

Game Maker Tutorial 1

Das hier ist mein erstes Game Maker Tutorial. Wir werden ein Maze Game mit D&D programmieren. Die Standard Version ist nicht erforderlich. In dem Spiel muss man einen Spieler mit den Pfeiltasten durch die Levels zum Ziel bringen.

Was in diesem Tutorial vorkommt:

- Spieler
- Wand
- Ziel
- Spieler mit Pfeiltasten bewegen
- Spieler in einem Raster bewegen
- Kollisionen

Erst mal erkläre ich wie der Game Maker funktioniert:

Game Maker besteht aus Objekten, Sprites, Sounds, Backgrounds und Rooms. Rooms sind die "Räume" von den Spielen, also in diesem Spiel die Level. In ihnen werden die Objekte platziert. Objekte sind z.B. Charakter, Wände, Gegner, Items... Es sind einfach „Gegenstände“ die in dem Spiel etwas bestimmtes machen sollen, in ihnen schreibt man dann z.B. dass ein Objekt wenn man die Taste Rechts drückt nach rechts laufen soll. Sprites sind die Bilder die den Objekten zugeordnet werden, Sounds sind die Töne und die Musiken und Backgrounds natürlich die Hintergründe.

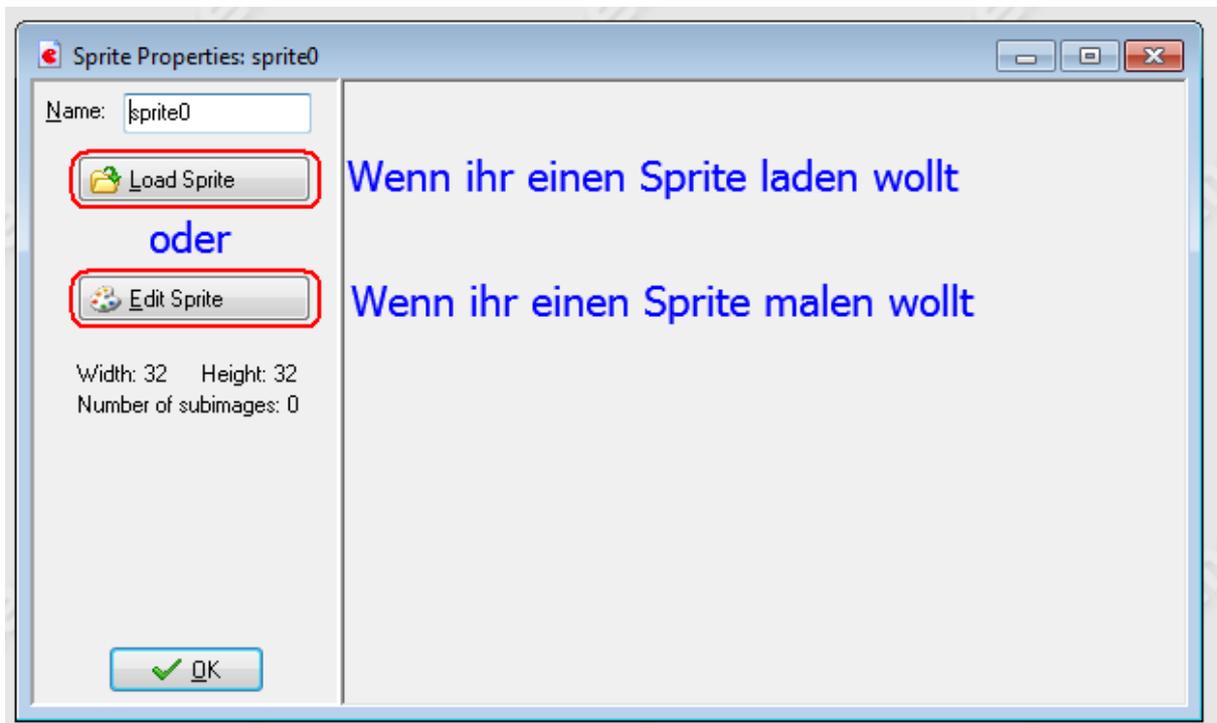
So dann kanns auch schon losgehen. Wir öffnen den Game Maker gehen auf „File“ und machen das Häkchen bei Advanced Mode weg, dann drücken wir auf das Blatt Papier um ein Neues Spiel anzufangen.

Sprites

Als erstes erstellen wir die Sprites, in dem wir auf den Roten Pacman klicken:



Nun könnt ihr auswählen ob ihr einen Sprite Laden oder zeichnen wollt. Ich habe mir einfach einen Smiley von den vorgemachten Game Maker Sprites genommen, aber wenn ihr das Spiel veröffentlichen wollt würde ich auf jedenfall selber welche malen.



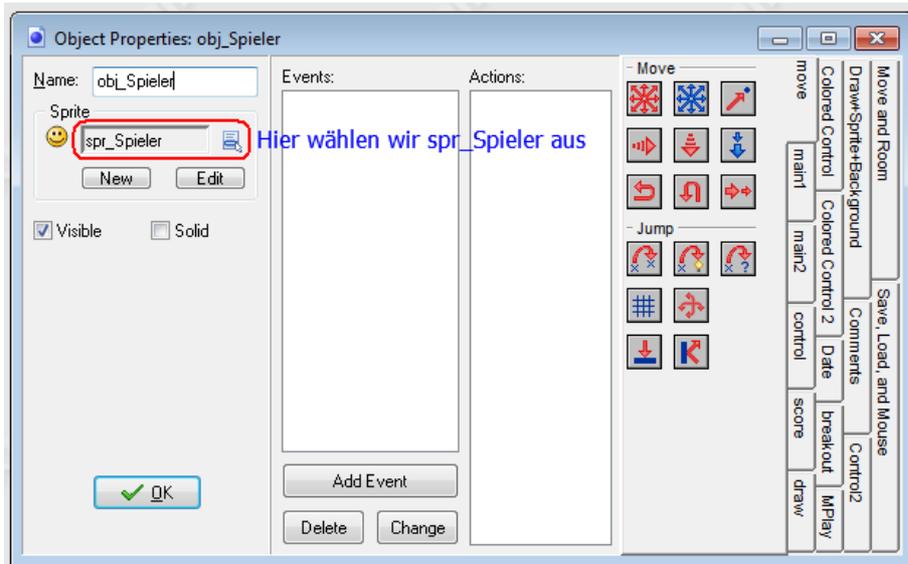
Das erste Sprite ist unser Spieler wir nennen ihn spr_Spieler, dazu geben wir oben in die Textbox wo Name steht den spr_Spieler ein. Nun haben wir den Sprite unseres Spielers erstellt, jetzt machen wir das Ziel, das spr_Ziel heißt und eine Wand die spr_Wand heißt. Für das erste Tutorial sind wir mit den Sprites fertig.

Der Spieler

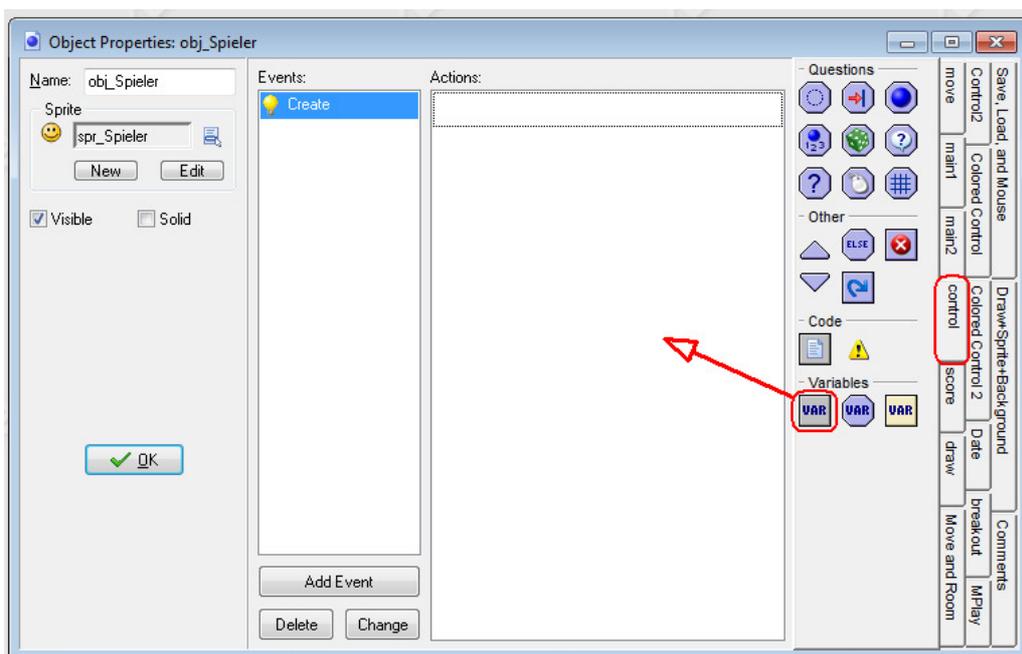
Jetzt erstellen wir das Objekt Spieler, dazu klicken wir auf den Blauen Ball:



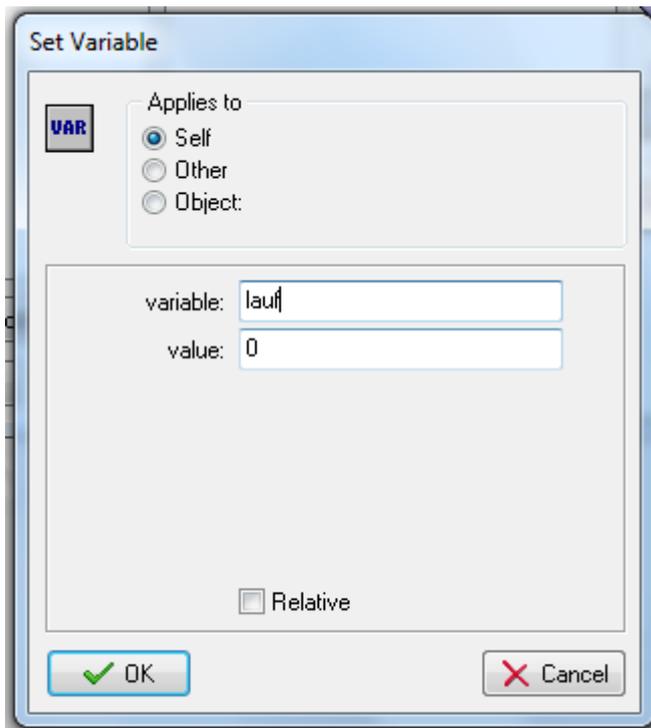
Das Objekt nennen wir obj_Spieler und geben ihm den Sprite spr_Spieler:



Und los geht's mit dem Programmieren. Als erstes wollen wir programmieren das der Spieler auf Tastendruck läuft, aber nur auf dem Raster. Wir drücken auf „Add Event“ und erstellen ein Create Event(Das mit der Glühbirne), alles was jetzt in dem Create Event steht wird ausgeführt wenn das Objekt „erstellt“ wird, also immer am Anfang von dem „Objekt“. Wir erstellen eine Variable indem wir rechts bei diesen Tabs auf Control Klicken und das Graue Bild wo VAR drauf steht in die Actions Box ziehen.

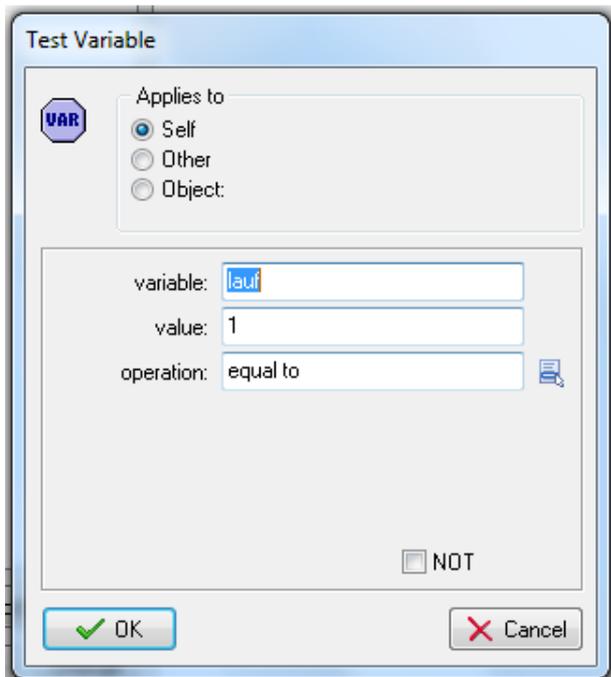


Dann erscheint ein Neues Fenster indem wir bei variable: „lauf“ eintragen und bei value: „0“

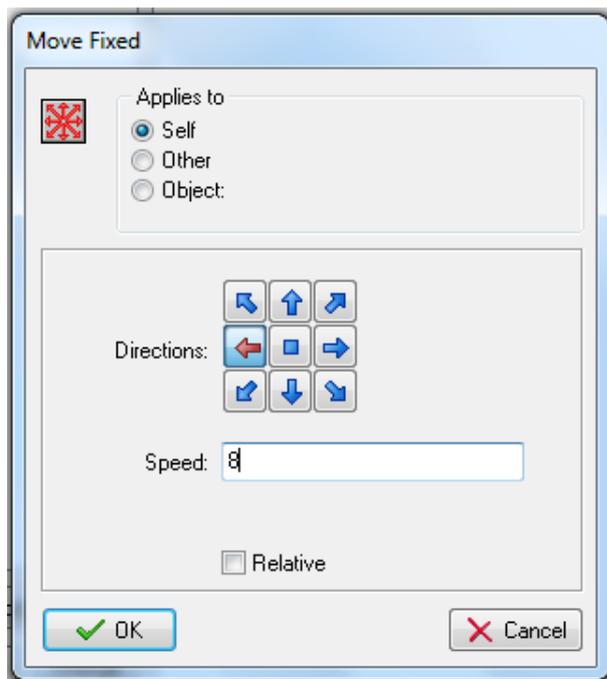


Dann auf OK drücken und wir haben unsere erste Variable „lauf“ gemacht. Variablen macht man im Create Event, Variablen speichern Werte die man auslesen und verändern kann, bei uns ist es dann so, dass wenn man eine Taste drückt der Wert lauf verändert wird, und das machen wir jetzt. Wir erstellen ein Keyboard <Left> Event, indem wir wieder das graue Bild mit VAR in die Actions Box ziehen, wir schreiben bei variable: „lauf“ und bei value: „1“ rein. Dieser Keyboard Event wird ausgeführt wenn man eine Taste drückt, bei uns ist es die „Linke Taste“, Dann wird die Variable lauf zu 1 gesetzt, denn immer oben steht die Variable und unten zu welchem Wert sie gesetzt werden soll. Jetzt soll der Spieler nach links laufen, wenn lauf = 1 ist.

Wir machen wieder ein Neues Event, das Step-Event. Das Step-Event wird jeden „Step“ ausgeführt, 30 steps sind 1 Sekunde. Dieser Event wird sogesagt immer ausgeführt. Dort ziehen wir von Control das 8-Eckige blaue VAR bild hinein(Das neben dem grauen VAR Bild). Diese Aktion testet verschiedene Sachen, bei uns soll sie Testen ob lauf = 1 ist. Dazu schreiben wir bei variable: „lauf“, bei value: „1“ und bei operation: „equal to“ rein.



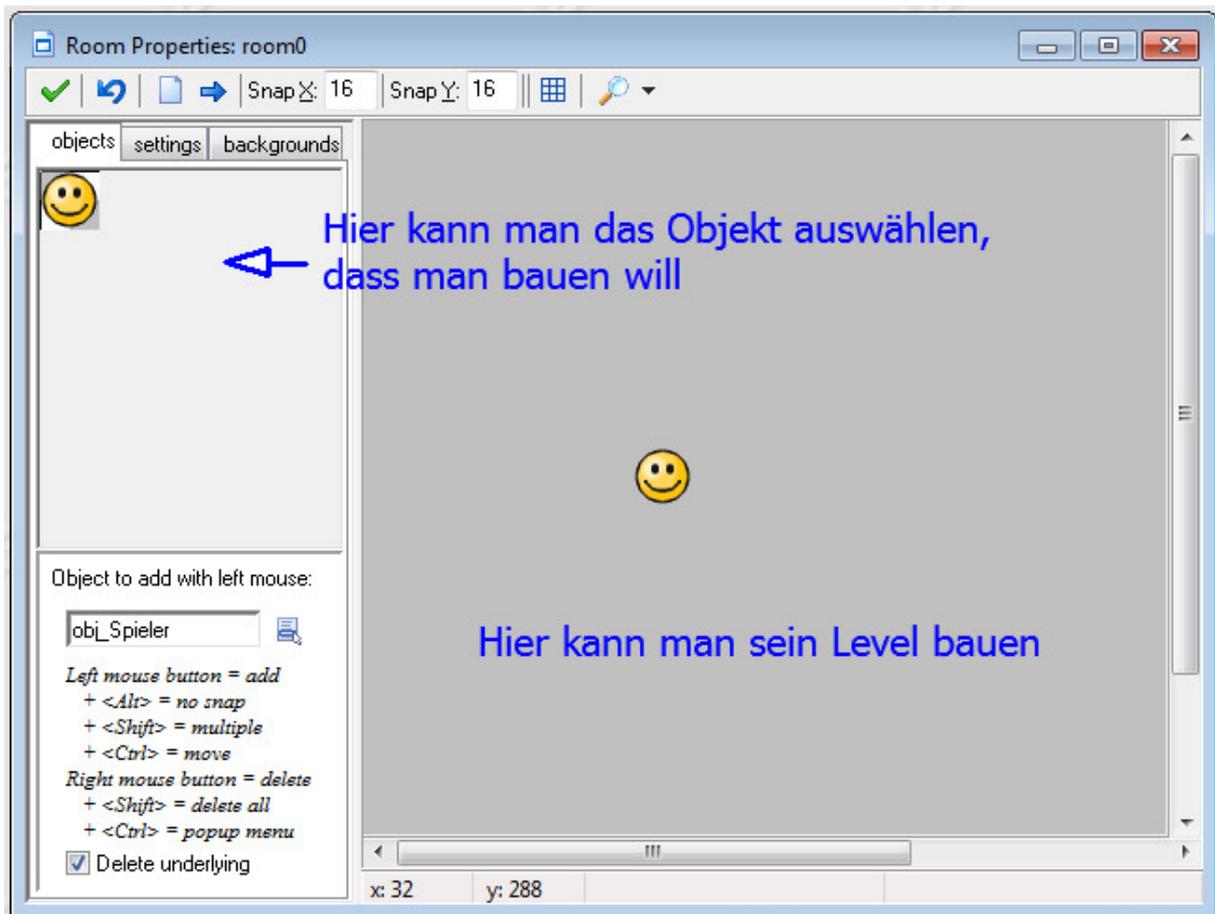
Dann gehen wir rechts bei den Tabs zu Move und ziehen das Feld ganz rechts oben mit den roten Pfeilen unter das Blaue Bild von vorhin. Dann drücken wir auf den linken Pfeil, weil der Spieler ja nach links laufen soll und unten die Geschwindigkeit, ich hab 8 eingegeben, das heißt wenn er diese Aktion ausführt springt der Spieler 8 Pixel nach links. Ihr könnt 1, 2, 4, 8, 16 und 32 bei der Geschwindigkeit eingeben, andere Zahlen würde ich nicht benutzen weil es nacher sonst Fehler mit dem Raster gibt.



Um das Spiel mal kurz zu testen machen wir einen Room.

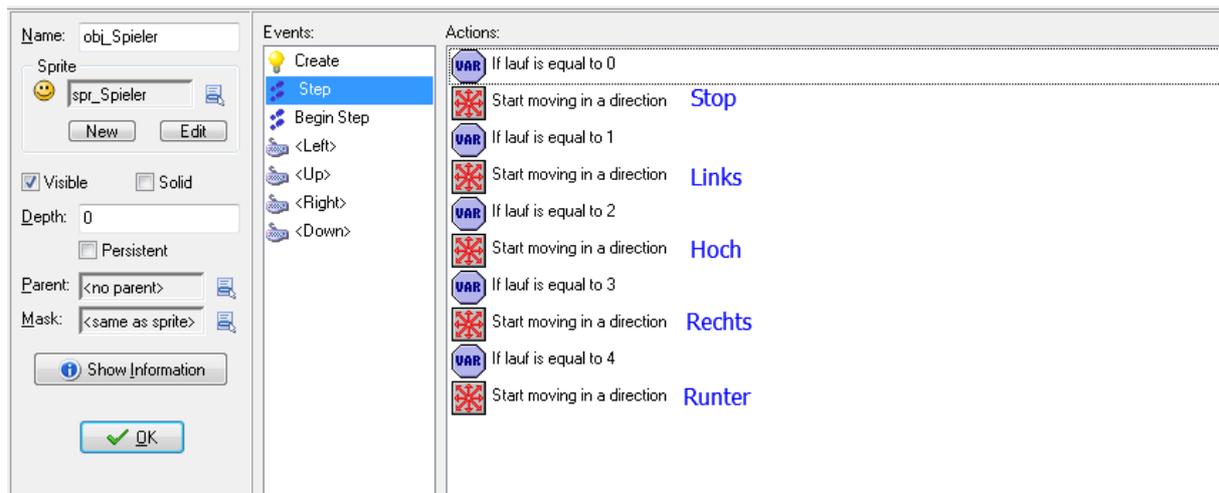


Dort erstellen wir unseren Spieler indem wir einfach in die Freie Fläche klicken, und somit ein Objekt bauen.



Jetzt können wir das Spiel auf dem Grünen Play Button oben oder auf F5 starten. Naja jetzt passiert halt noch nicht viel wenn man die Taste links Drückt geht der Spieler nach links. Jetzt müssen wir natürlich noch einstellen das er anhält wenn man die Taste losläßt. Dazu machen wir ein Begin Step Event, dieser Event wird so wie der Step Event ausgeführt, außer dass er immer **vor** allen Events dran kommt. Also wird erst dieser Event und dann die anderen ausgeführt. Dort setzen wir mit dem grauen VAR Bild die Variable lauf zu 0. Die Reihenfolge der Events ist jetzt so: Begin Step, Left Key, Step Event. Jetzt wird lauf durch den Begin Step immer jeden Durchgang am Anfang zu 0 gesetzt, wenn dann aber die Linke Taste gedrückt wird, wird lauf danach zu 1 gesetzt und dann kommt der step Event wo der Spieler dann nach Links läuft. Aber wir müssen noch einstellen das der Spieler bei 0 anhält, dazu gehen wir in den Step Event und testen ob die Variable lauf = 0 ist (mit dem Blauen VAR Bild), darunter machen wir wieder Moved Fixed drunter und drücken einfach nur auf den Mittleren Stop Button.

Nun stellen wir noch die anderen Tasten ein, wir machen ein Keyboard <Up> Event, ein Keyboard <Right> Event und ein Keyboard <Down> Event. Bei <Up> setzen wir lauf zu 2, bei <Right> zu 3 und bei <Down> zu 4. Im Step Event testen wir ob die Variable lauf = 2 ist, wenn ja machen wir Moved Fixed rein stellen die Geschwindigkeit ein und drücken den Button nach oben. Darunter testen wir ob lauf = 3 ist, wenn ja soll der spieler nach rechts laufen (wieder mit Moved Fixed) und wenn lauf = 4 ist soll er nach unten laufen.



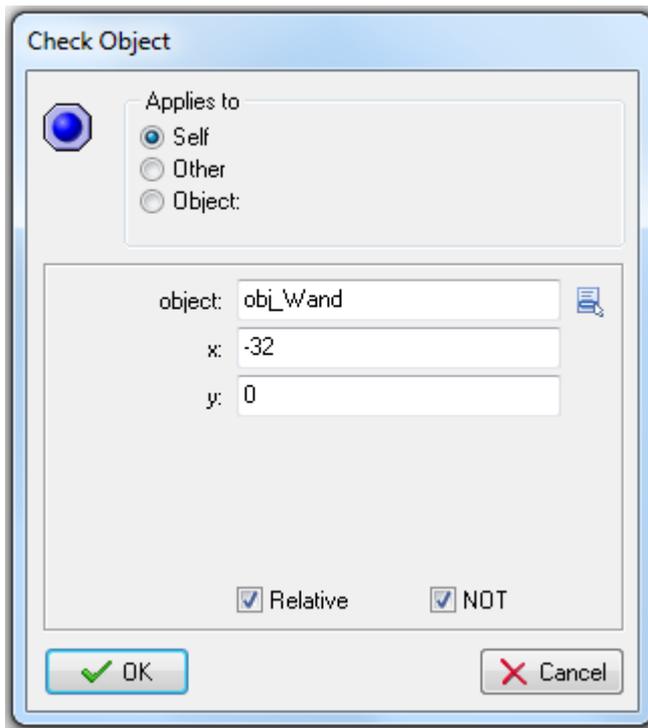
Wenn wir jetzt Testen kann unser Spieler laufen :D

Jetzt soll der Spieler aber noch nach einem Raster laufen, also dass er immer von Feld zu Feld geht. Wir gehen zum <Left> Event und ziehen „Check Grid“  von Control **über** das VAR Bild, wir geben bei snap hor und snap vert 32 ein. Das machen wir auch bei den Anderen **Pfeiltasten** und bei **Begin Step**. Dazu könnt ihr auch einfach auf die Aktion rechtsklick drücken Copy auswählen und bei den anderen Events einfach auf das VAR Bild Rechtsklick und Paste Drücken (Aber zum Üben könnt ihr auch jedesmal nochmal schreiben ;)). Diese Aktion testet ob der Spieler auf dem Raster steht. Jetzt müssen wir unseren Spieler aber noch auf das 32x32 Raster setzen (außer er ist das schon). Ihr geht zum Room stellt oben bei Snap X und Snap Y 32 ein und setzt den Spieler noch mal neu in den Raum (Snap X und Y sind das Raster). Der Spieler startet jetzt auf dem Raster dann kann man eine Taste drücken und der Spieler läuft zum nächsten Feld, man kann dem Spieler nur Befehle geben wenn er auf dem 32x32 raster steht. Ihr könnt es ja jetzt testen.

Die Wand

Damit der Spieler jetzt nicht nur Sinnlos rumläuft kommen noch Wände ins Spiel. Wir erstellen ein neues objekt, nennen es „obj_Wand“ und geben ihm den jeweiligen Sprite. Und schon ist das wand Objekt fertig wir können auf OK drücken. Und regelmäßig speichern nicht vergessen ;).

Der Spieler soll wenn eine Taste gedrückt wird erst schauen ob in der entsprechenden Richtung eine Wand steht, wenn ja bleibt er stehen, wenn nein läuft er einfach ganz normal weiter. Dazu gehen wir zum <Left> Event suchen das „Check Object“ Bild  bei Control und ziehen es mal wieder über das VAR Bild bei den Aktionen. Wir geben bei object obj_Wand ein, bei x -32, bei y 0, bei Relative und NOT machen wir ein Häkchen hin.



Diese Aktion testet ob an einer bestimmten Position ein Objekt ist, bei uns soll sie Testen ob 32 Pixel (Weil so groß das Raster ist) links neben dem Spieler **keine** Wand ist. Wir wollen ja machen wenn **keine** Wand dort ist kann er laufen. Das machen wir durch das Häkchen bei NOT. Relative heißt hier ob die Zahl bei x und y zu den Koordinaten von dem Spieler dazugezählt werden sollen, also sozusagen wenn der Spieler auf der x-Position 100 ist, dann wird hier getestet ob auf der position 68 eine Wand ist weil: $100 - 32 = 68$. Bei y steht nichts weil wir ja nur nach links laufen wollen und dazu wird dann nur die x-Position geändert. Jetzt testet der Spieler erst ob er im Raster steht und dann ob Links eine Wand ist, dann läuft er. Jetzt noch für die anderen Tasten: Wir machen wieder überall „Check Object“ mit den gleichen Einstellungen rein außer bei x und y. Hier steht was für die Jeweiligen Tasten bei x und y eingetragen werden soll:

<Up> x: 0 y: -32

<Right> x: 32 y: 0

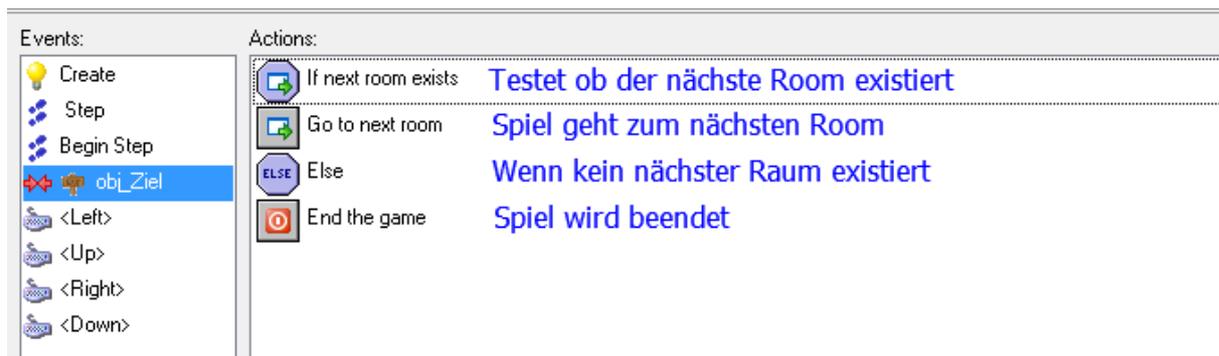
<Down> x: 0 y: 32

Die x und y Koordinaten sind jeweils anders weil der Spieler ja immer in andere Richtungen laufen will. Jetzt könnt ihr ein Level mit Wänden bauen und es spielen :D .

Das Ziel

Jetzt machen wir noch ein Ziel damit man von einem Level zum nächsten kommt. Wir erstellen ein neues Objekt obj_Ziel, wählen den jeweiligen Sprite aus und drücken auf OK. Der Spieler soll jetzt wenn er das Ziel berührt ins nächste Level kommen, dazu erstellen wir ein neues Event beim Spieler, das „Collision with obj_Ziel“ Event. Dieser Event wird dann

ausgeführt wenn sich die beiden Objekte (die Sprites davon) berühren. Wir wollen, das wenn der nächste Room existiert wir zum nächsten Room können. Dazu brauchen wir unter main1 unten bei den Rooms „Check Next“ . Hier müssen wir nichts eingeben denn da wird nur getestet ob der nächste Room existiert. Darunter machen wir „Next Room“  hier könnt ihr einen Effekt bei dem wechseln von dem Room auswählen. Jetzt wollen wir aber noch wenn der nächste Room nicht existiert wird das Spiel beendet. Wir gehen zu Control und ziehen „Else“  unten hin. „Else“ wird ausgeführt wenn die obere „Check“ Aktion nicht ausgeführt wird, also wenn sie nicht wahr ist (bei uns ist es „Check Next“). Jetzt nur noch unter main2 End Game ganz unten reinziehen.



So jetzt könnt ihr schon ein paar Sinnlose Level machen :D .

Hier nur nochmal eine kurze erklärung:

 Diese Aktionen führen immer etwas aus, wie z.b. Variablen verändern.

 Diese Aktionen testen immer irgendwas wenn sie stimmen wird die Aktion drunter ausgeführt

Das war mein Tutorial ich hoffe ihr habt es verstanden, wenn Fragen sind fragt einfach.