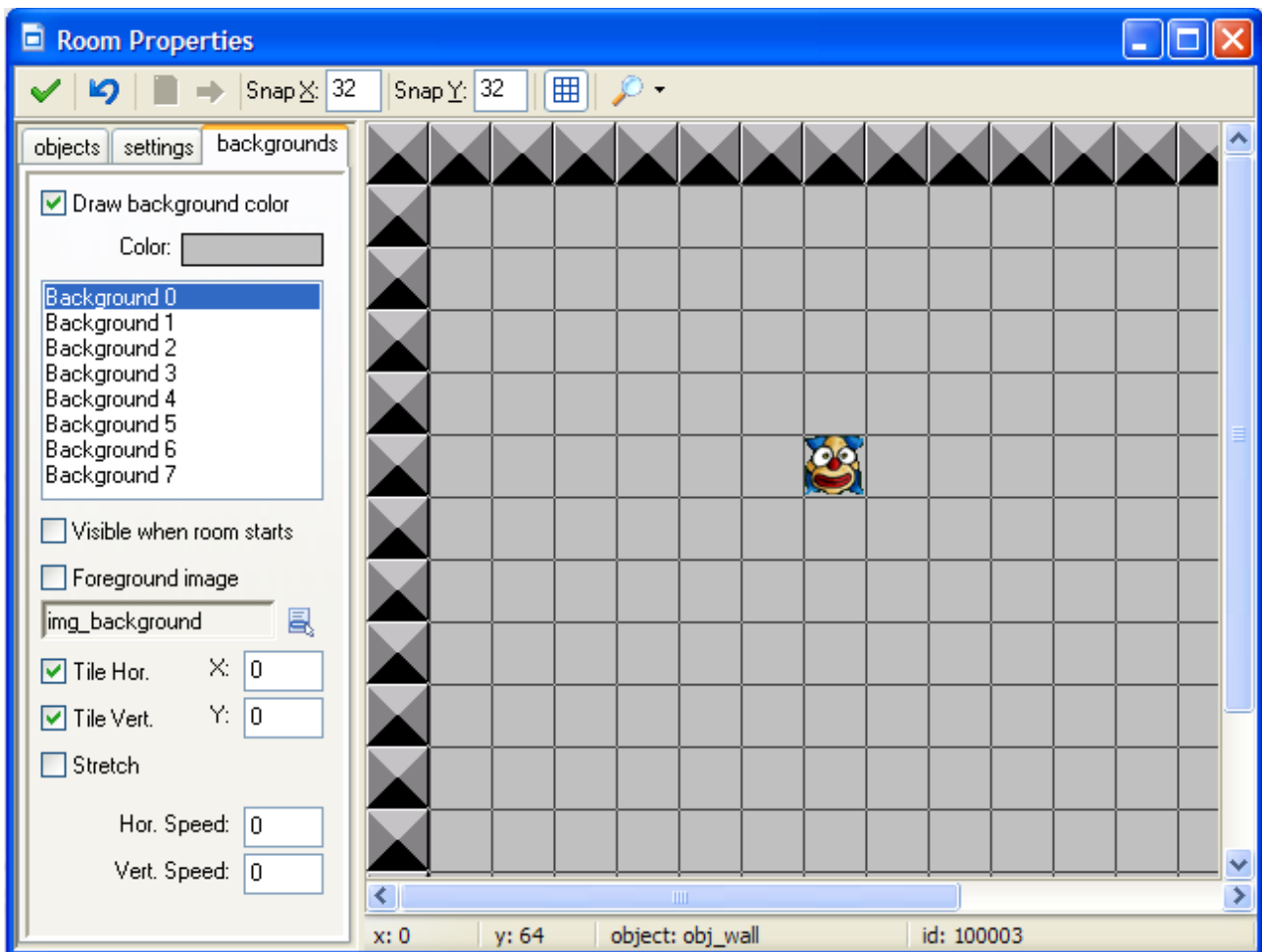
Abschließende Handgriffe

Unser erstes Spiel ist fertig, aber es wird mit einigen Handgriffen noch schöner. Zuerst brauchen wir noch etwas Hintergrundmusik. Das ist sehr einfach. Wir fügen einfach eine Soundressource namens `snd_background` hinzu. Als Hintergrundmusik verwendet man keine Wav Dateien, die sind dazu einfach zu groß. Am besten geeignet sind da Midi Dateien, welche eine andere Methode zum Speichern von Musik sind. Du kannst eine Menge von ihnen im Internet finden. Die Musik sollte aber dann schon zum Spiel passen, das ist wichtig. Im Create Ereignis des Clowns fügen wir einfach noch eine Aktion zum Abspielen der Hintergrundmusik hinzu. Dieses mal wählen wir **Loop** aus damit die Hintergrundmusik in einer Endlosschleife läuft.

Als zweites fügen wir ein Hintergrundbild ein, der graue Hintergrund ist doch zu langweilig. Dazu benutzen wir eine neue Ressourcenart, die Background Ressource. Im **Add Menü** wählst du **Add Background**. Ein Dialog ähnlich dem bei Sprites erscheint. Klicke auf den **Load Background** Knopf und wähle ein schönes Bild aus. Der Dialog sieht dann wie folgt aus:

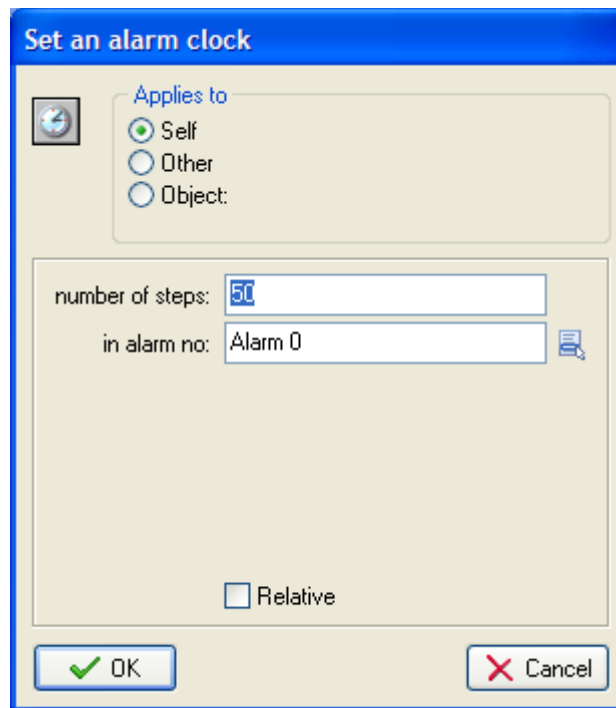


Schaue das **Transparent** ausgeschaltet ist. Klicke auf **OK**. Um dem Raum den Hintergrund zu geben musst du ihn wieder öffnen. Doppelklicke einfach auf die Raumressource. Nun benutze das Register **backgrounds** das sich links oben befindet. Die folgenden Informationen werden angezeigt:



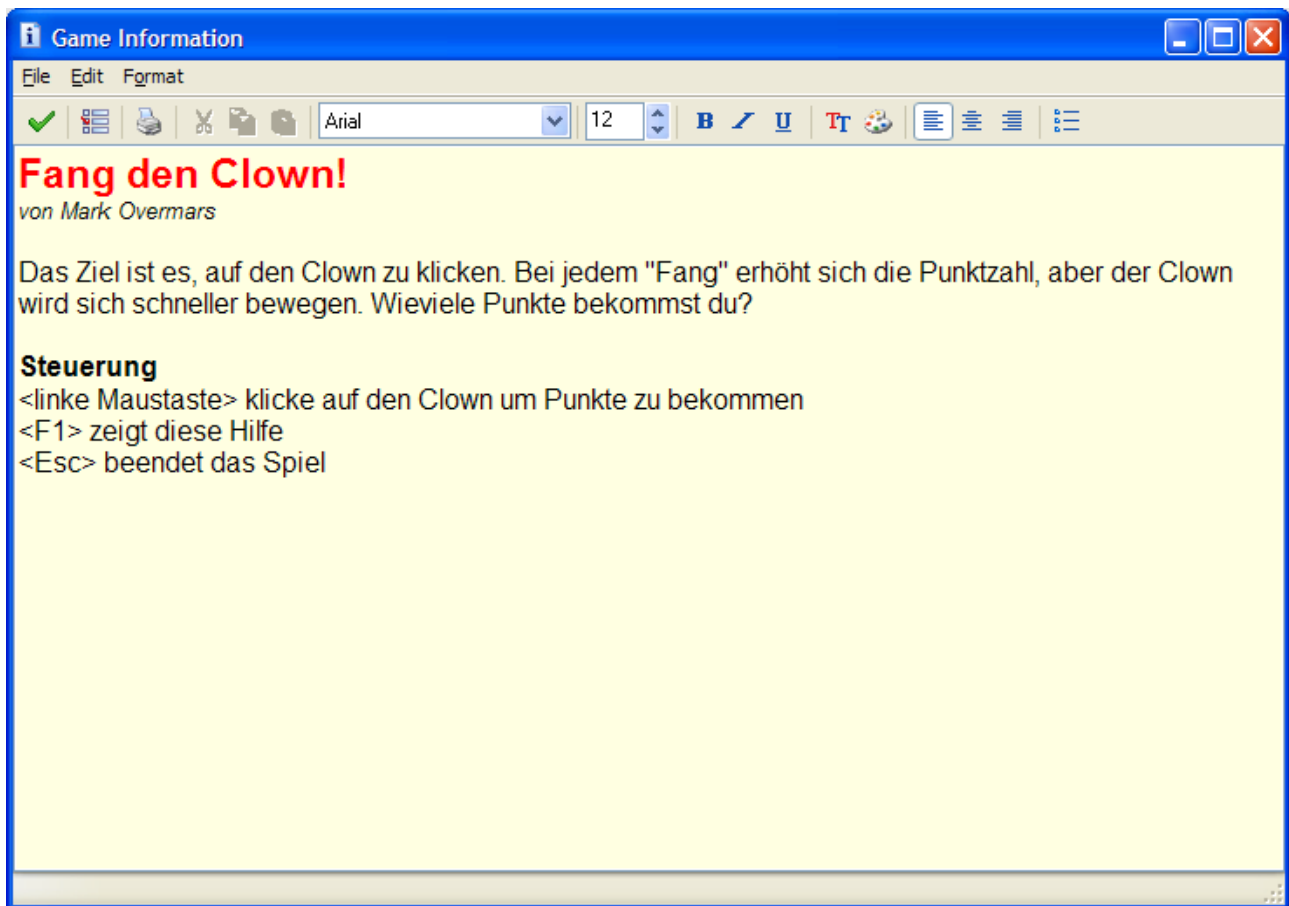
Entferne das Häkchen bei **Draw background color** weil der graue Hintergrund nicht mehr gezeichnet werden muss. Klicke dann auf das Menüicon und wähle den Hintergrund aus. Nun hat der Raum einen schönen Hintergrund. Werfe auch einen Blick auf die Einstellungen **Tile Hor.** und **Tile Vert.** Sie geben an, dass der Hintergrund horizontal und vertikal wiederholt wird um den ganzen Raum zu füllen. Um dabei gut auszusehen muss das Hintergrundbild aber dafür entworfen sein. Glücklicherweise gibt es viele solche Bilder.

Wenn du das Spiel eine Weile gespielt hast wirst du merken dass es zu leicht ist, man kann dem Clown einfach folgen. Der Clown sollte deswegen seine Bewegungsrichtung von Zeit zu Zeit ändern. Dazu verwenden wir einen Alarm. Alarme zählen bis null und lösen ein Alarmereignis aus. Jedes Objekt kann bis zu acht Alarme haben, aber wir brauchen nur einen. Im Creation Ereignis fügen wir die Aktion zum Erstellen eines Alarmes ein (Register **main2**). Wir benutzen Alarm 0 und setzen 50 Schritte ein (ungefähr 1.5 Sekunden weil das Spiel 30 Schritte pro Sekunde macht; dies kann bei **settings** im Raum geändert werden). Der Dialog sieht dann so aus:



Als nächstes definieren wir was passiert wenn der Alarm null erreicht. Klicke auf **Add Event**, wähle das **Alarm** Ereignis und Alarm 0 aus dem Menü. In diesem Ereignis setzen wir eine zufällige Richtung durch die Aktion mit den roten Pfeilen (Geschwindigkeit bleibt bei 0, markiere **Relative** um sie nicht zu ändern). Den Alarm 0 setzen wir auch wieder auf 50 damit das Ereignis nach 1.5 Sekunden wieder auftritt und der Clown seine Richtung wieder ändert.

Ganz zum Schluss sollte man dem Spieler noch sagen was er in diesem Spiel tun soll. Eine Hilfe wird benötigt. *Game Maker* bringt hierfür schon einen Mechanismus mit. Wähle **Change Game Information** aus dem **Add** Menü. Ein Texteditor erscheint, wo du deinen Text eingeben kannst. Du kannst verschiedene Schriftgrößen und Farben verwenden. Es könnte dann so aussehen:



Während des Spiels kann der Spieler F1 drücken damit die Hilfe angezeigt wird (wie auch in anderen Programmen).

Dein erstes Spiel ist fertig

Glückwünsche! Du hast dein erstes Spiel fertiggestellt. Und das erste Spiel ist immer das schwierigste. Dir wurden auch die wichtigsten *Game Maker* Bestandteile vorgestellt: Sprites, Bilder und Sounds, Spielobjekte, Ereignisse und Aktionen, und natürlich Räume. Diese sind nicht nur in *Game Maker*. Sprites, Bilder und Sounds sind wichtige Ressourcen in jedem Programm zum Spieleerstellen. Auch ist es in jedem anderen Spieleerstellungsprogramm nützlich die Objekte die auf Ereignisse reagieren und Aktionen ausführen zu kennen. Und natürlich arbeiten alle mit Leveln oder Räumen wie wir sie im *Game Maker* nennen.

Bevor wir ein weiteres Spiel beginnen möchtest du evtl. noch etwas mit dem *Fang den Clown!* herumspielen. Hier sind einige Anregungen:

- Lasse zwei Clowns herumlaufen. (Sehr einfach, erstelle eine weitere Instanz.)
- Füge einen dunklen Clown hinzu und lasse bei einem Klick auf ihn Punkte abziehen.