

Das Bodenfallenrätzel- Tutorial

für

RPG Maker MV

(Version 1.0 – 09.05.2018)

Für kommerzielle und nicht-kommerzielle
Verwendung im RPG Maker.

Jedoch bitte ich um Credits inkl. Webseitenlink:

Credits:

Game Alchemists
<http://gamealchemists.com>

Dieses Tutorial darf weitergegeben und vervielfältigt werden.
Eine Editierung oder Ausgabe ohne Credits bzw.
Änderung des Urhebernemens ist nicht gestattet.



COPYRIGHT BY GAME ALCHEMISTS

Vorwort

Das Tutorial ist für den *RPG Maker MV* geschrieben. Da der *RPG Maker VX Ace* noch keine Maussteuerung besitzt, kann das Tutorial leider nicht auf den *RPG Maker VX Ace* adaptiert werden.

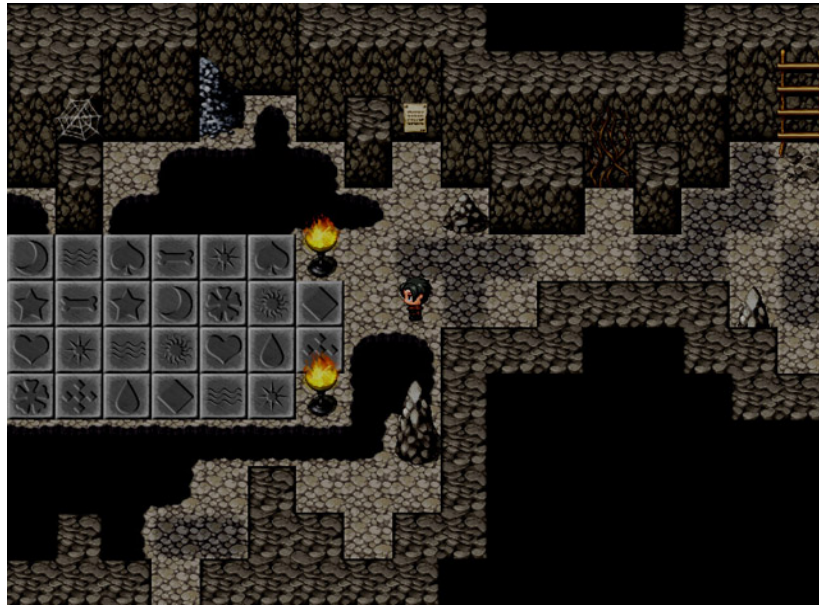
Weitere Tutorials und Fragen?

Habt ihr Interesse an weiteren Tutorials, werdet ihr hier sicher fündig: [**http://gamealchemists.com**](http://gamealchemists.com)

Habt ihr Fragen zum Tutorial? Dann meldet euch einfach im Forum: [**http://gamealchemists.com/forum**](http://gamealchemists.com/forum)

Ihr habt Ideen zu Minispielen und wisst nicht, wie ihr sie umsetzen sollt oder sucht ein exclusives Minispiel? Dann schreibt mir einfach eine E-Mail: [**http://gamealchemists.com/kontakt/**](http://gamealchemists.com/kontakt/)

Einleitung

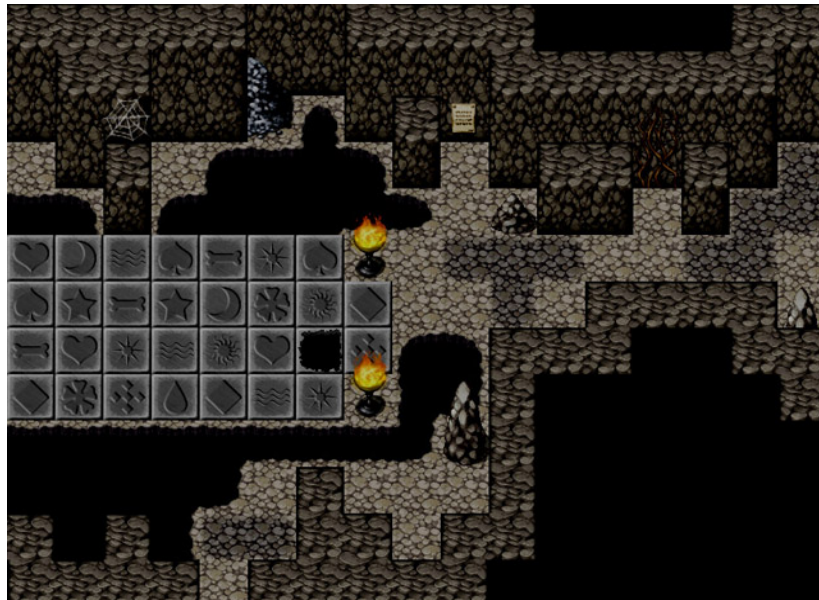


Kennen wir diesen kleinen Herrn nicht von irgendwo her? Eigentlich nein... er ist der neue Protagonist meiner Tutorial-Reihe für den *RPG Maker MV*.

Laurenz Jr. ist in einem Dungeon angekommen, doch was ist das? Merkwürdige Bodenplatten! Besonders vertrauenserrückend sehen dies nicht aus!

Was nun?

Probieren wir einfach einen Sprung aus und ...



Aha! Einige Bodentafeln scheinen nicht sicher zu sein! Bestimmt sind einige Symbole sicher und einige nicht. Laurenz Jr. muss also von einer richtigen Bodenplatte zur nächsten springen!

Um was geht es in diesem Tutorial?

In diesem Tutorial befassen wir uns mit einem Bodenfallenrätsel. Auf dem Boden sind mehrere Tafeln verteilt, einige werden begehrbar sein, andere werden den Protagonisten Unheil bringen. Des weiteren werde ich in diesem Tutorial viele grundlegende Maker-Befehle ausführlich erklären.

Wir benötigen

1. Gute Laune
3. Das Plug-in TDDP_MouseSystemEx von Galenmereth
3. Ein paar Events

Schwierigkeitsgrad: ★★☆☆☆

Das Plugin

Das Erste, was wir brauchen, ist das **MouseSystemEx** Plugin von **TDDP**.

Was ist ein Plug-in?

Ein Plug-in ist eine Ergänzung des *RPG Maker MV*. Er bietet Neuerungen und Funktionen, die der *RPG Maker MV* standardmäßig nicht mitbringen.

Woher bekomme ich das Plug-in?

Der *RPG Maker MV* kommt mit einer ganzen Reihe an Zusätzen daher, die der Anwender meistens gar nicht bemerkt bzw. kennt. Grund dafür ist, dass die Daten an einem ziemlich merkwürdigen Ort gespeichert werden, die normalerweise kein Anwender aufsucht.

Daher geht bitte in folgendes Verzeichnis:

➡ **c:\Programme(x86)\Steam\steamapps\common\RPG Maker MV\dlc\RPGmakerWeb_plugins\Galenmereth**

In diesem Ordner findet ihr das Plugin: **TDDP_MouseSystemEx**

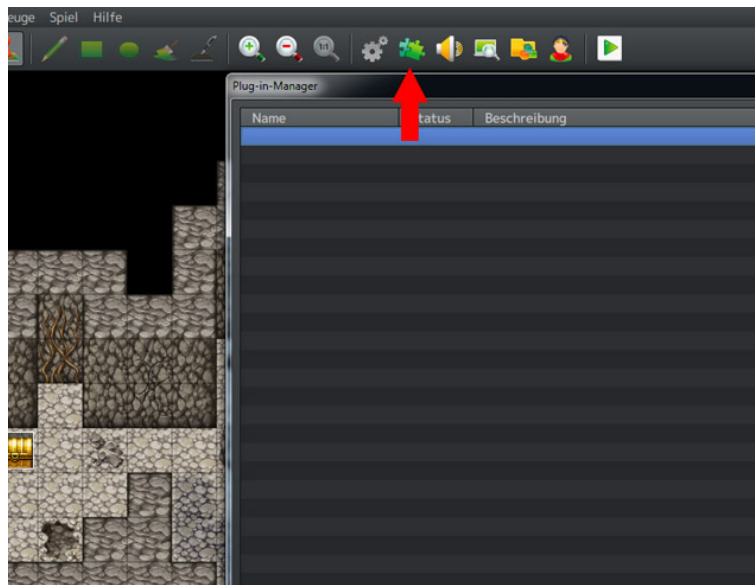
Es ist eine Java Script File die ihr nun kopiert.

Geht nun in euer RPG Maker MV Projekt, normalerweise sollte dies zu finden sein unter:

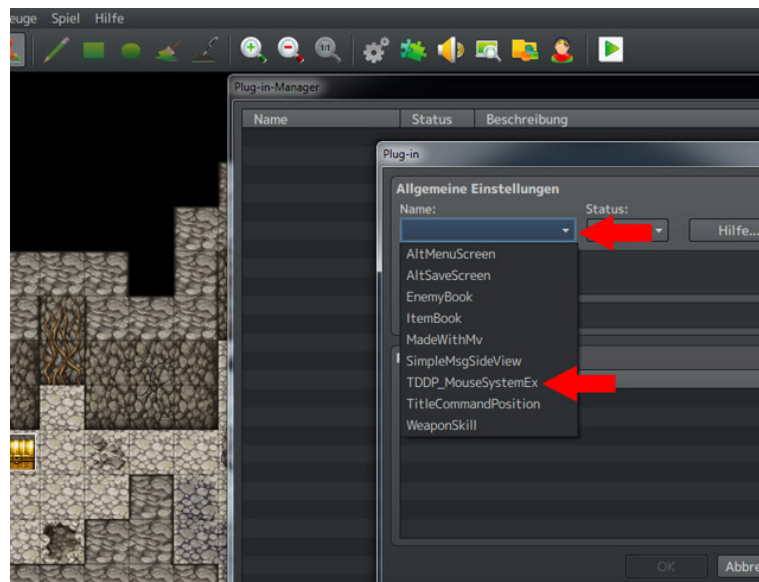
➡ **Bibliotheken\Dokumente\Eigene Dokumente\Games\NAME EURES PROJEKTES**

In eurem Projekt geht ihr in den Ordner **js**, in diesem Ordner wiederum geht ihr in den Ordner **plugins**. In diesen fügt ihr nun das Plug-in ein.

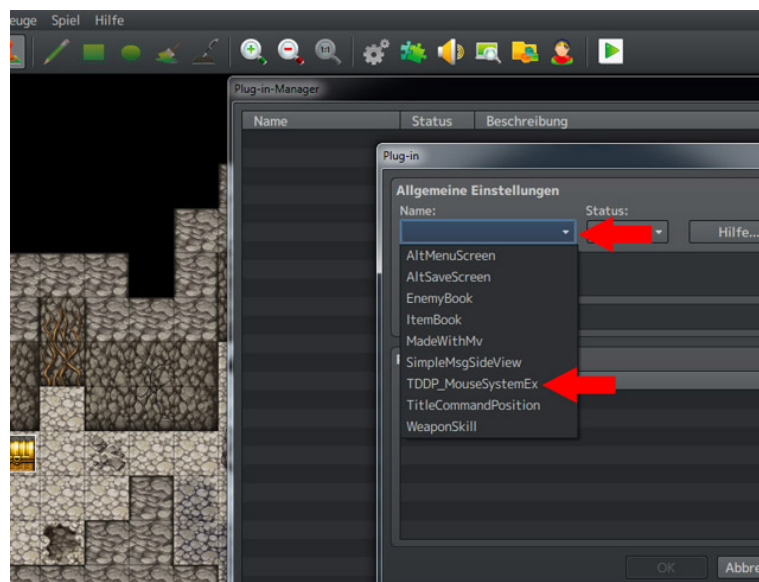
Öffnet nun den RPG Maker MV und startet euer Projekt und öffnet den Plug-in-Manager:



Klickt im neuen Menü in die obere Zeile und öffne damit das neue Auswahlfenster. Unter Name sucht ihr das neue Plug-in:



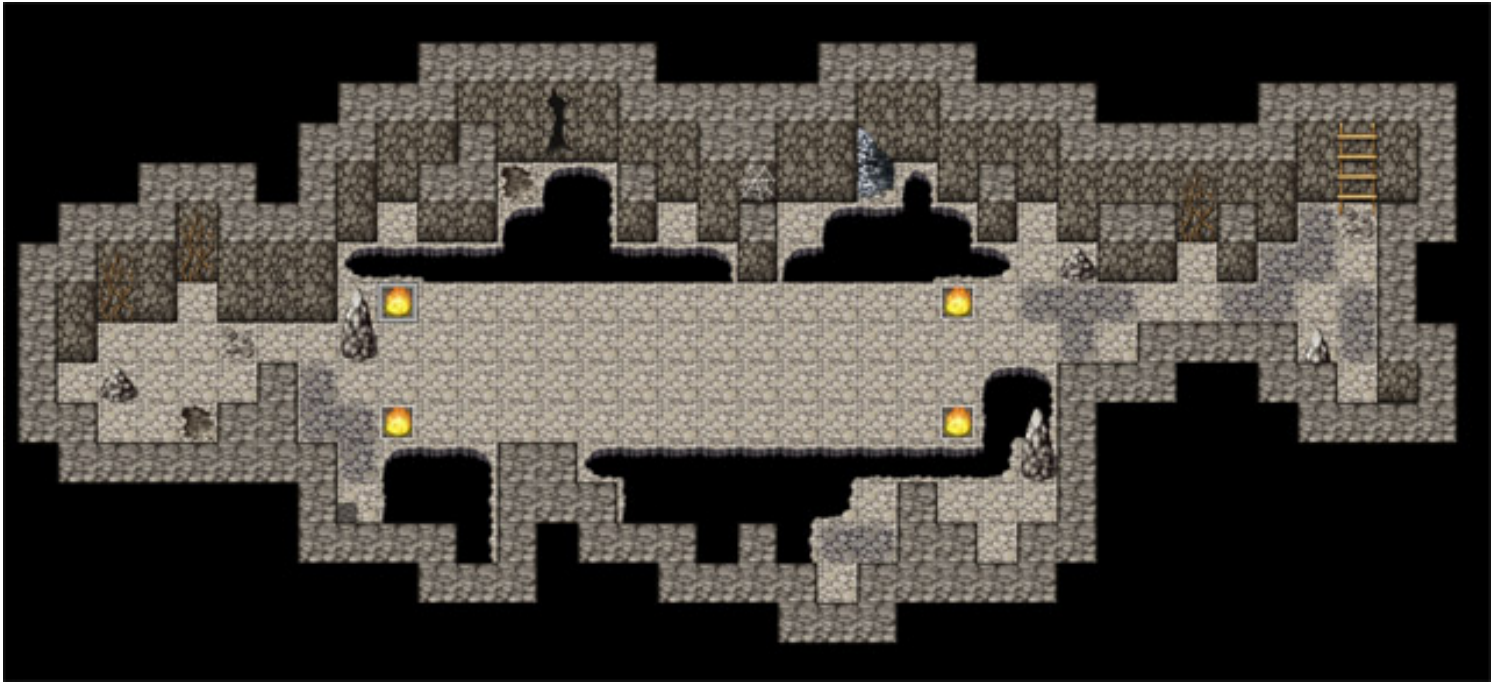
Wählt das **Plug-in** aus und betätigt zwei Mal die **OK-Taste**, um den **Plug-in-Manager** zu schließen. Weitere Einstellungen müsst ihr hier nicht machen.



Damit wäre nun euer Projekt mit dem neuen **Plug-in** ausgestattet.

Die Karte und das erste Event

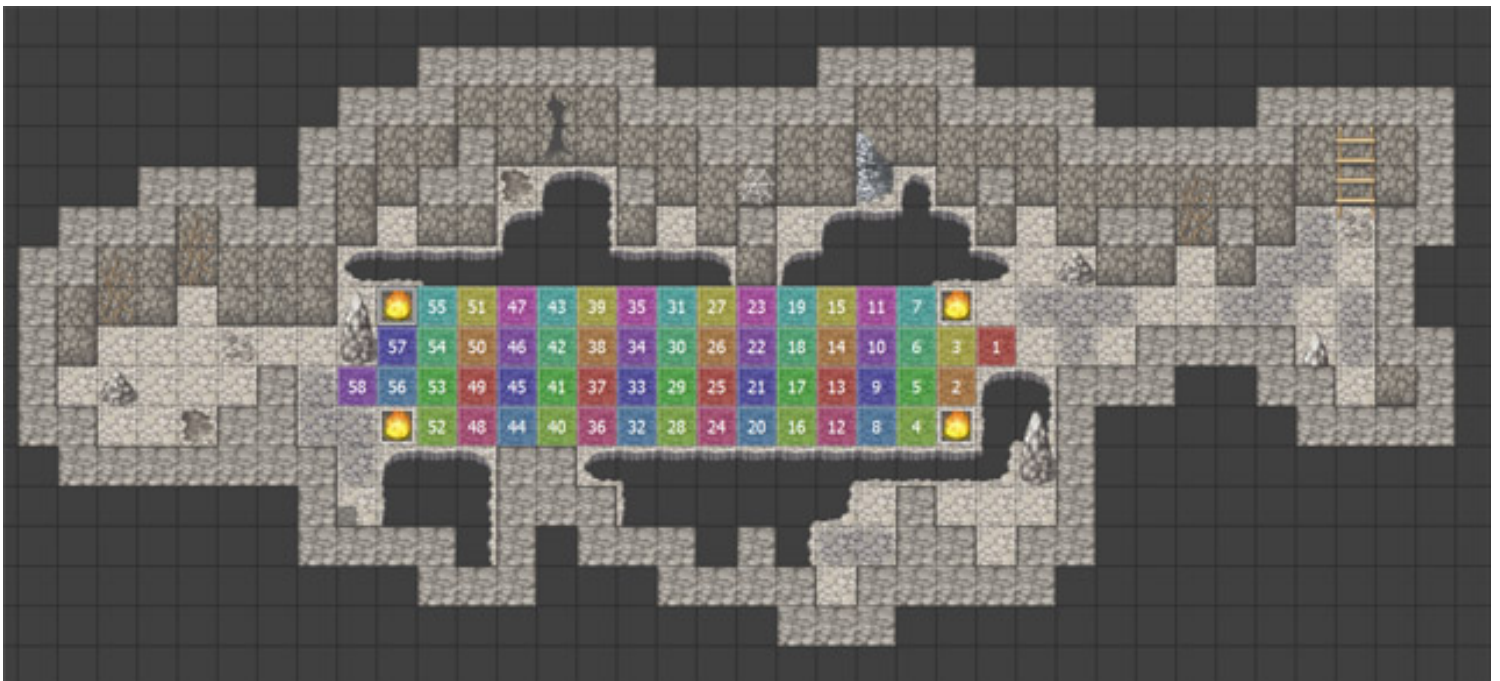
Das nächste, was wir benötigen ist eine passende Map:



Wie ihr seht ist meine Karte recht einfach gehalten, eine Steinbrücke, ein Eingang und ein kleiner Bereich, links wo später die Belohnung liegt. Zwischen den Fackeln werden die Bodenplatten liegen.

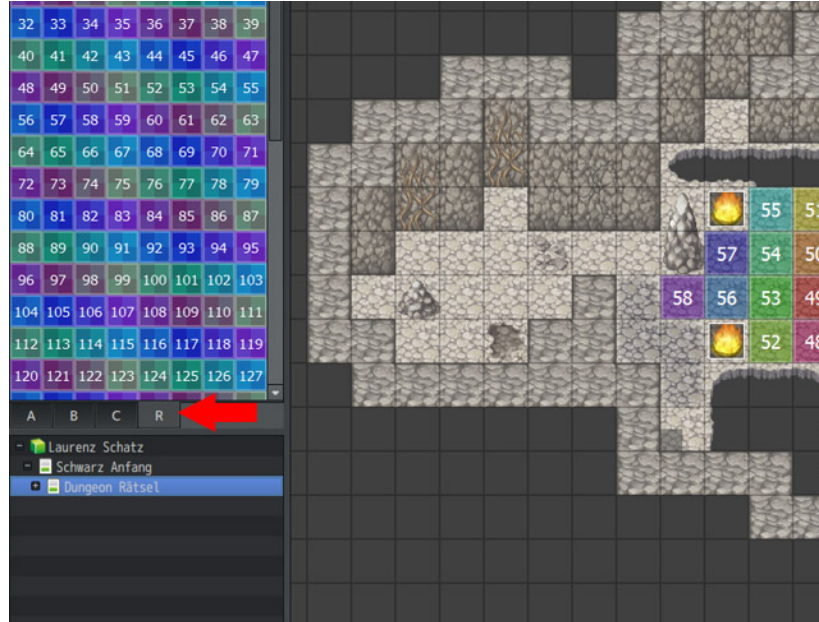
❗ Wichtig ist hier, dass ihr euch eine Karte erschafft, bei der es keinem Weg gibt, mit der man die Bodenplatten umgehen kann.

Wenn die Map fertig ist, muss die Karte im Bereich der zukünftigen Bodenplatten mit sogenannten **Regions IDs** versehen werden:





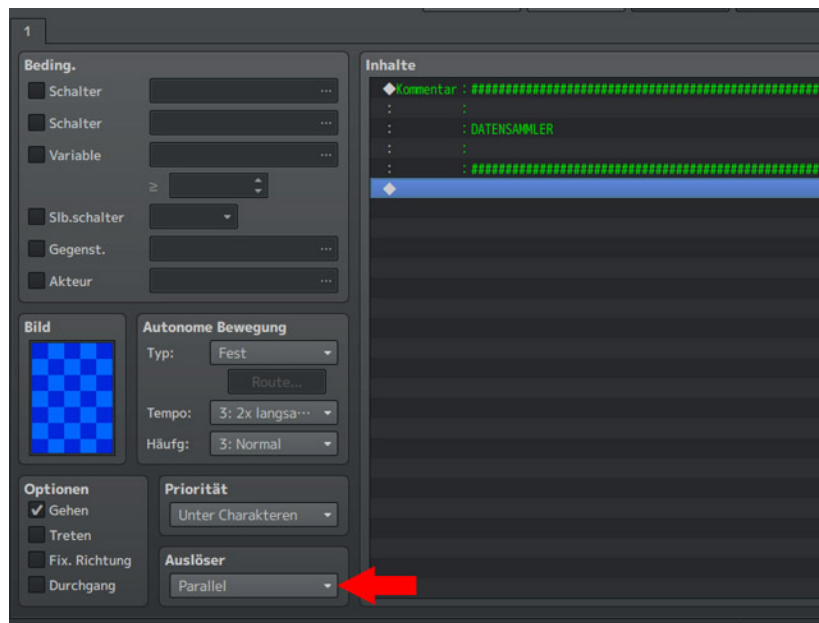
We



Dc

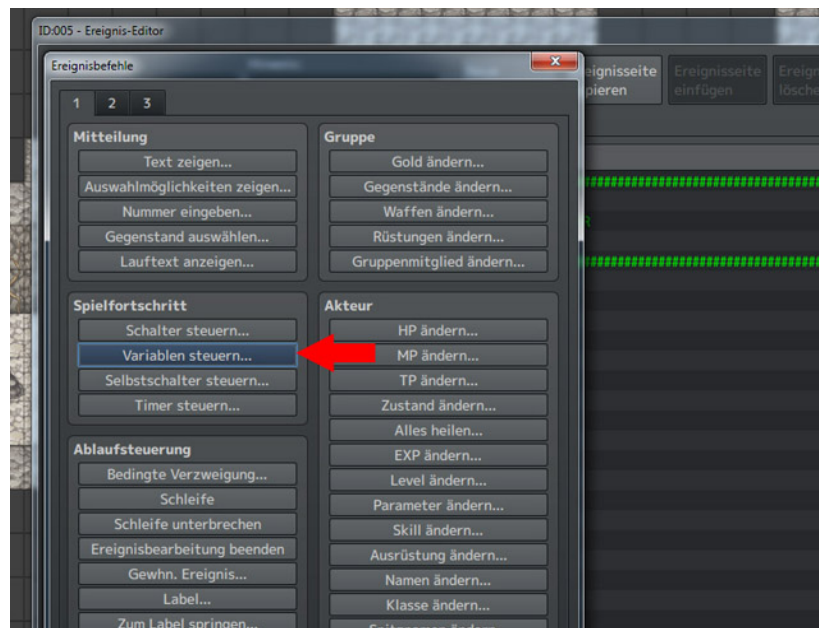
W

Es

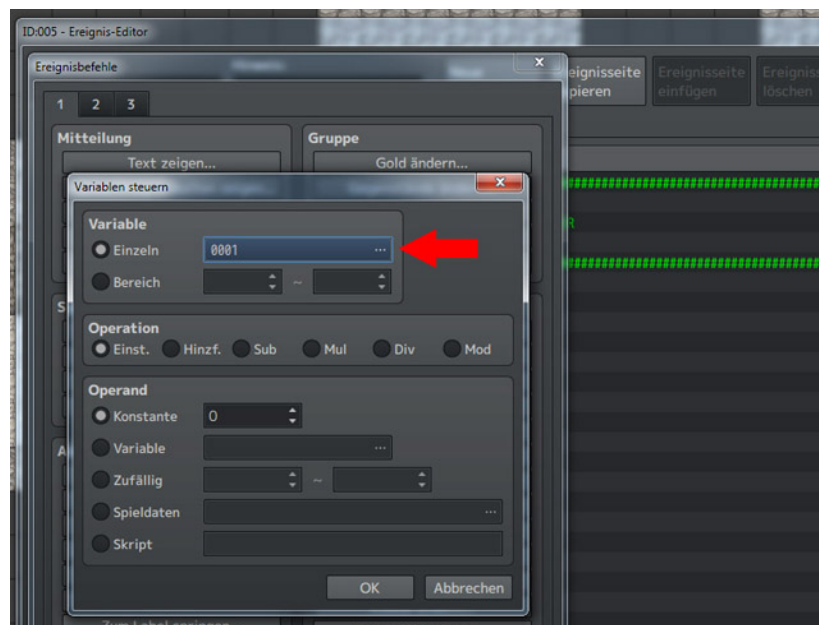


Doppelklickt nun auf die oberste Zeile des Inhalts oder klickt im **Inhalt** auf die **rechte Maustaste** und dort auf **Neu**.

Im neu erschienen **Ereignisbefehle-Fenster** geht ihr auf **Variablen steuern...** :

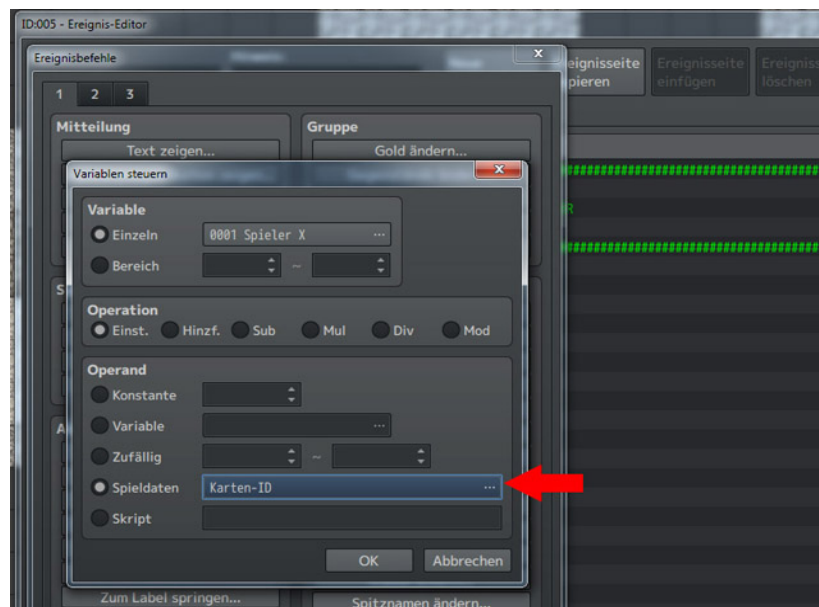


In der **Variablen Steuerung** geht ihr ganz oben auf **Einzeln**:

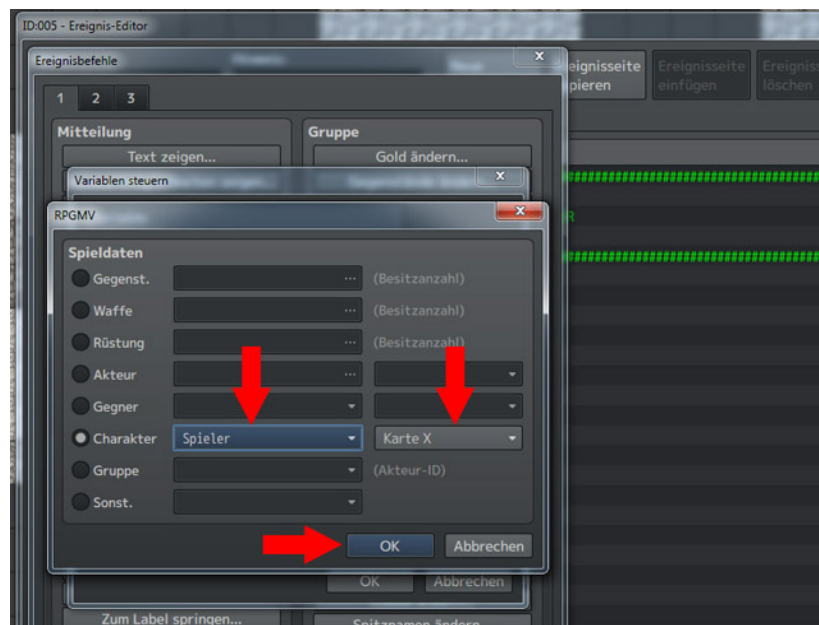


➡ Im neuen Fenster tragt ihr folgenden Namen ein: **Spieler X**
Abschließend übernimmt ihr das gesamte durch Klicken auf **OK**:

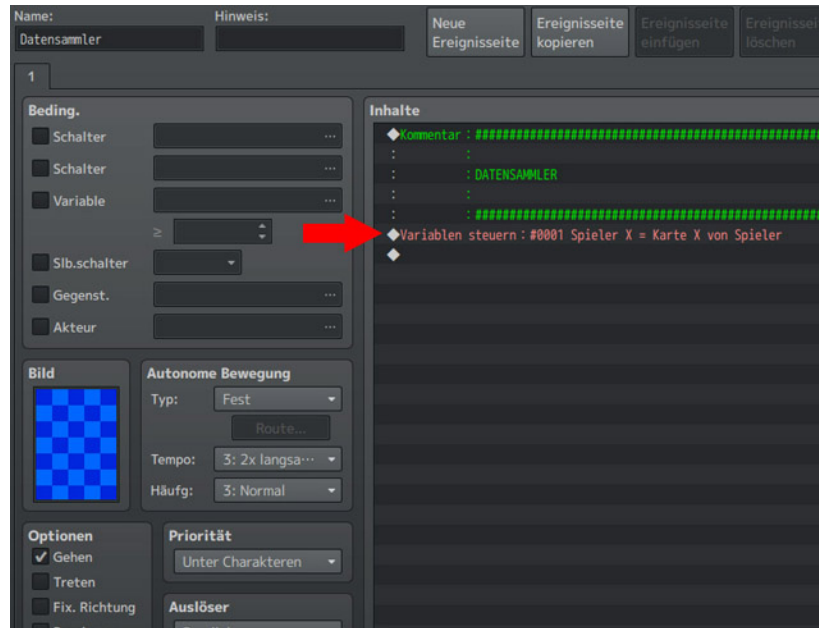
Im nächsten Schritt werden die Spielerdaten in der Variablen Steuerung ausgelesen:



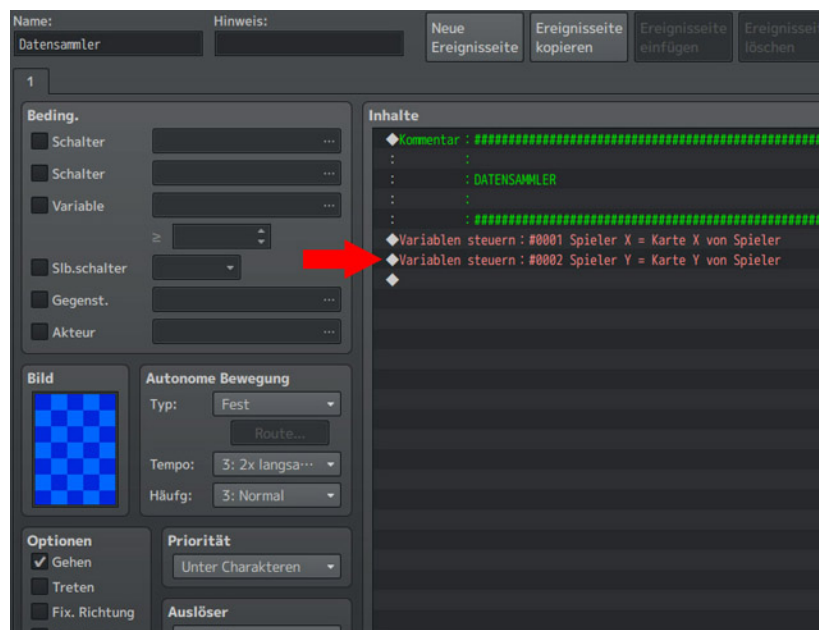
Tragt unter **Charakter** den **Spieler** ein und der Auslesefaktor soll **Karte X** sein. Hiermit können wir die **X-Koordinate des Spielers** auslesen. Anschließend gehen wir auf **OK**:



Bei euch müsste es nun so aussehen:

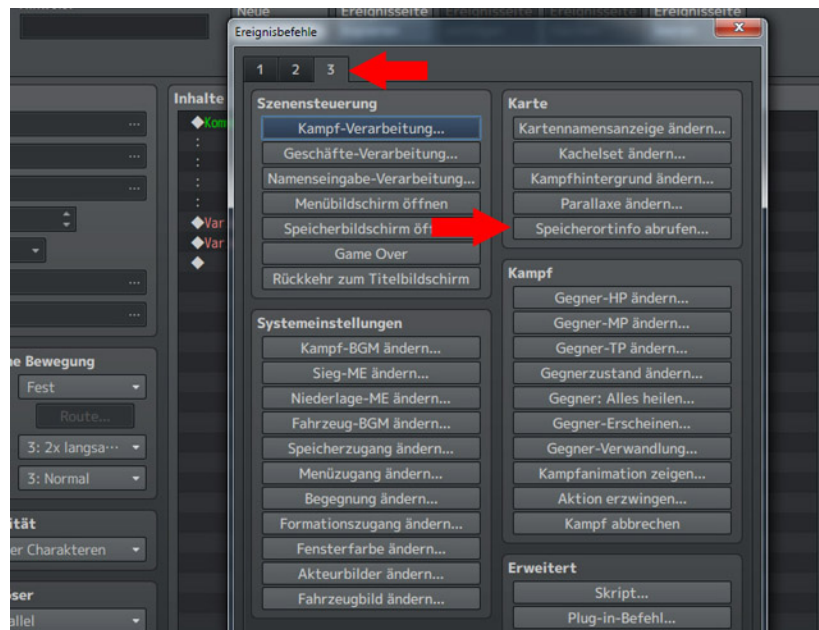


Dasselbe macht ihr nun mit der **Y-Koordinate**:



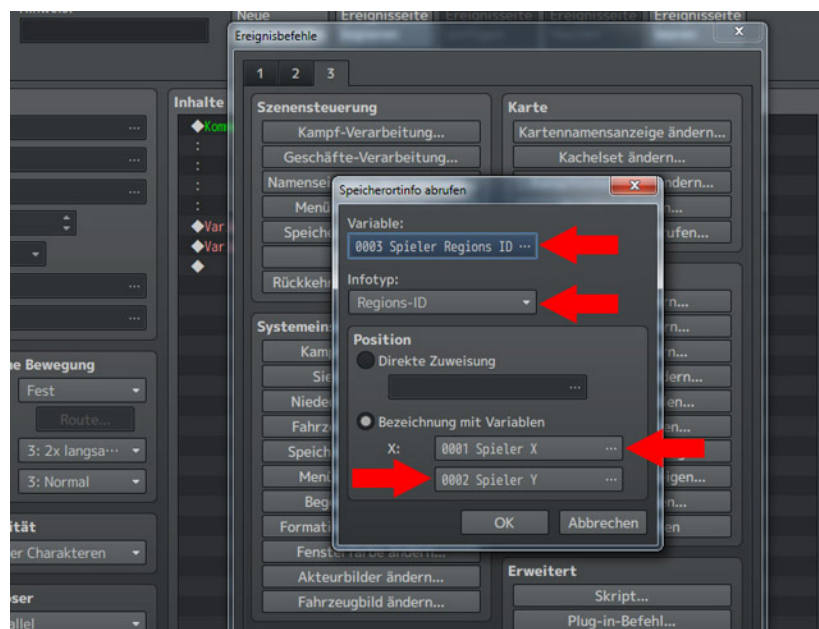
Fertig? Dann werden wir nun aus der **X und Y- Koordinate** die **Regions ID** berechnen, in dem wir wieder ins **Ereignisbefehle-Fenster** gehen.

Dieses Mal suchen wir den **dritten Reiter** auf und klicken in diesem Bereich auf **Speicherortinfo aufrufen...**:



In der **Speicherortinfo** haben wir nun gleich mehrere Dinge einzustellen:

- ➡ Erzeugt eine neue Variable mit dem Namen **Spieler Regions ID** und trägt diese unter **Variable** ein
- ➡ Stellt den **Infotyp** auf **Regions-ID**
- ➡ **Bezeichnung mit Variablen X** stellt ihr auf unsere Variable **Spieler X**
- ➡ **Bezeichnung mit Variablen Y** stellt ihr auf unsere Variable **Spieler Y**



Schließt alles wieder durch die Bestätigung mit **OK**.

Hiermit wird nun dauerhaft abgefragt, auf welchem **Regions ID** Feld sich der Spieler befindet. Damit der Maker jedoch auch einmal eine kleine „Verschnaufpause“ bei der Abfrage gegönnt wird, erstellen wir noch ein **Warte-Frame**, welches **1/60 einer Sekunde** anhält, bevor es die Daten wieder aktualisiert.

Inhalte

- ◆ Kommentar : #####
- : :
- : DATENSAMMLER
- : :
- : #####
- ◆ Karten : 1 Frame
- ◆ Variablen steuern : #0001 Spieler X = Karte X von Spieler
- ◆ Variablen steuern : #0002 Spieler Y = Karte Y von Spieler
- ◆ Speicherortinfo abrufen : Spieler Regions ID, Regions-ID, ({Spieler X},{Spieler Y})
- ◆

Autonome Bewegung

Typ: Fest
Route...

Tempo: 3: 2x langsamer

Häufig: 3: Normal

Priorität
Unter Charakteren

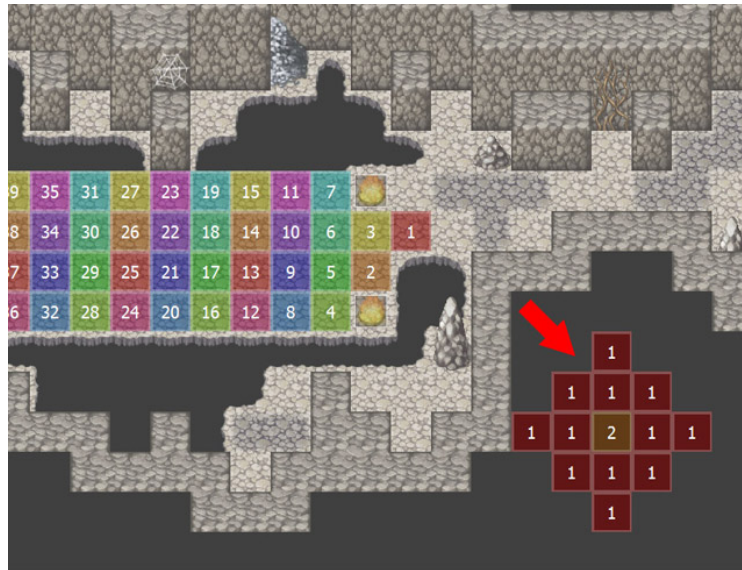
Auslöser
Parallel

OK Abbrechen

Die Steuerevents

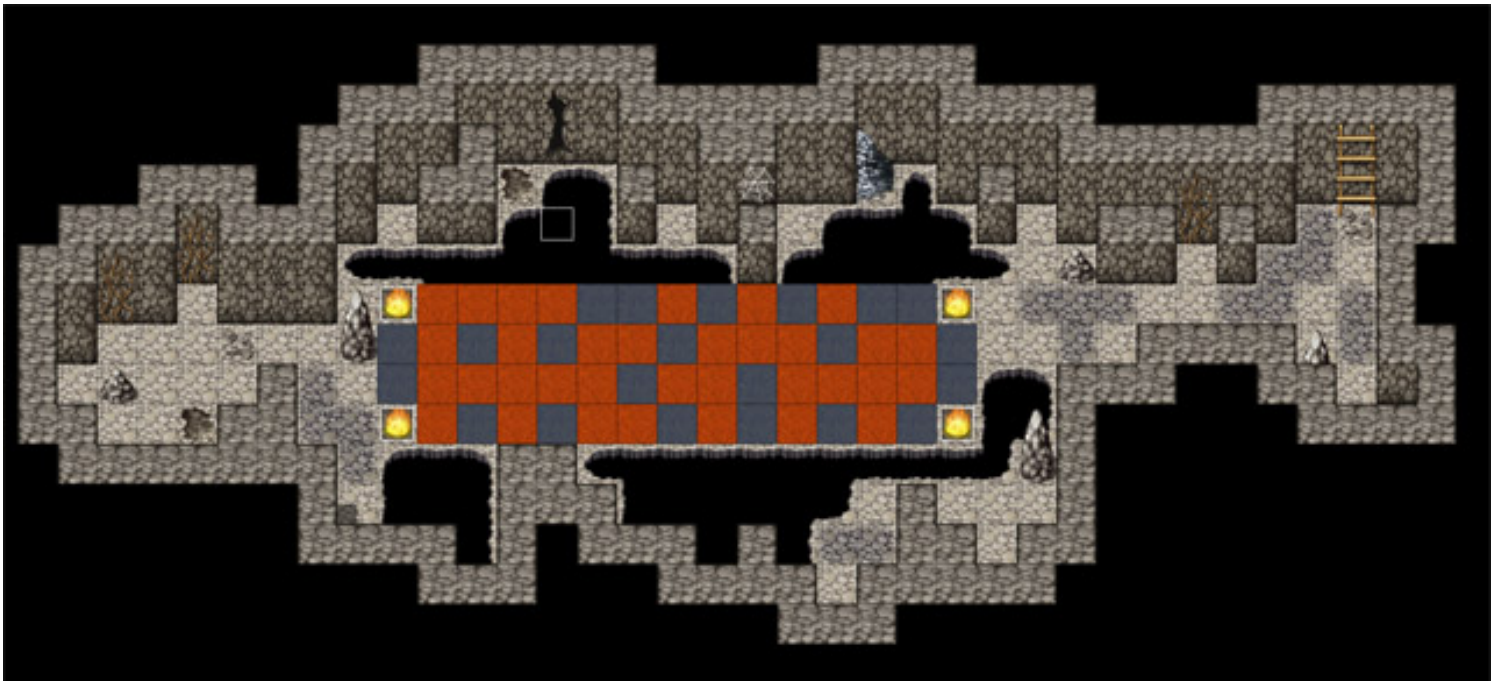
In diesem des Tutorial werden wir nun festlegen, wie sich der Protagonist bewegen kann.

Hier einmal eine Übersicht, welchen Bewegungsbereich der Spieler haben wird:



Da wir nun wissen, in welchem Bereich sich der Protagonist bewegen kann, legen wir nun fest, welches der Felder der Protagonist später betreten kann und welches besser nicht.

Ich habe mir für die Markierung aus dem **Tileset** den roten und blauen Teppich genommen und Markierungen auf den Boden gebracht. Alles was nun rot zu sehen ist, wird später tödlich sein, alles was blau ist, wird später begehbar sein:



An dieser Stelle wird es etwas komplizierter, lasst euch also nicht verwirren! Wir erschaffen unser zuallererst ein Event auf **Regions ID 2**.

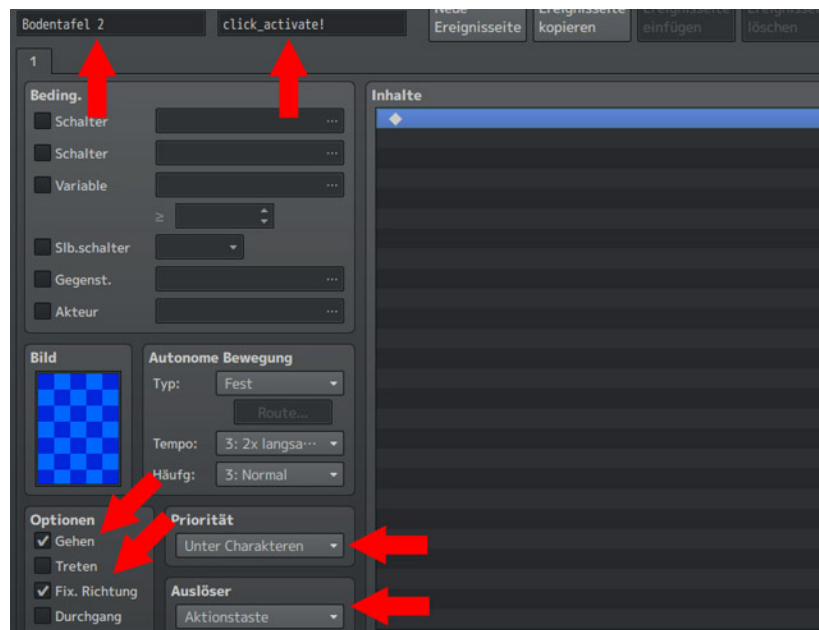
❗ Wichtig: Auf **Regions ID 1** und **58** kommen **keine Events**, sie dienen nur zur Kontrolle wo der Spieler ist.

Wir erstellen uns also ein Event auf **Regions ID 2** und nennen es **Bodentafel 2**. (habe ich so benannt um die gleiche Zahl wie bei den **Regions IDs** zu haben)

Dieses Event hat nun folgende Eigenschaften:

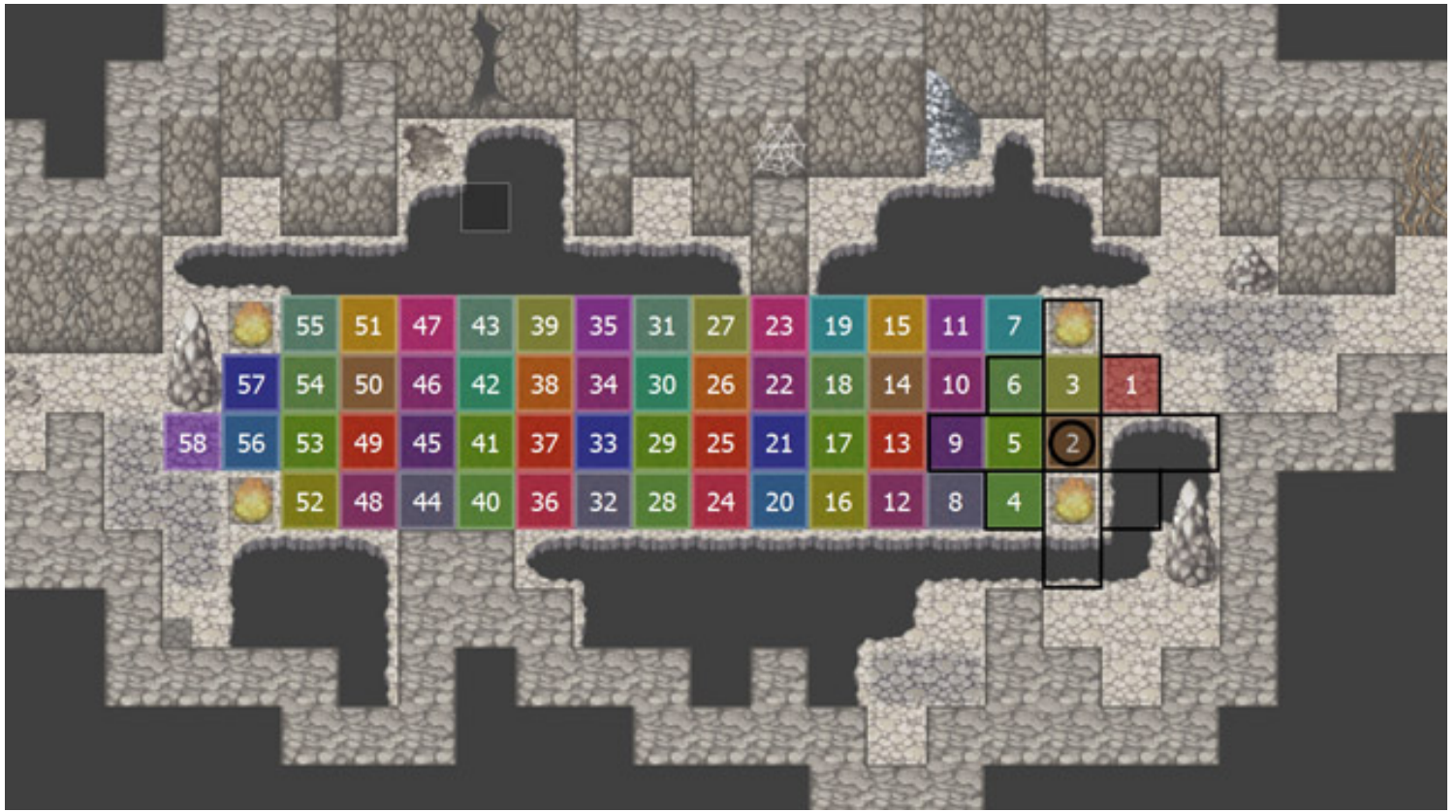
- ➡ Unter Hinweis, rechts neben dem Eventnamen schreibt ihr: **click_activate!**
- ➡ Optionen: Gehen und Richtung Fixieren
- ➡ Priorität: Unter Charakter
- ➡ Auslöser: Aktionstaste

⚠ Ihr fragt euch nun sicher, was wir da in den Hinweis geschrieben haben. Ganz einfach, normalerweise funktioniert ein Event im *RPG Maker MV* so: Man klickt ein Event an, der Protagonist läuft zu diesem Event, erst dann wird es ausgeführt. Dies wollen wir aber gerade nicht! Unsere Events sollen aus der **Distanz aktiviert werden**, dafür haben wir als erstes das **MouseSystemEx Plugin** eingebaut, dieser Befehl im Hinweis ist ein Teil dieses **Plug-in**.



Bisher ja ganz einfach, nun beginnt aber die Denksportaufgabe. Wenn wir auf ein Event klicken, soll der Spieler aus einem bestimmten Bereich auf dieses Event geholt werden. Diesen Bereich in Form eines Sterns haben wir weiter oben im ersten Bild festgelegt. Deshalb müssen wir nun im Event **Bodentafel 2** abfragen, ob der Spieler innerhalb des Einzugsbereiches dieses Events ist.

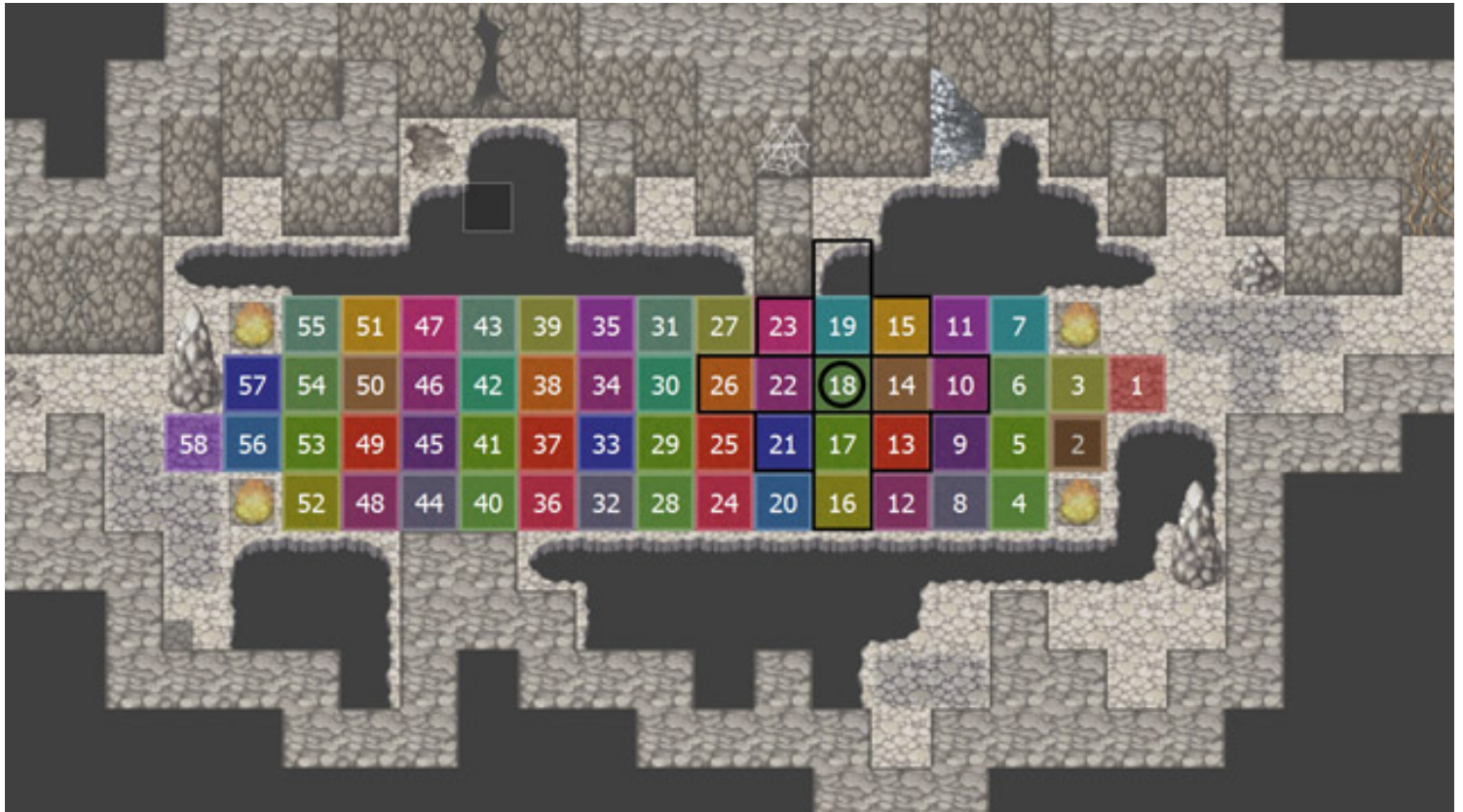
Um es euch etwas verständlicher zu machen, hier einmal der **Einzugsbereich** von **Event 2**:



Wie ihr seht, liegt **Event 2** in der Mitte, der Einzugsbereich ist durch die **schwarzen Außenkanten** ersichtlich.

Würde der Protagonist nun entweder auf **Regions ID 1, 3, 4, 5, 6** oder **9** stehen, wäre er im **Einzugsbereich der Regions ID 2**. Klickt man nun auf das Event, welches auf **Regions ID 2** liegt und steht der Spieler auf den besagten **Regions IDs**, würde der Spieler nun auf das Feld der **Regions ID 2** springen.

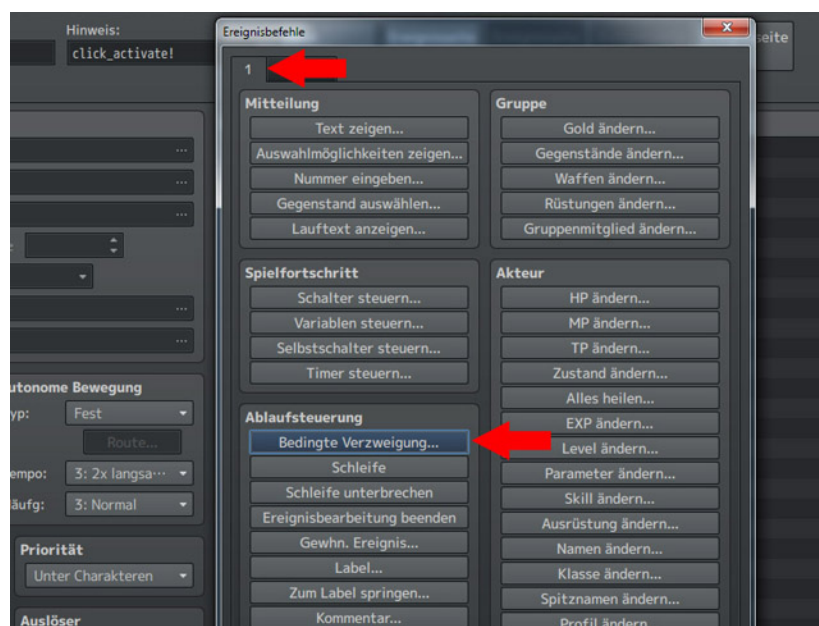
Noch ein Beispiel:



Hier wäre nun **Regions ID 18** der Mittelpunkt. Klickt der Protagonist auf das Event, welches auf **Regions ID 18** liegt, würde er auf dieses Feld springen, wenn er auf den **Regions IDs: 10, 13, 14, 15, 16, 17, 19, 21, 22, 23** oder **26** steht.

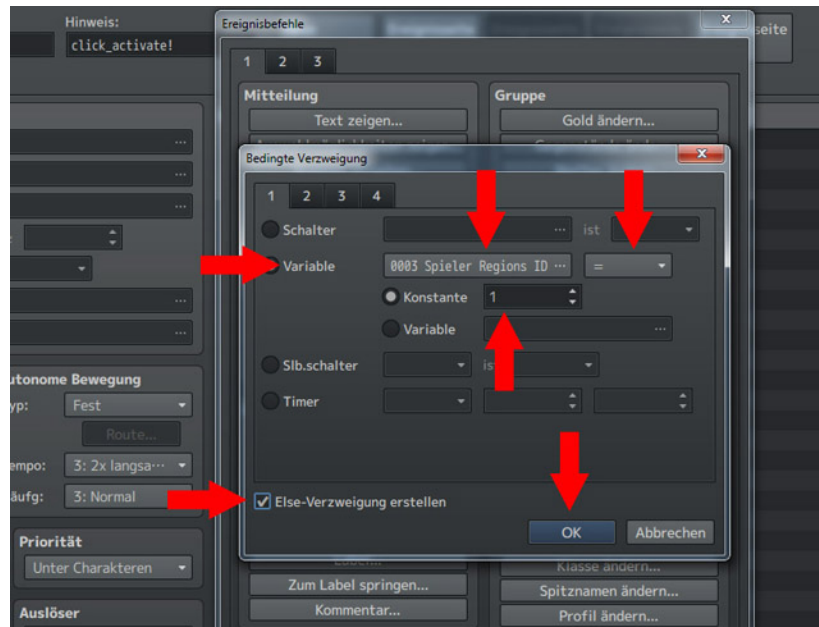
Also müssen wir bei unserem **Event 2**, mit dem Namen **Bodentafel 2**, abfragen, ob der Protagonist sich auf den **Regions ID 1, 3, 4, 5, 6** oder **9** steht.

Wir gehen also in unser Event und öffnen die **Ereignisbefehle** und wählen uns die **Bedingte Verzweigung** aus:

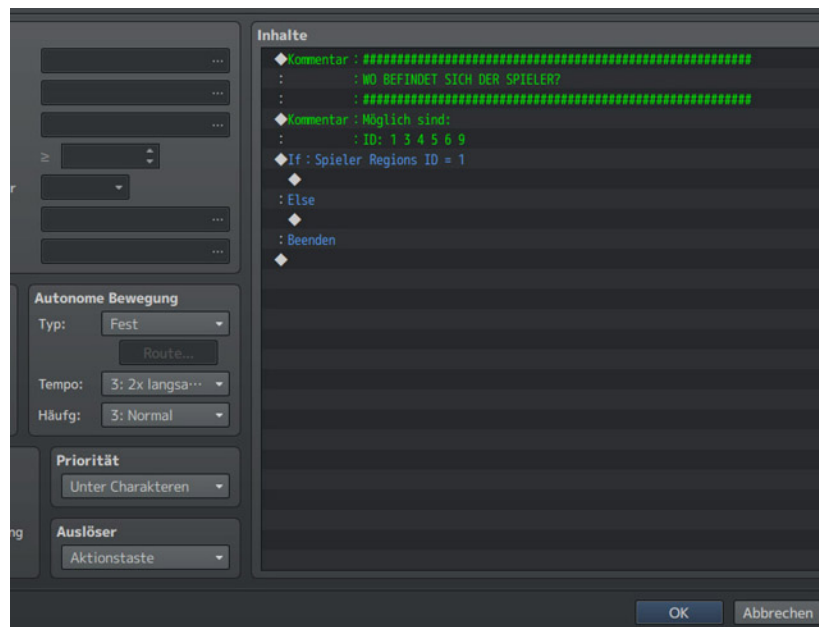


Hier tragen wir folgendes ein:

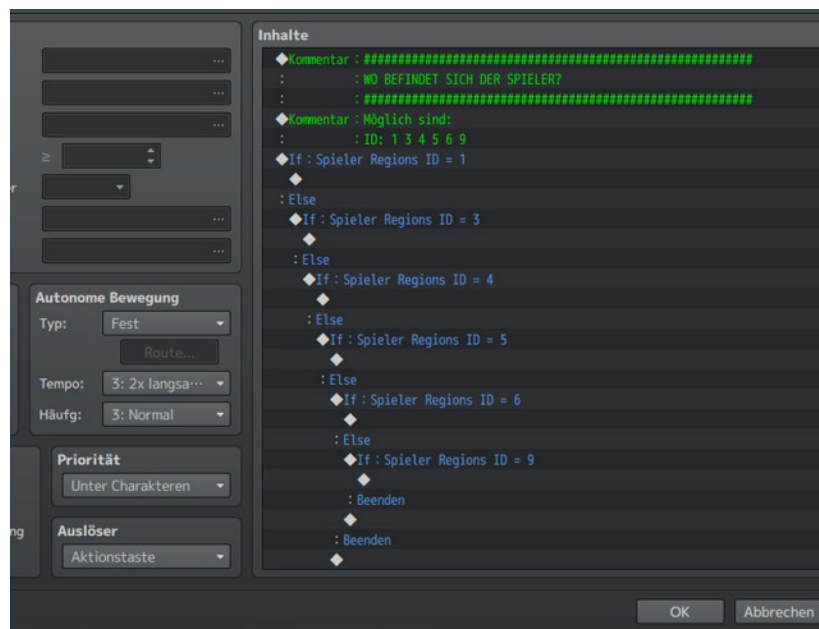
- ➡ **Variable 0003 Spieler Regions ID** als **Bedingung** soll gleich sein einer **Konstante** (eine Konstante ist eine feste Zahl, in unserem Fall die Zahl der Regions ID)
- ➡ Es muss eine **Else-Verzweigung** erstellt werden (Else bedeutet „andernfalls“ – falls also die erste Bedingung nicht erfüllt ist, mach etwas anderes)



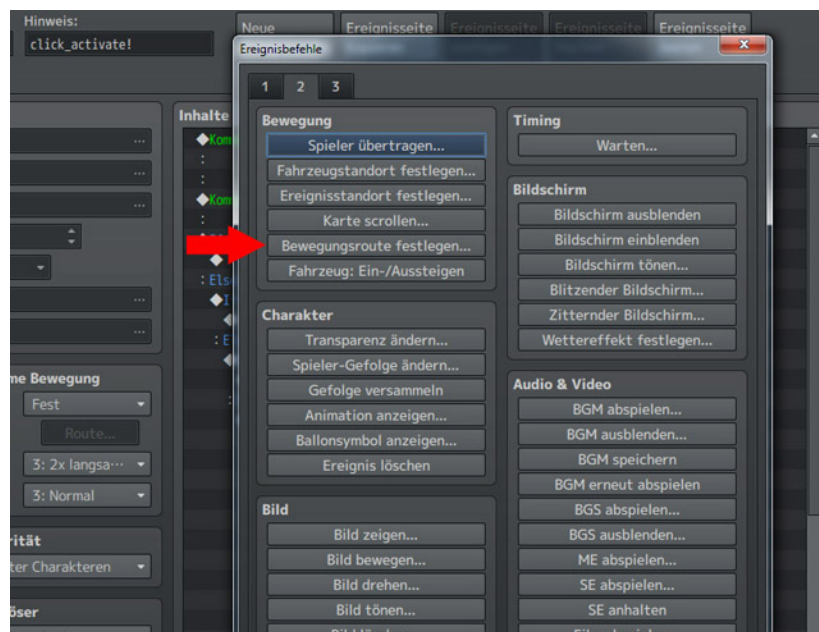
Das ganze müsste bei euch so aussehen:



Zur Erinnerung: Wir müssen abfragen, ob der Spieler auf den **Regions IDs 1, 3, 4, 5, 6 oder 9** steht. Die **1** wäre fertig, folgen nun noch die anderen Zahlen, genau so, wie wir es gerade eben schon gemacht haben:

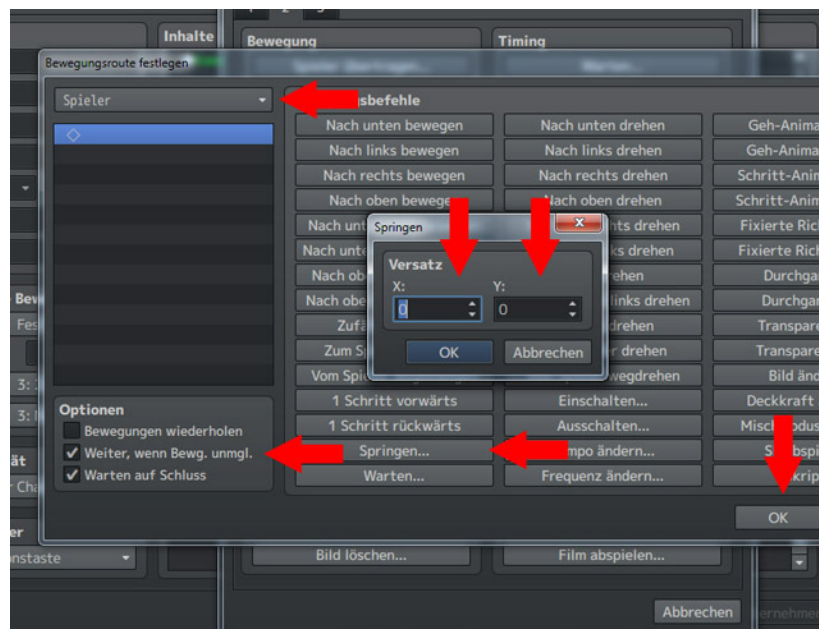


So weit, so gut, jetzt nützen uns jedoch die Bedingungsabfragen nichts, wenn wir nicht festlegen, was passiert, wenn die Bedingung der Abfrage zutrifft, also z.B. der Spieler wirklich auf **Regions ID 1** steht und **Event 2, Bodenplatte 2** angeklickt wird. Im Gedächtnis ist bestimmt hängen geblieben, dass wir den Protagonisten auf das Event holen wollen, welches er anklickt, falls er in Reichweite ist, doch wie machen wir dies? Ganz einfach, mit **Bewegungsrouten**! Mit **Bewegungsrouten** können wir den Protagonisten ganz einfach steuern. Die Einstellung dafür, findet ihr wie immer in den **Ereignisbefehlen**:



Wir tragen also in den **Bewegungsrouteneinstellung** folgendes ein:

- ➡ Der Spieler soll sich bewegen
- ➡ Die Optionen **Weiter, wenn Bewegung unmöglich** und **Warten auf Schluss** sind aktiviert
- ➡ Die Bewegungsart soll **Springen** sein



Aber was tragen wir nun bei X und Y ein? Ich hoffe ihr habt Mathematik in der Schule nicht ganz gehasst, aber leider brauchen wir es. X und Y benutzen wir schon die gesamte Zeit und dürften euch aus dem Koordinaten System bekannt vorkommen:



Dabei ist die linke obere Ecke die Position 0 | 0, also der Ursprungsort der Koordinaten, dies bedeutet:

- ➡ Um so weiter man nach **rechts** geht, um so **größer** wird X
- ➡ Um so weiter man nach **unten** geht, um so **größer** wird Y

Im umgekehrten Fall, würde dies bedeuten:

- ➡ Um so weiter man nach **links** geht, um so **kleiner** wird X
- ➡ Um so weiter man nach **oben** geht, um so **kleiner** wird Y

Da wir dies nun wissen und wollen den Protagonisten von **Regions ID 1** auf **Regions ID 2** bekommen, welche XY- Änderung ergibt sich?

Ganz einfach: Der Protagonist geht **eine Zeile nach unten**, würde bedeuten $Y + 1$ und der Spieler geht **eine Zeile nach links**, würde bedeuten $X - 1$!

Wie würde es bei den anderen Zahlen aussehen?

Regions ID 3: Der Protagonist geht **eine Zeile nach unten**, also nur $Y + 1$.

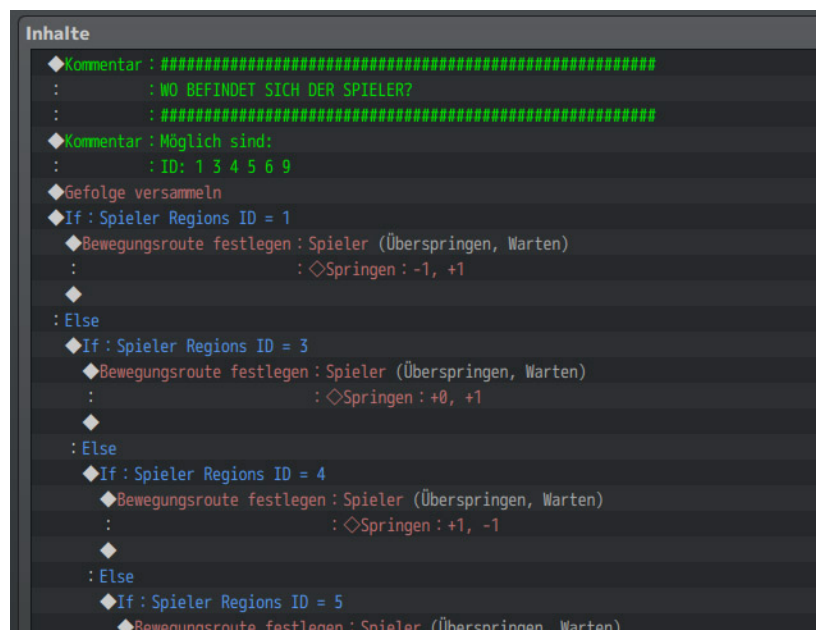
Regions ID 4: Der Protagonist geht **eine Zeile nach oben**, also $Y - 1$ und **eine Zeile nach rechts**, also $X + 1$.

Regions ID 5: Der Protagonist geht **eine Zeile nach rechts**, also nur $X + 1$.

Regions ID 6: Der Protagonist geht **eine Zeile nach unten**, also $Y + 1$ und **eine Zeile nach rechts**, also $X + 1$.

Regions ID 9: Der Protagonist geht **zwei Zeilen nach rechts**, also nur $X + 2$.

Damit hätten wir schon sämtliche Lösungen für das **Event 2, Bodenplatte 2** gefunden:



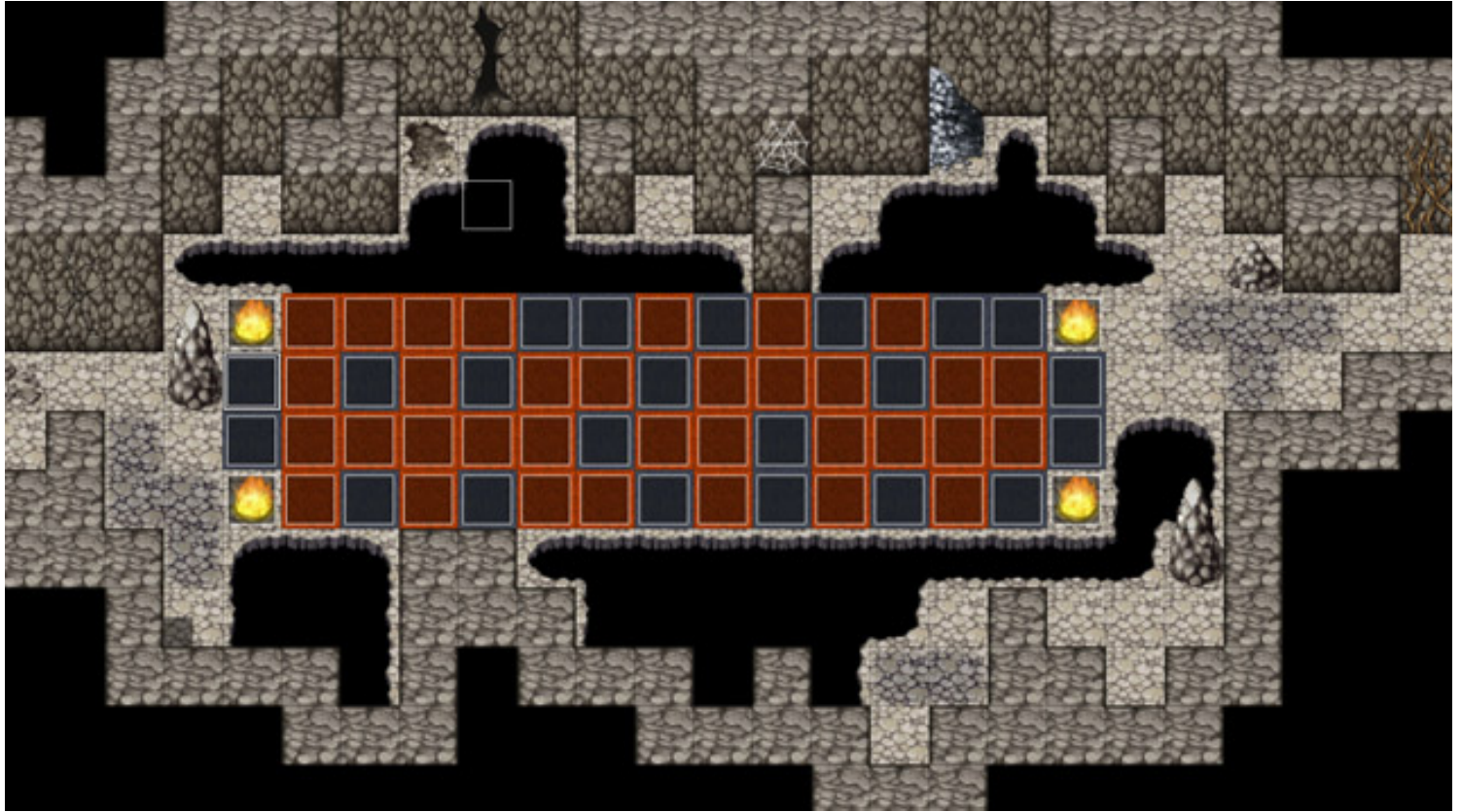
```
Inhalte
◆Kommentar : #####
:           : WO BEFINDET SICH DER SPIELER?
:           : #####
◆Kommentar : Möglich sind:
:           : ID: 1 3 4 5 6 9
◆Gefolge versammeln
◆If : Spieler Regions ID = 1
  ◆Bewegungsroute festlegen : Spieler (Überspringen, Warten)
  :                           : ◇Springen : -1, +1
  ◆
: Else
  ◆If : Spieler Regions ID = 3
    ◆Bewegungsroute festlegen : Spieler (Überspringen, Warten)
    :                           : ◇Springen : +0, +1
    ◆
  : Else
    ◆If : Spieler Regions ID = 4
      ◆Bewegungsroute festlegen : Spieler (Überspringen, Warten)
      :                           : ◇Springen : +1, -1
      ◆
    : Else
      ◆If : Spieler Regions ID = 5
        ◆Bewegungsroute festlegen : Spieler (Überspringen, Warten)
```

Würde also der Protagonist nun auf einer **Regions ID** sein, die im Einzugsbereich vom **Event 2, Bodenplatte 2** ist, und würde der Spieler dieses Event anklicken, würde er durch die **Bewegungsroute** auf dieses Event kommen, bzw. auf **Regions ID 2**. Sollte der Protagonist nicht auf einer zugehörigen **Regions ID** sein und das Event starten, wäre keine der aufgelisteten Bedingungen erfüllt und somit würde auch nichts passieren.

An diesem Punkt fängt für euch die eigentliche Arbeit erst einmal an! Das was wir bei **Event 2, Bodenplatte 2** gemacht haben, muss nun noch bei **Event 3 bis 57** gemacht werden!

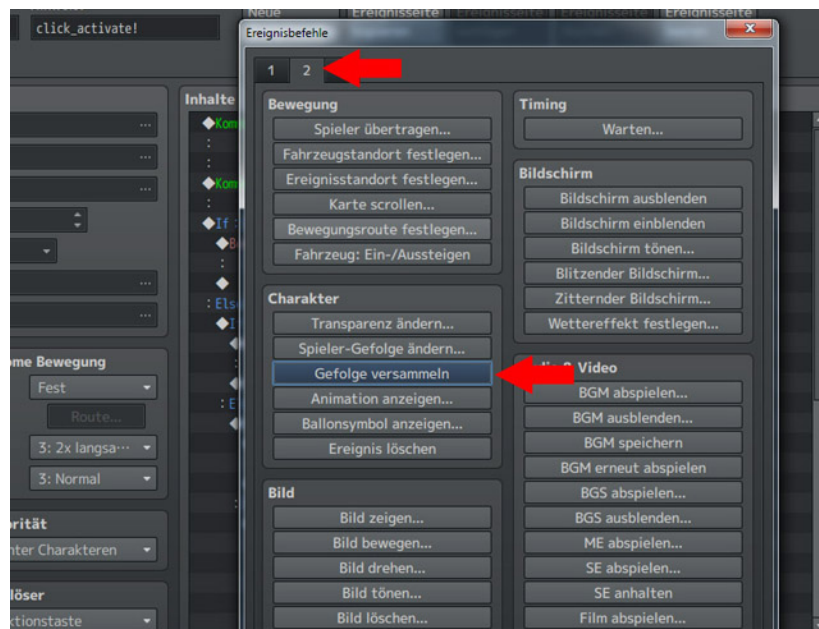
⚠️ Zur Arbeitserleichterung erstellt ihr am besten von euren **Regions ID Feldern** einen **Screenshot** und ebenso eine Markierungshilfe, wie ich sie verwendete, um euch zu zeigen, welche Felder in Reichweite eines bestimmten Feldes sind.

Die Belohnung nach der harten Arbeit dürfe bei euch nun so aussehen:

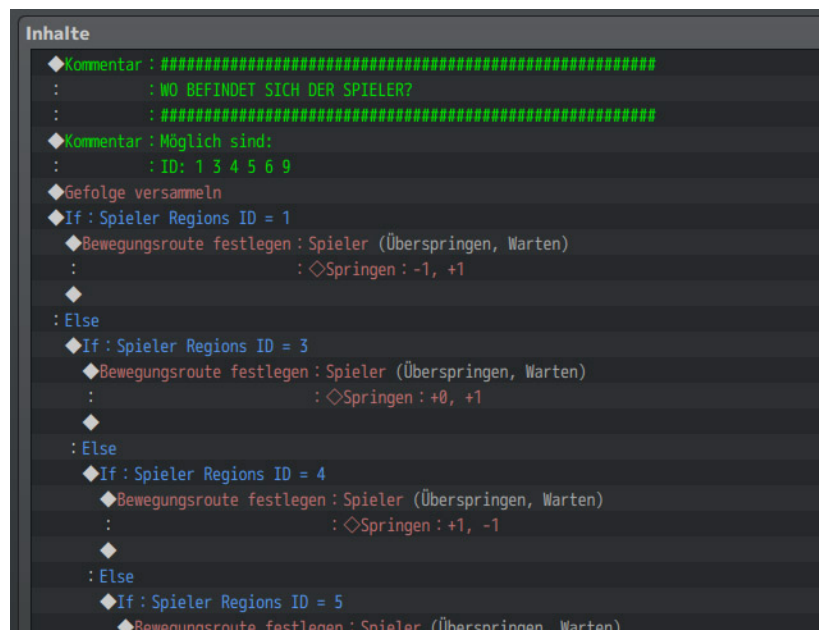


⚠️ Eine Kleinigkeit in diesem Bereich fehlt aber noch! Ist der Protagonist allein, ist diese Steuerung, wie sie nun existiert kein Problem und fertig. Wollt ihr jedoch eine größere Gruppe über diese Brücke führen, benötigt es eine kleine Änderung. Denn sollte der Protagonist springen und hat mehrere Mitstreiter, werden die Mitstreiter nicht auf derselben Kachel landen wie der Protagonist, sondern auf Kacheln hinter dem Spieler. Etwas unschön, vor allem wenn einer der Mitstreiter auf einer Kachel landet, die normalerweise tödlich ist, jedoch nichts passiert.

Keine Panik, dafür gibt es eine einfache Lösung, beim Start jedes Events führen wir am Anfang den Befehl Gefolge sammeln aus. Dieser ist in den **Ereignisbefehlen** zu finden:



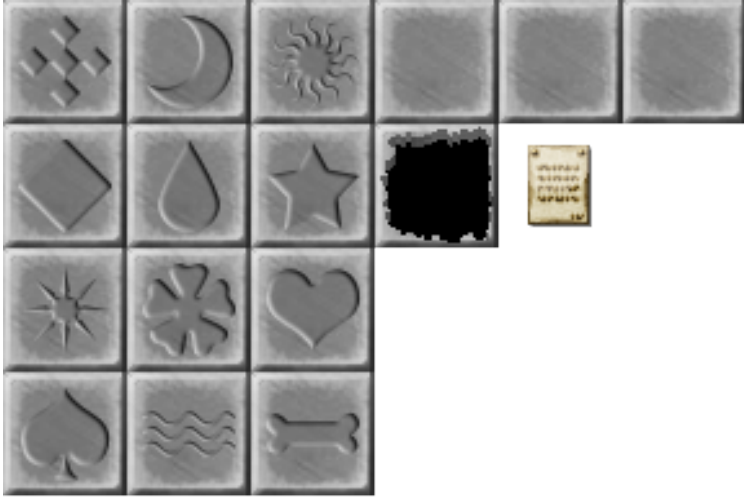
Ist dieser Befehl aktiv, wird bei dem Sprung der Protagonist und sein Gefolge auf derselben Kachel stehen. Der Befehl wirkt aber nur bei diesem einen Event, aus diesem Grund muss er in jedes Event eingebaut werden.



Die guten Kacheln

Wenn ihr euer kleines einmal ausprobiert, dürftet ihr nun euren Protagonisten herumspringen sehen. Ganz gut bisher, doch fehlen noch ein paar Kleinigkeiten!

Zum Beispiel unsere Kacheln fehlen noch, hierzu habe ich für euch einmal etwas vorbereitet:



⚠ Anmerkung: Die Grafik wurde aus dem Original Tilesset des *RPG Maker MV* erstellt, ihr könnt sie also frei nutzen, so fern ihr den *RPG Maker MV* besitzt.

Diese Grafik müsst ihr in euer Projekt einfügen unter:

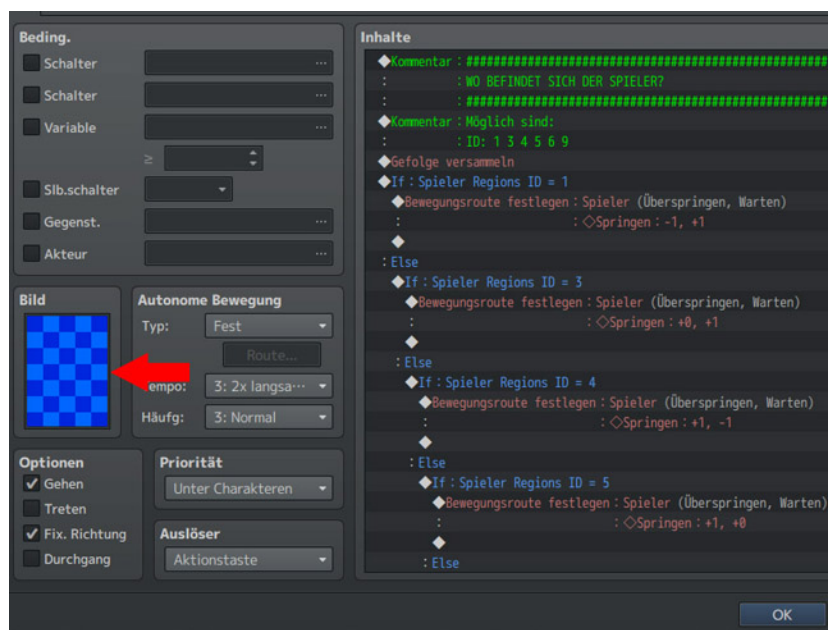
➡ **Bibliotheken\Dokumente\Eigene Dokumente\Games\NAME EURES PROJEKTES\img\characters**

⚠ Anmerkung: Der *RPG Maker MV* hat manchmal etwas Problem mit dem Nachladen von Grafiken, startet am besten noch einmal den Maker neu, wenn ihr die Grafik eingefügt habt.

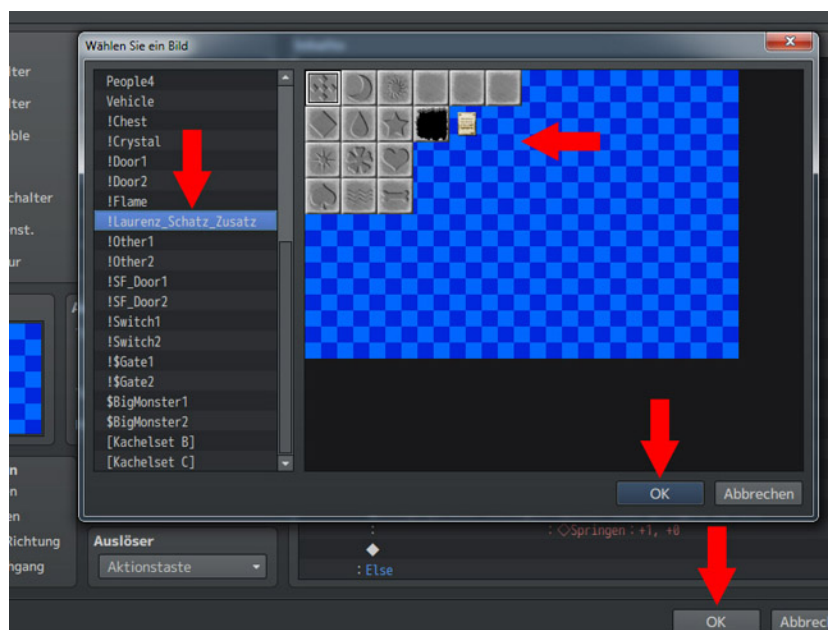
Wie ihr noch von ganz am Anfang des Tutorials sicher wisst, sind dies die Bodenplatten, welche über dem Abgrund liegen. Ihr könnt euch nun heraus suchen, welche Bodenplatten begehbar werden und welche später zu einem Absturz führen. Oder ihr erstellt euch mit den freien Bodenplatten neue. Ich werde als begehbare Platten die erste Spalte nutzen, dazu den Mond aus der zweiten Spalte. Somit hätte ich fünf Symbole die ich betreten kann und sieben die ich nicht betreten kann.

Weiter geht es! Wir legen also jetzt fest, welche Tilesets begehbar werden. Dazu hatten wir uns ja auf dem Boden Markierungen gemacht. Alle Events, die auf einem blauen Untergrund liegen, bekommen jetzt eine Grafik zugewiesen.

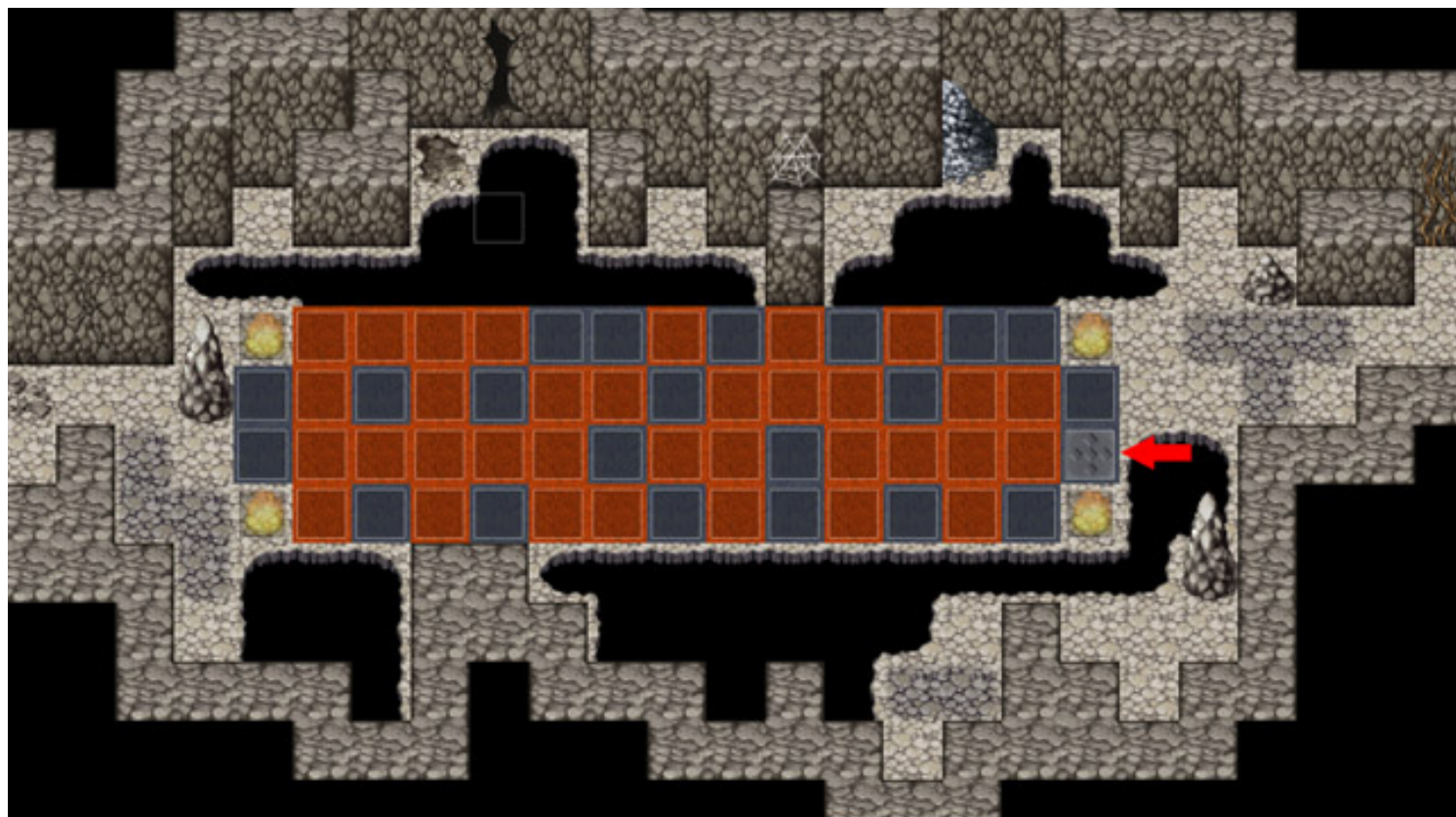
Öffnet also euer **Event 2, Bodenplatte 2**, und klickt auf das Bild ganz rechts in eurem **Ereignis-Editor**:



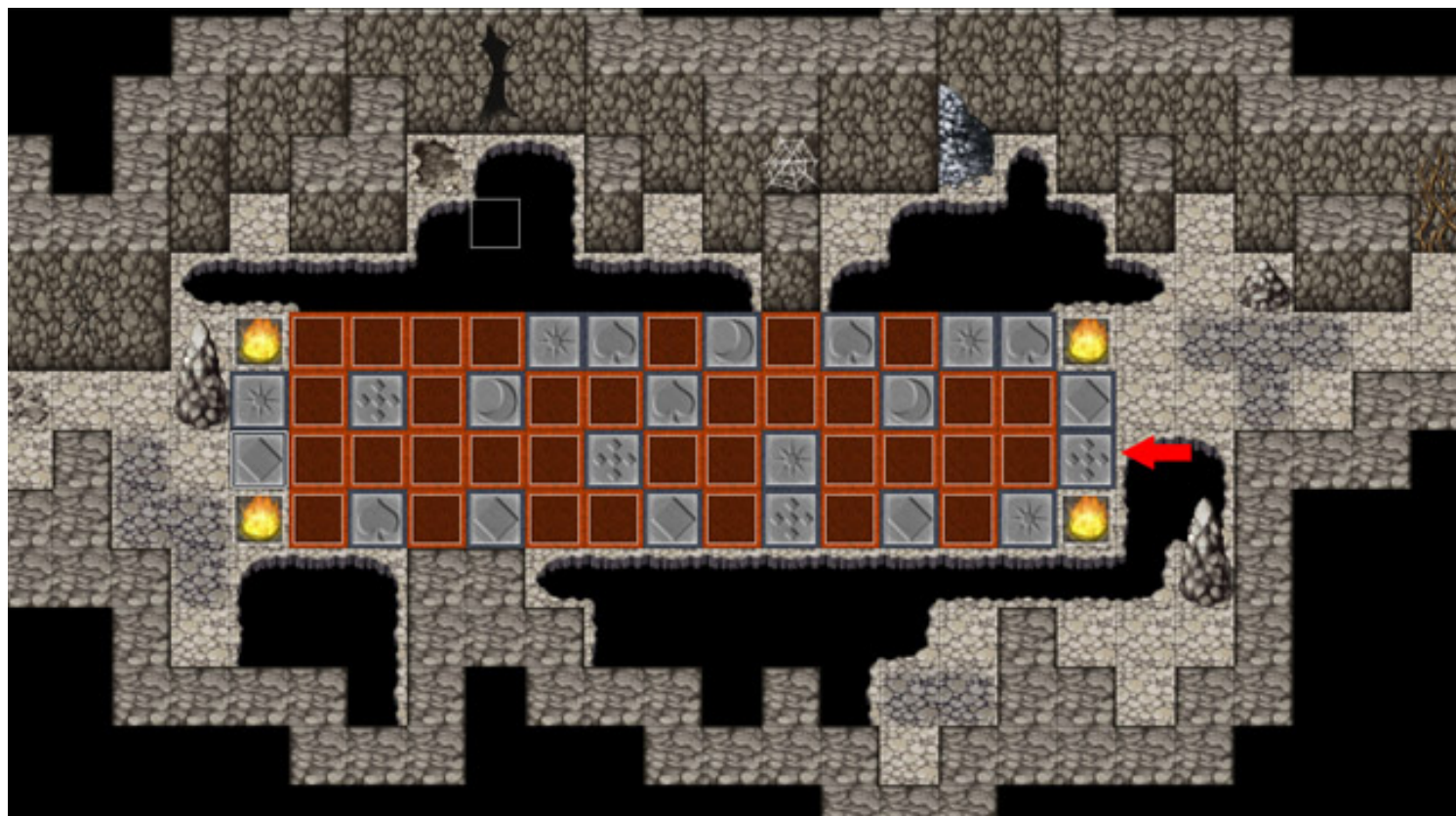
In der **linken Leiste** suchen wir nun unsere **Grafikdatei** und wählen auf der **rechten Seite** unser **Symbol** aus. Ist es ausgewählt bestätigt ihr zwei Mal mit **OK**:



Habt ihr alles richtig gemacht, sieht es bei euch nun so aus:



Diesen Schritt wiederholt ihr nun bei allen anderen Events, die auf einem blauen Untergrund liegen. Achtet dabei etwas auf Abwechslung der Symbole:



Damit wären wir mit den „guten“ Kacheln fertig!

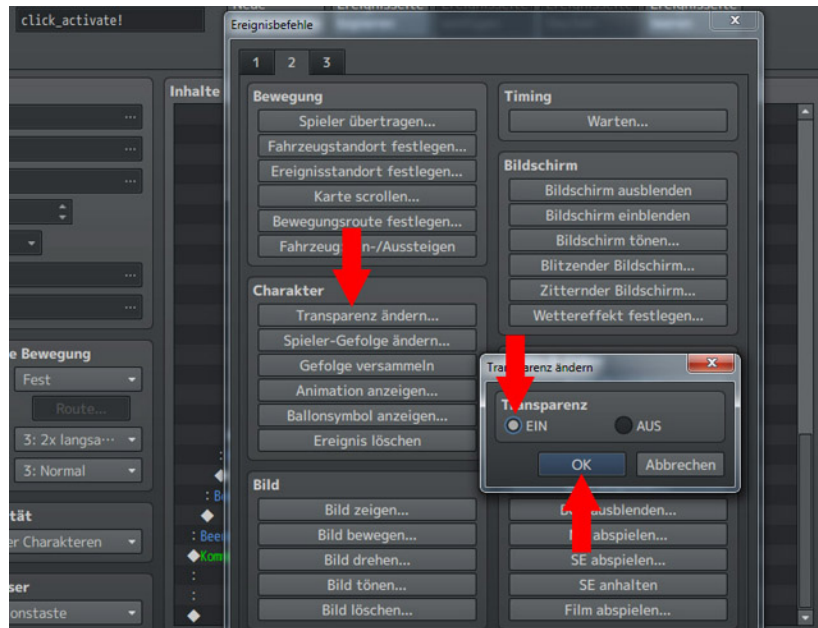
Die bösen Kacheln

In diesem Schritt werden wir uns nun um die „bösen“ Kacheln kümmern. Denn wir müssen noch festlegen was passiert, wenn der Protagonist auf eine falsche Kachel tritt. Entscheiden könnt ihr dies selbst, ob nun ein Pfeil kommt, der den Spieler abschießt oder ob er durch die Bodenplatte fällt oder etwas ganz anderes. Ich habe mich dazu entschieden, dass der Protagonist durch die Bodenplatte fällt.

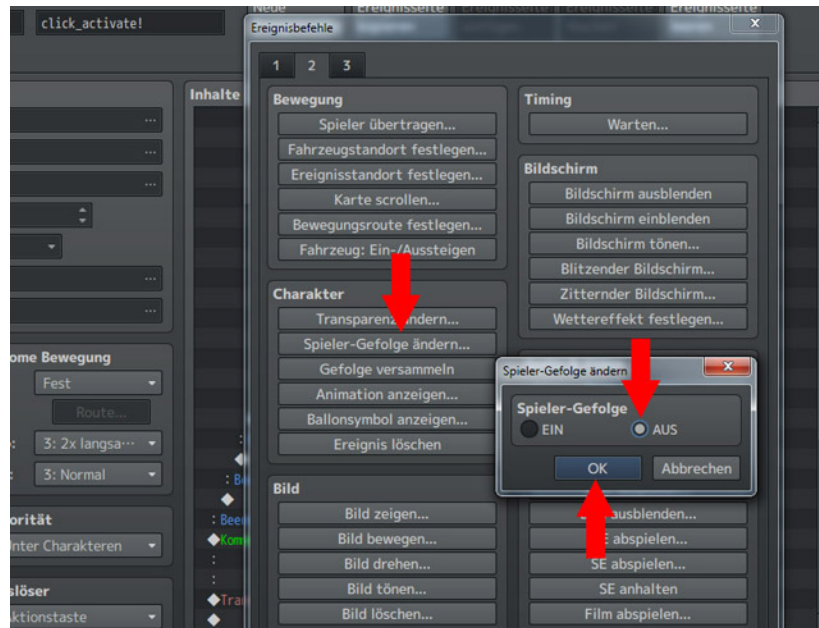
Demzufolge müssen wir, sobald der Protagonist die Kachel betritt folgendes machen:

- ➡ Der Protagonist muss unsichtbar werden, da er durch die Bodenplatte fällt
- ➡ Die Begleiter des Protagonisten müssen unsichtbar werden, da sie ebenfalls durch die Bodenplatte fallen
- ➡ Es muss ein Sound abgespielt werden, dass die Bodenplatte bricht
- ➡ Es muss ein Loch im Boden angezeigt werden, durch welches Protagonist und seine Begleiter fallen
- ➡ Der Spieler muss Game Over gehen

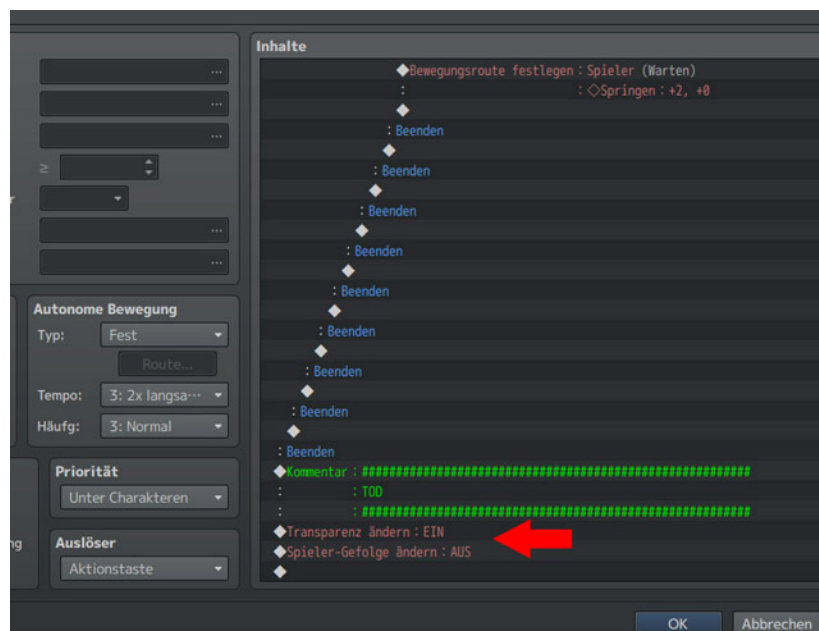
Als Beispiel dient uns **Bodenplatte 5. Am Ende unserer Bedingungsabfrage** kommt als erstes der Befehl, dass der Protagonist unsichtbar ist. Dazu gehen wir in die **Ereignisbefehle** auf die **zweite Reiterseite**. Auf dieser Seite aktivieren wir **Transparent ändern...**, Schalten die **Transparenz** auf **EIN** und bestätigen mit **OK**:



Auf derselben **Reiterseite** finden wir auch den passenden Befehl, um unsere Gruppe unsichtbar zu machen. Wir gehen auf **Spieler-Gefolge ändern** und sagen hier **AUS**, bestätigt wird wieder mit **OK**:



Bei euch sieht das ganze demnach so aus:



Fertig? Dann geht es weiter, wir brauchen noch einen Sound, der zeigt, dass die Bodenplatte bricht. Geht wieder in die Ereignisbefehle auf die zweite Reiterseite und suchen uns SE abspielen.

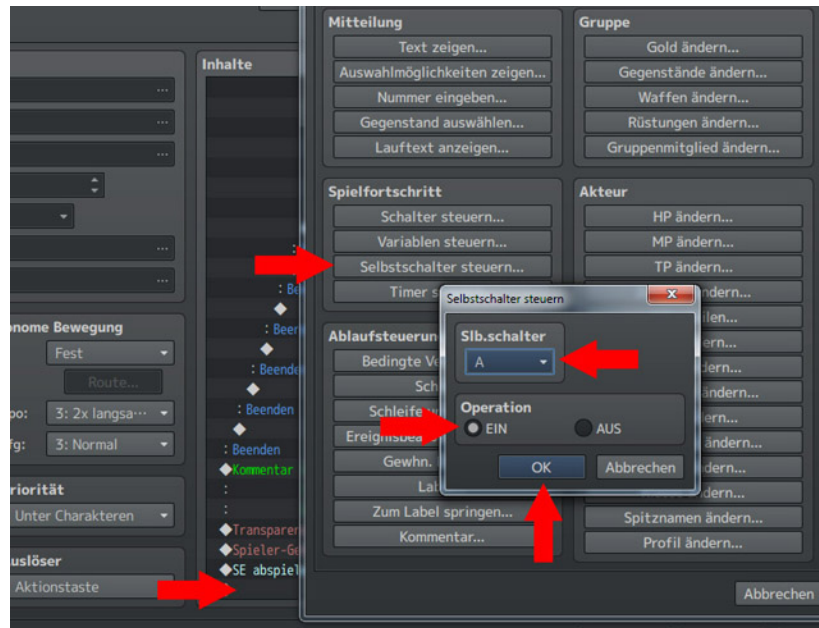
Eingestellt wird hier folgendes:

- ➡ SE-Sound: **Damage2**
- ➡ Lautstärke: **90%**
- ➡ Tonhöhe: **50%**

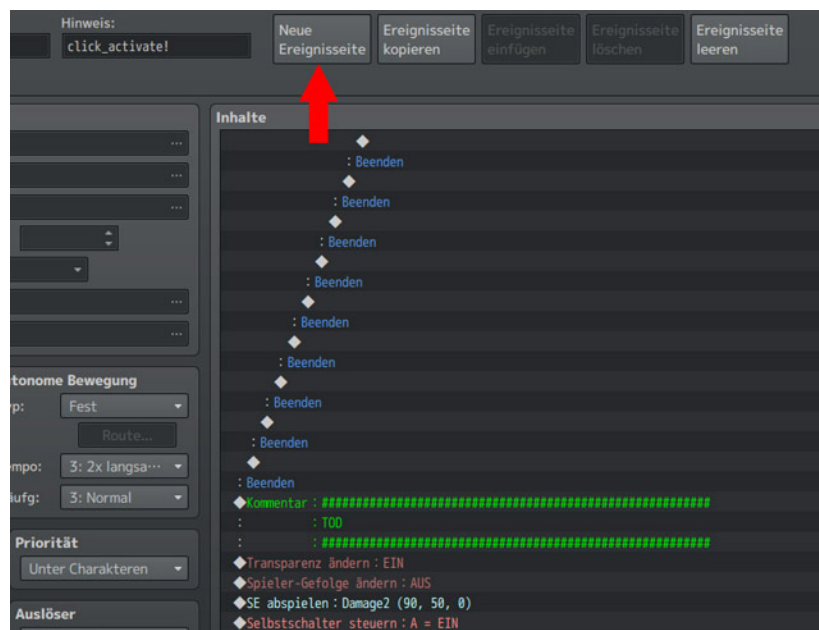
Anschließend wird wieder mit **OK** bestätigt.

Was nun noch fehlt, ist das Loch im Boden, dazu erstellen wir erst einmal einen **Eigenschalter**, der uns auf eine **neue Eventseite** bringt. Geht in die **Ereignisbefehle** auf die **erste Reiterseite**. Klickt hier auf **Selbstschalter steuern**, im neuen Fenster auf **Selb.schalter A**, **Operation EIN**, wie immer bestätigen wir mit **OK**.

Passt auf, dass dieser Befehl **ganz am Ende** steht:

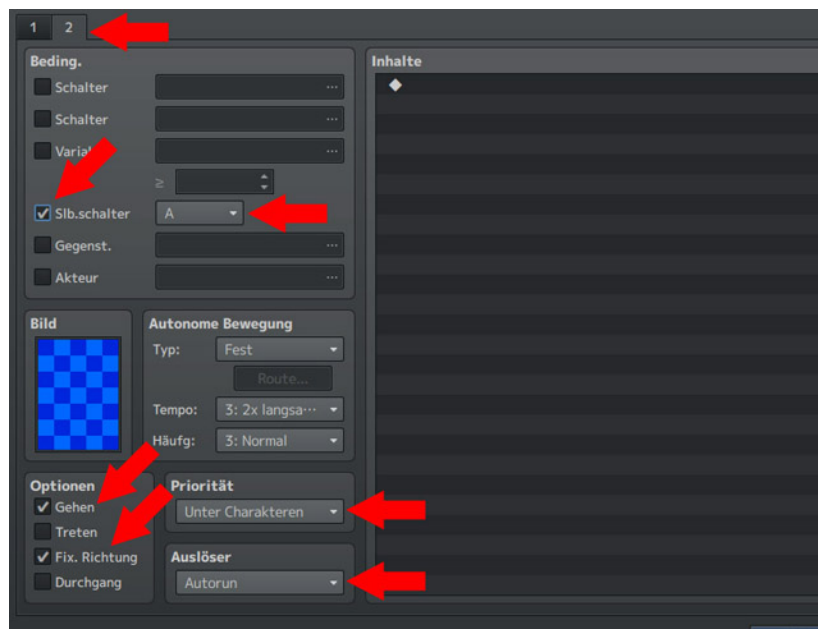


Erzeugt wird nun als letztes eine **neue Ereignisseite**, dies funktioniert recht einfach im **Ereignis Editor**:



Es öffnet sich ein **neuer Reiter**, in diesem stellen wir folgendes ein:

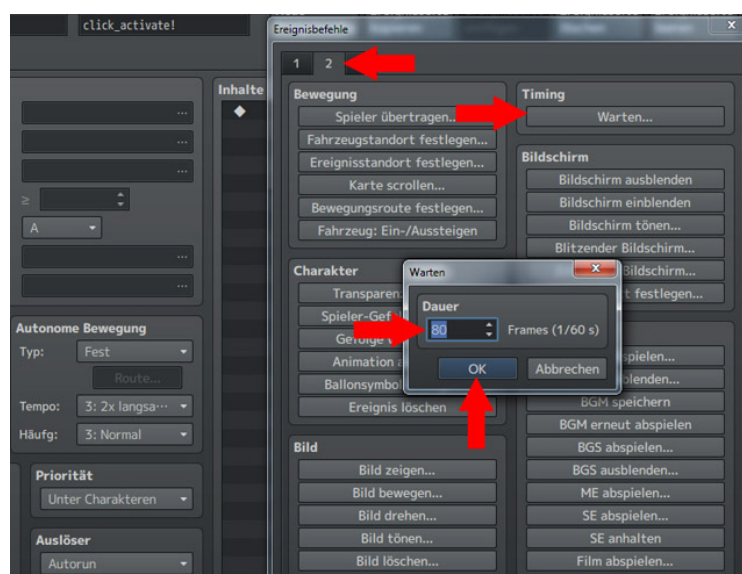
- ➡ **Selb.schalter: A**
- ➡ **Optionen: Gehen und Richtung Fixieren**
- ➡ **Priorität: Unter Charakter**
- ➡ **Auslöser: Autorun** (Autorun ist dafür da, dass das Event sofort gestartet wird, sobald es aufgerufen wird und der Protagonist sich nicht bewegen kann)



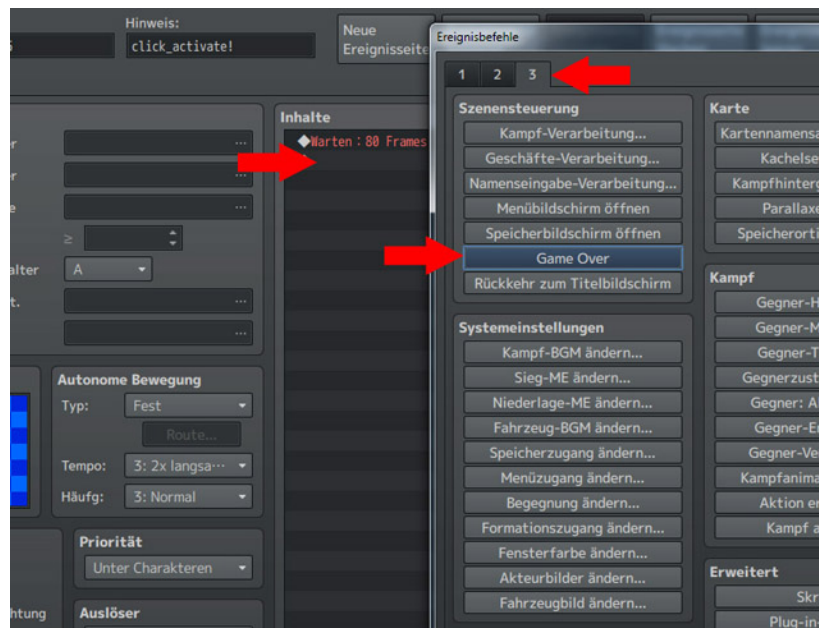
Fassen wir noch einmal zusammen, was passiert nun:

Sobald der Protagonist das Event betritt, wird sowohl er, als auch seine Gruppe unsichtbar geschaltet. Ein Sound wird abgespielt, dass die Bodenplatte bricht und das Event wird auf eine neue Eventseite umgeleitet. Dies alles würde innerhalb eines Bruchteils einer Sekunde passieren. Der Protagonist geht also auf die Platte und wäre sofort **Game Over**. Recht unschön, denn der Spieler wird nun machen: Hääää???? Es muss also Zeit gegeben werden, das der Spieler realisiert, dass der Protagonist durch die Bodenplatte bricht.

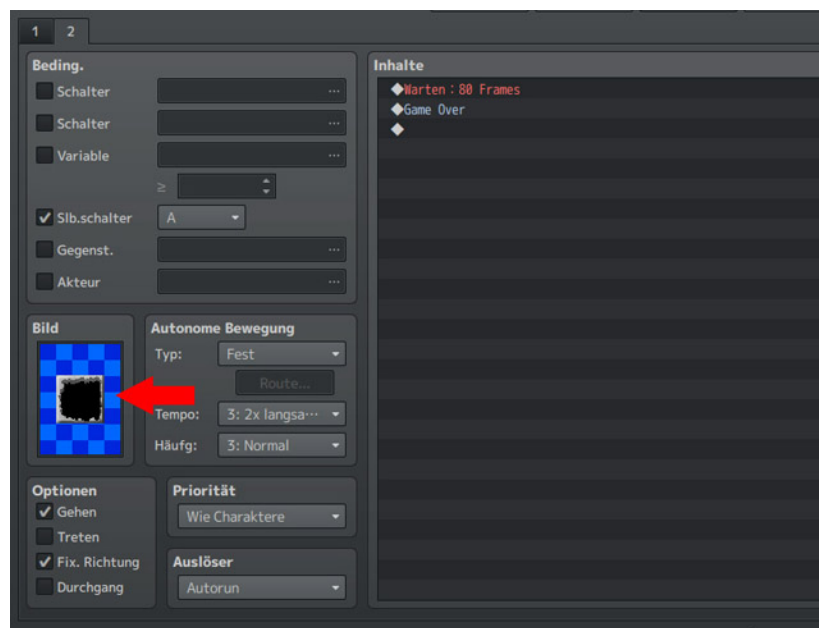
Wir sorgen also für einen Halt, dies nennt man **Warte-Befehl**. Geht also auf die **zweite Reiterseite** der **Ereignisbefehle** und da auf **Warten**. Hier stellen wir einen passenden Wert ein, ich trage **80 Frames** ein. Ein Frame bedeutet 1/60 einer Sekunde. Stelle ich also **80 Frames** ein, wird ein Wartebefehl von **1,3 Sekunden** ausgeführt. Ihr könnt hier gern etwas mehr oder weniger eintragen, ist halt Geschmackssache. Bestätigt wird wie immer mit OK.



Im letzten Schritt auf dieser Eventseite wird noch der **Game Over** ausgelöst. Wieder einmal begeben wir uns in das Fenster der **Ereignisbefehle**, dieses Mal **dritte Reiterseite**, hier finden wir ganz groß **Game Over**:

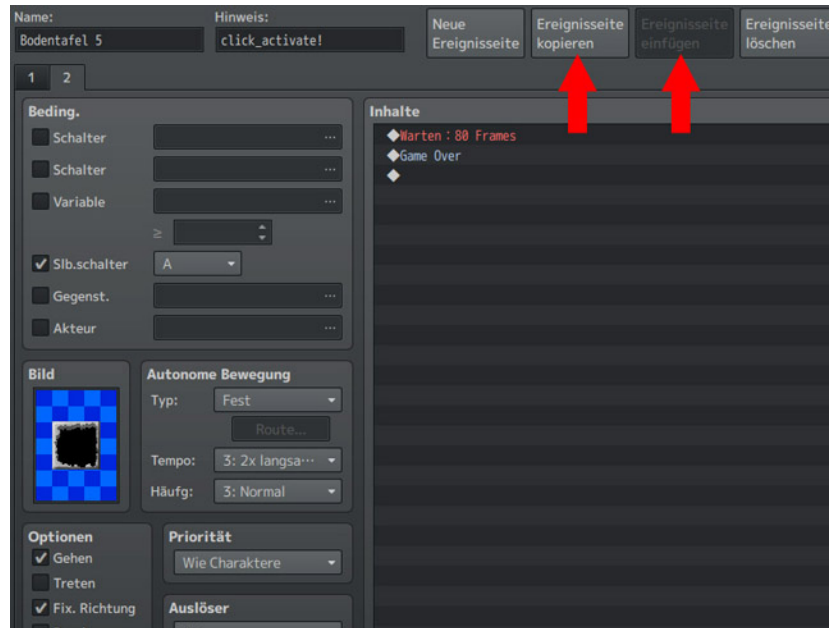


Als aller letztes auf dieser Eventseite müssen wir die neue Grafik des Events eintragen, den **Schaden im Boden**. Ihr kennt den Schritt schon aus **Teil 4 des Tutorials**, aus diesem Grund erkläre ich nicht noch einmal alles. Das Event würde nun so aussehen:

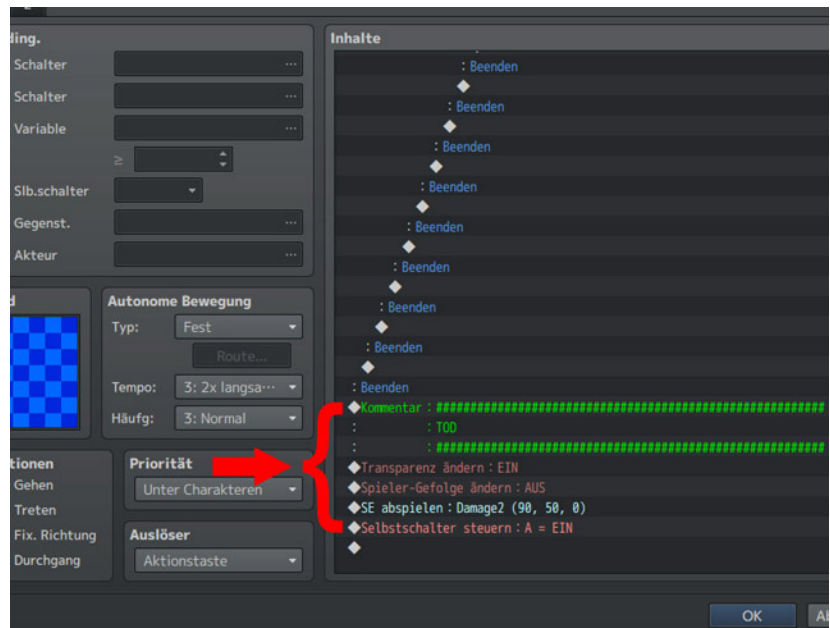


Damit wäre diese Eventseite auch fertig. Da auf jeder Bodenplatte dasselbe passiert, können wir die nun erstellten Eventbefehle **Kopieren** und in den Events einfügen, die einen **roten Untergrund** haben.

Zu allererst kopieren wir uns die Eventseite und fügen sie durch Einfügen in den anderen Events ein:

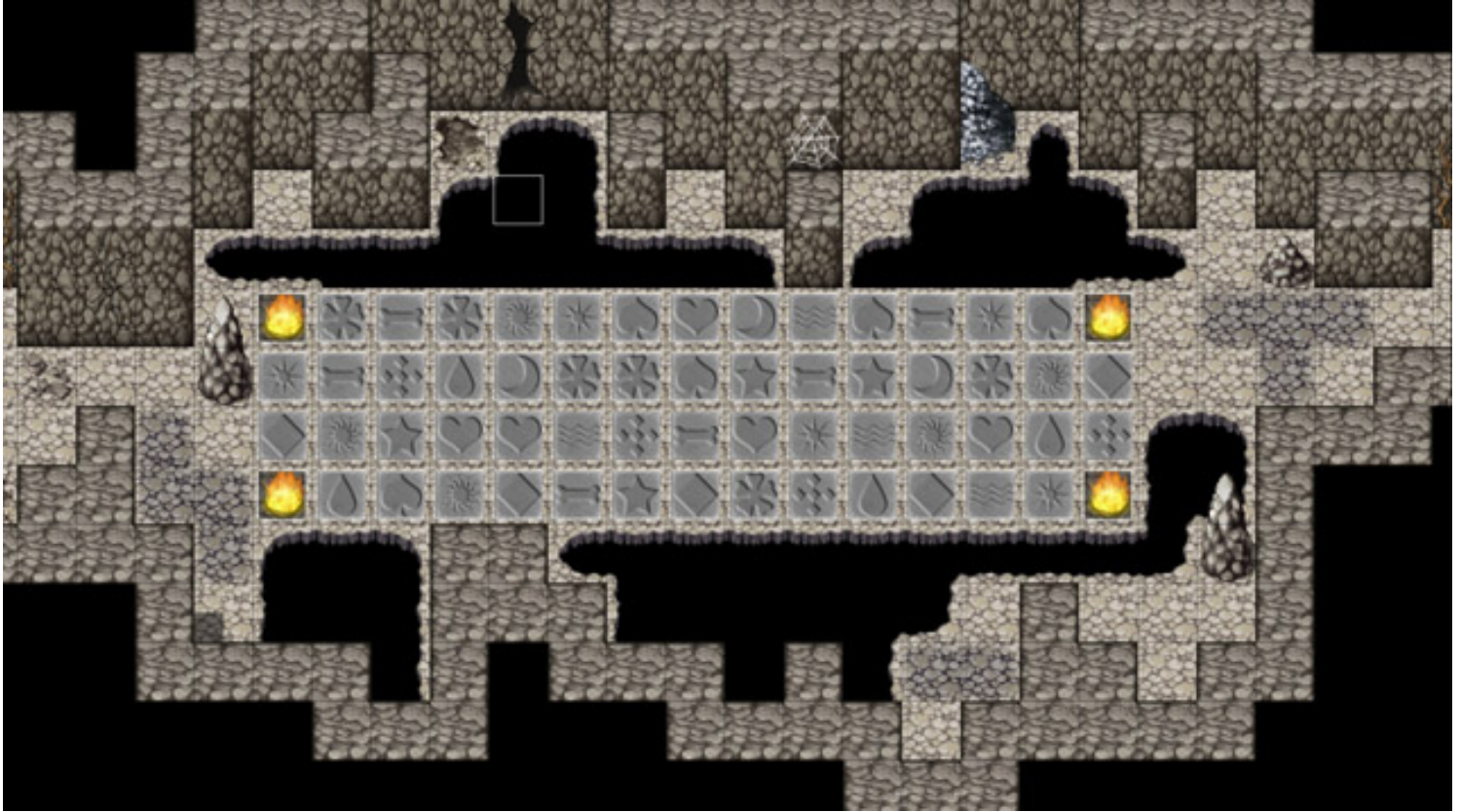


Danach müssen wir noch den letzten Teil des Events kopieren:



Mit **gedrückter Shift-Taste**, markiert ihr euch den gesamten Bereich und fügt ihn in alle anderen Events ein, die einen **roten Untergrund** haben.

Fertig? Dann fehlen nun noch die restlichen Grafiken der roten Events. Dieses Mal nutzt ihr die Grafiken die übrig sind, keinesfalls die gleichen, die wir bei den guten Kacheln verwendet haben, bei mir sieht das ganze nun so aus:

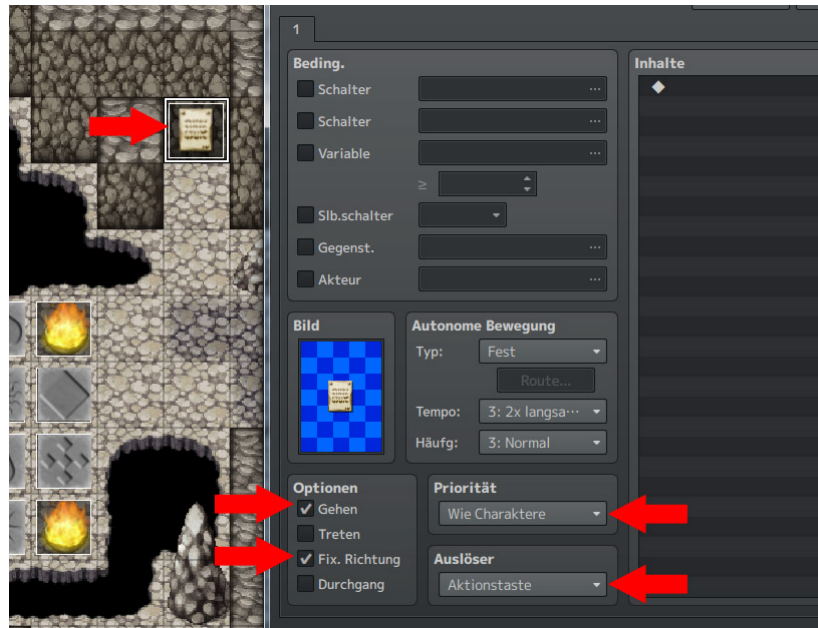


Wie ihr seht, habe ich den Boden wieder komplett grau gefärbt. Im Grund genommen könnt ihr die Markierungen unter den Kacheln lassen, es ist eine gute Hilfe, falls ihr mal etwas verändern wollt. Da die Kacheln genau so groß sind wie der Untergrund würde man die Farbe auch nicht sehen. Ich habe sie nur einmal für euch entfernt, damit ihr die nun fertigen Kacheln besser sehen könnt.

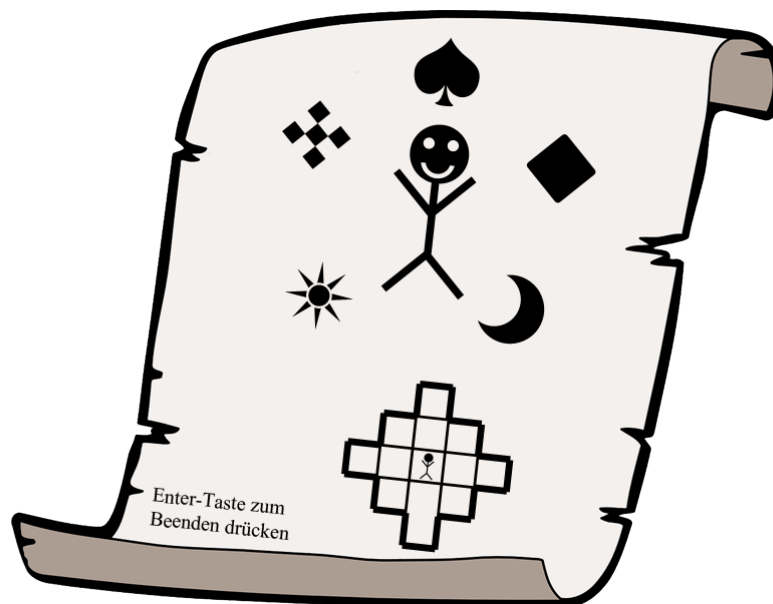
Im Grunde genommen wäre das Event nun fertig und bespielbar. Doch fehlt dem Spieler noch der Hinweis, welche Kacheln er betreten kann und welche besser nicht. Ihr könnt dies machen wie ihr wollt, ich werde dem Spieler im nächsten Teil des Tutorials eine kleine Hilfe geben.

Die Hilfe

In diesem Teil werden wir den Spieler eine kleine Hilfe geben, wie er unbeschadet auf die andere Seite kommt. Dazu habe ich an der Wand einen kleinen Zettel angebracht und ein Event erzeugt. Da ich dies alles schon einmal erklärt habe, hier nur einmal kurz meine Einstellungen:



Um den Spieler eine Hilfe zu geben, werden wir den Spieler eine Grafik anzeigen lassen, genauer gesagt dies hier:



Eingefügt wird die Grafik in eurem Projekt unter:

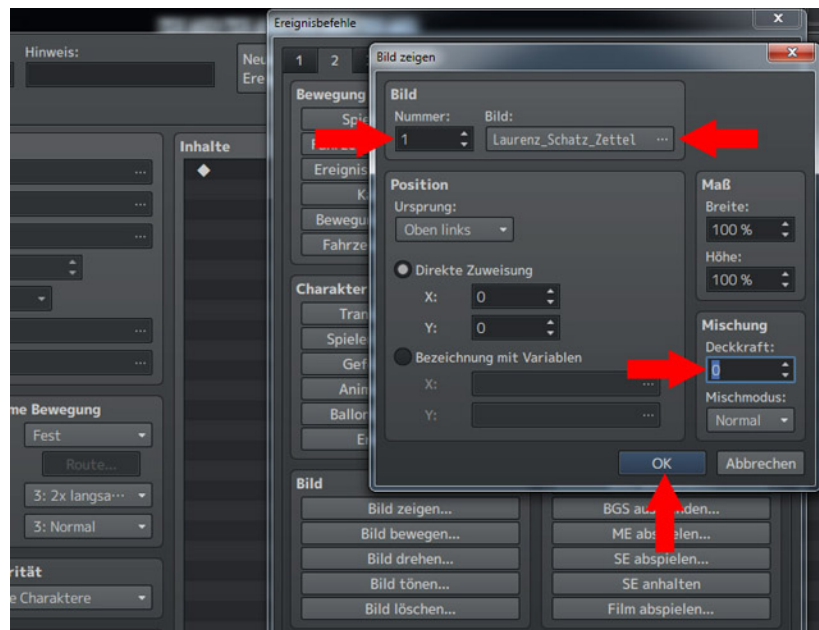
➡ **Bibliotheken\Dokumente\Eigene Dokumente\Games\NAME EURES PROJEKTES\img\pictures**

⚠ Anmerkung: Die Grafik des Pergaments ist frei von pixbay.de von OpenClipart-Vectors, das Geschmiere darauf ist von mir, ihr könnt sie gern verwenden.

Gehen wir zurück zum Event, hier soll das Bild langsam, mit einem kleinen Blendeffekt eingefügt werden. Dazu gehen wir wieder einmal in den **Ereignisbefehlen**, auf die **zweite Reiterseite**. Wir suchen uns den Befehl **Bild anzeigen...** und stellen folgendes ein:

- ➡ Bild Nummer: 1
- ➡ Bild: Laurenz_Schatz_Zettel
- ➡ Deckkraft: 0 (bedeutet das, dass Bild zwar aufgerufen wird, es jedoch keine Deckkraft hat, sprich es ist unsichtbar)

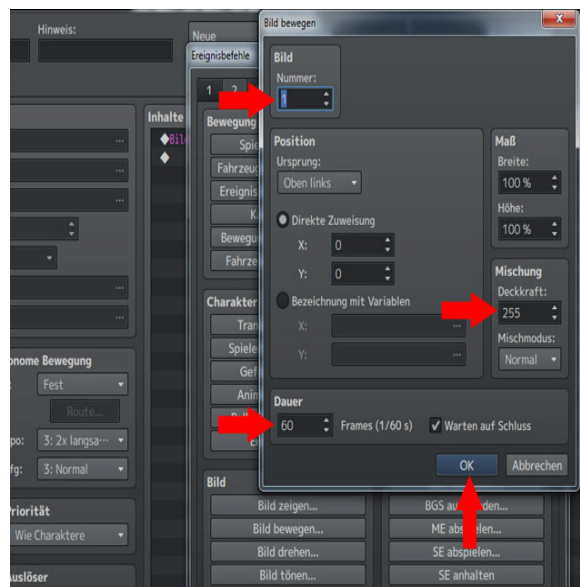
Alles andere lassen wir genau so und bestätigen wie immer mit **OK**.



Damit wir einen Blendeffekt bekommen, gehen wir abermals in die **Ereignisbefehlen** und klicken auf **Bild bewegen...**, hier werden folgende Einstellungen vorgenommen:

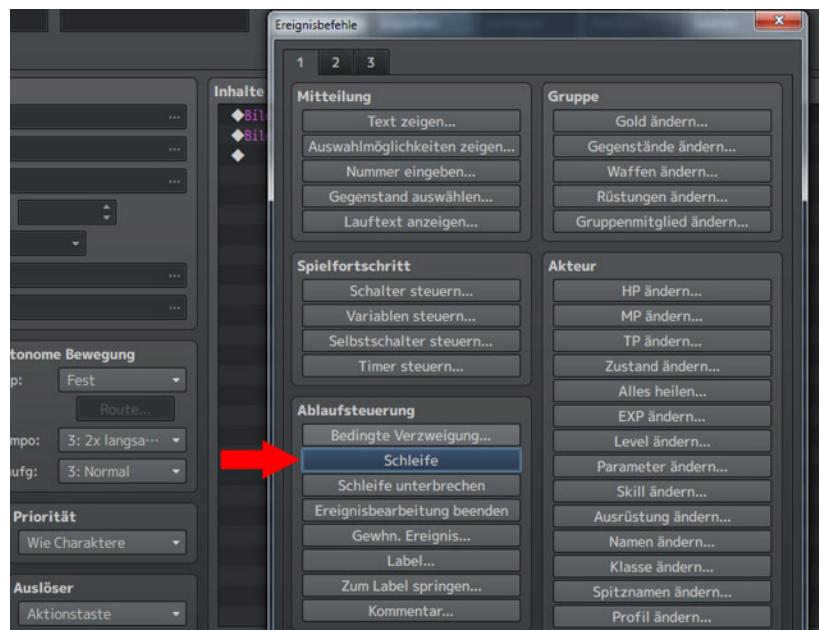
- ➡ Bild Nummer: 1
- ➡ Deckkraft: **255** (bedeutet das, dass Bild nun auf volle Sichtbarkeit geschaltet wird)
- ➡ Dauer: **60** (bedeutet das, dass Bild innerhalb von 60 Frames (1 Sekunde) voll sichtbar wird)
- ➡ Warten auf Schluss muss markiert sein (sorgt dafür das, dass Bild erst komplett angezeigt wird, bis ein weiterer Eventbefehl bearbeitet wird)

Wie immer, OK zum Bestätigen, damit hätten wir das Bild erst einmal angezeigt:

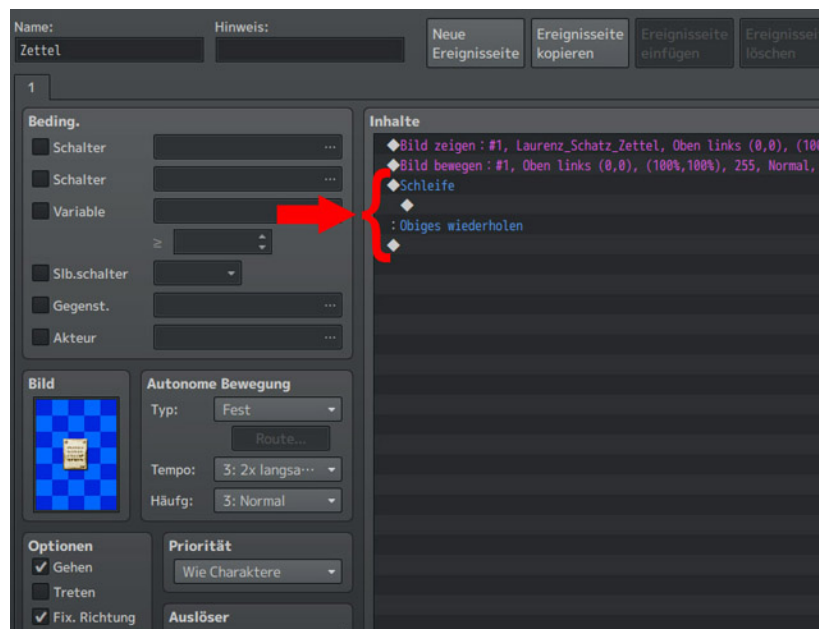


Das Bild wäre jetzt für den Spieler sichtbar, doch wollen wir auch, dass der Spieler das Bild wieder entfernen kann. Aus diesem Grund nutzen wir nun eine **Schleife**. Eine Schleife im *RPG Maker* ist ein Prozess der **innerhalb eines Events so lange abgespielt wird, bis man ihn beendet**, keinesfalls ist dies mit einem Parallelen Prozess zu verwechseln. Eine Erläuterung folgt gleich.

Die Schleife findet ihr in den **Ereignisbefehlen**, auf die **ersten Reiterseite**:

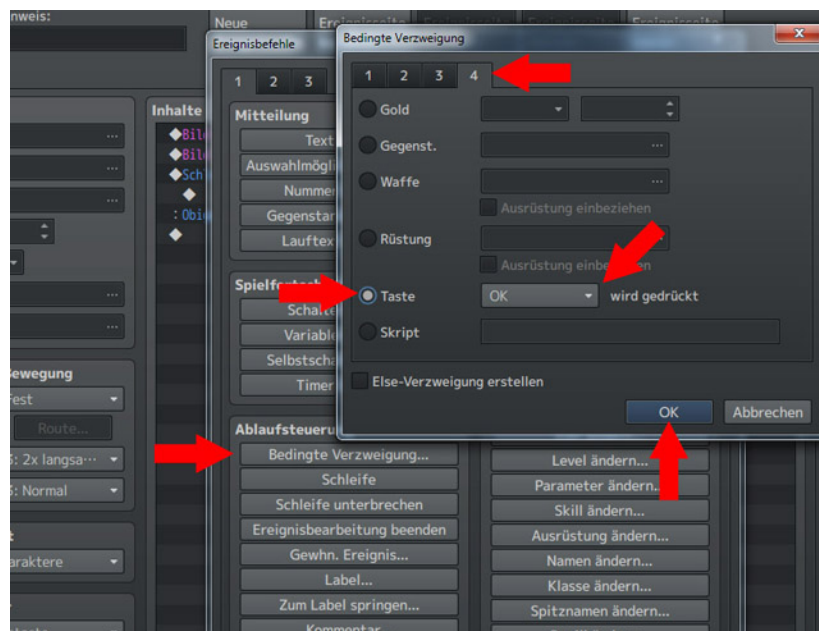


Nun zur Erklärung, die **Schleife** würde **nur den dauerhaft markierten Bereich wiederholen**, so lange, bis aus dem **inneren Bereich der Schleife ein Abbruch Befehl kommt**. Der Unterschied zu einem **Parallelen Prozess** ist, dass der **Parallele Prozess** ständig das **gesamte Event** abarbeitet, die **Schleife** nur einen **begrenzten Bereich**:

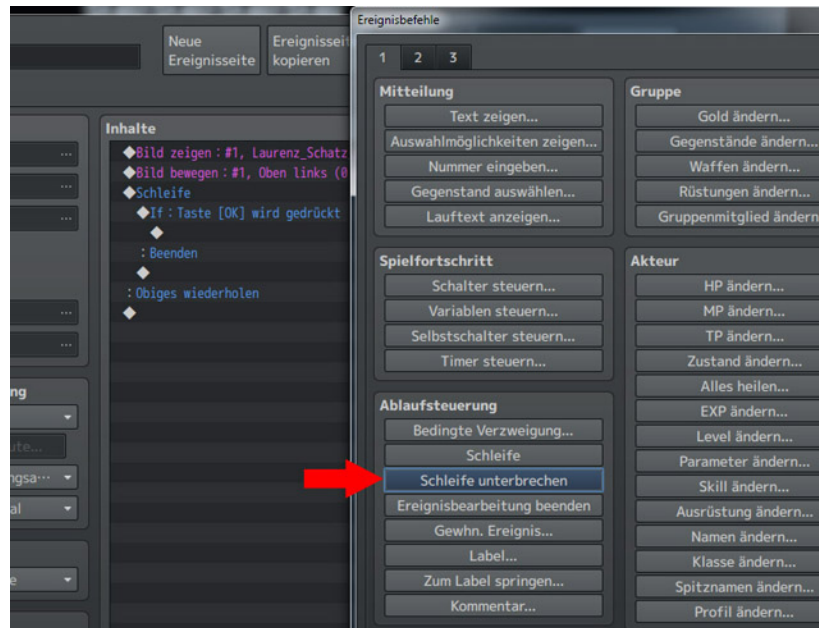


Warum brauchen wir eine **Schleife**? Wir werden jetzt, so lange das Bild angezeigt ist dauerhaft abfragen, ob der Spieler eine Taste drückt. Drückt er diese Taste, wird das Event beendet und die Grafik wird wieder vom Bildschirm verschwinden.

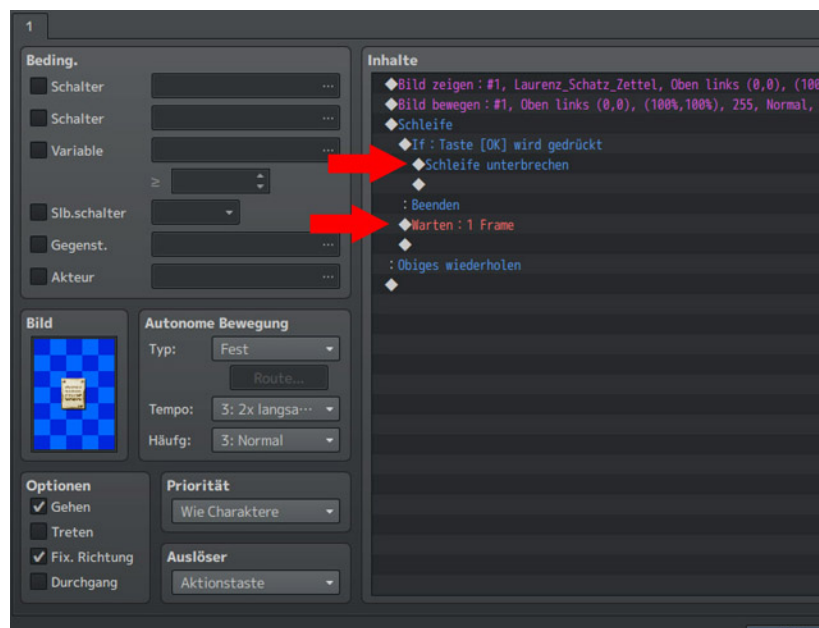
Dazu erzeugen wir eine **Bedingungsabfrage**, gehen in **Bedingte Verzweigung** auf die **vierte Reiterkarte** und wählen eine **Tastenabfrage** aus, hierbei die **Taste OK**, diese Taste wäre die **Enter-Taste** auf der Tastatur:



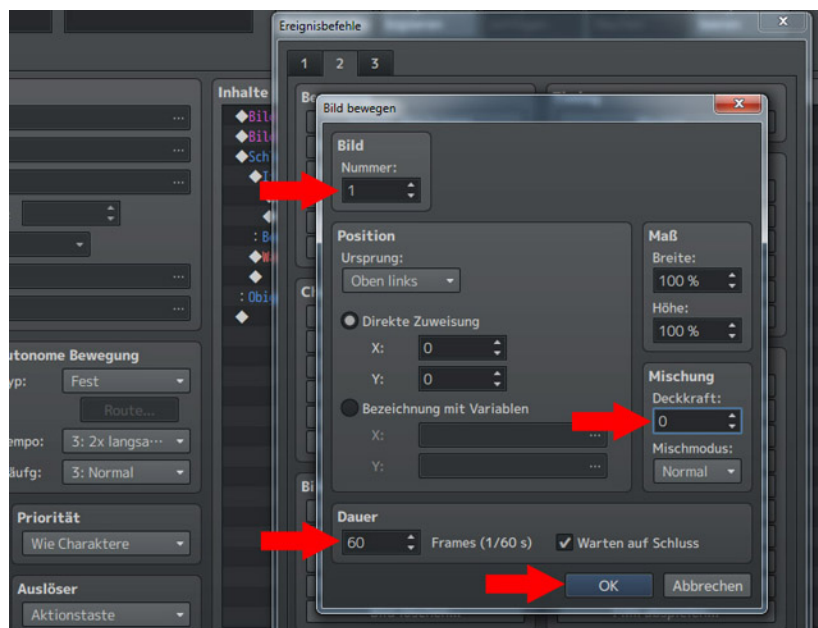
Wir fragen also dauerhaft ab, ob die **Enter-Taste** gedrückt wird, passiert dies, soll die **Schleife verlassen** werden, dazu nutzen wir den Befehl **Schleife unterbrechen** in den **Ereignisbefehlen**, auf **Reiterseite 1**:



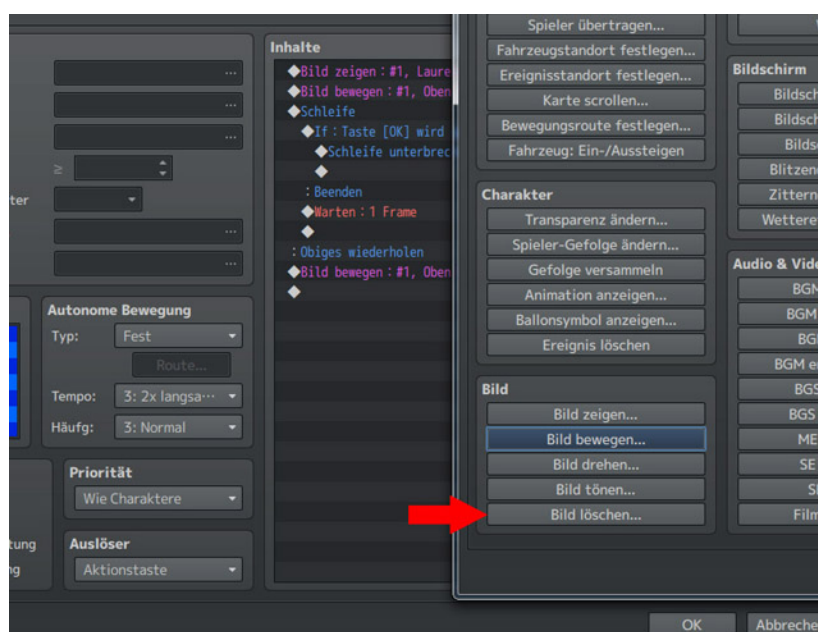
Die **Schleife** hat beim *RPG Maker* einen kleinen Bug. Würde man diese Einstellungen nun so nutzen würde sich der *RPG Maker* bzw. das Spiel aufhängen. Grund ist, dass die **Schleife** sehr schnell abgearbeitet wird und durch die schnellen und vielen Wiederholungen sich beinahe versucht, selbst zu überholen. Dabei entsteht ein Bug und das Spiel friert ein. Diesen Bug kann man aber leicht beheben, in dem wir einen Warte-Frame einfügen, dieser löst das Problem:



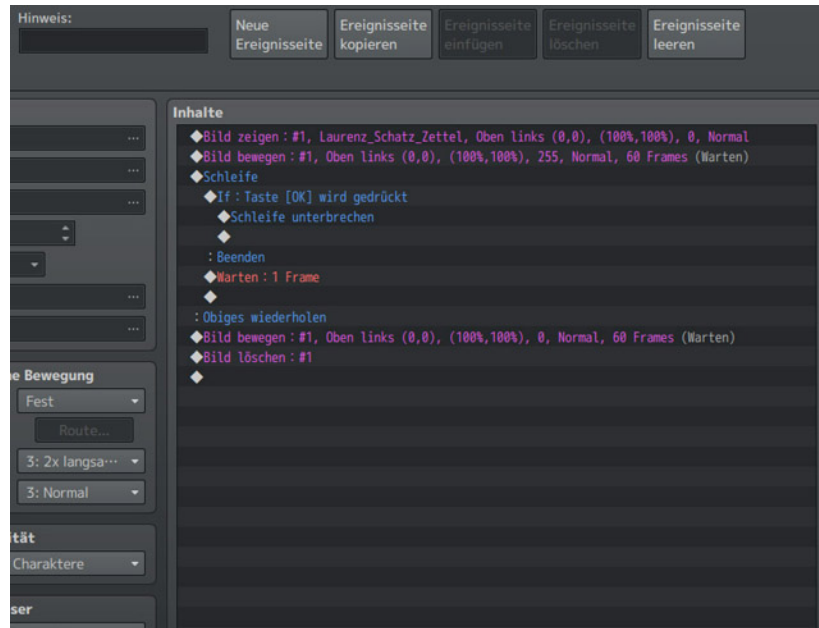
Was soll nun noch passieren? Das Bild soll langsam ausgeblendet werden, dazu nutzen wir wieder den Befehl **Bild bewegen**..., dieses Mal wird aber die **Deckkraft** auf **0** gesetzt:



Zum Schluss soll das Bild wieder verschwinden, dazu nutzen wir den Befehl **Bild löschen**...:



Am Ende sieht das Event so aus:



Damit wären wir am Ende unseres kleinen Tutorials angekommen. Ich hoffe ihr hatten Spaß daran und könnt euer Spiel um ein schönes, kleines Minispiel erweitern.

Ich wünsche Euch viel Erfolg!!

Demo-Tutorial

Das komplette Tutorial könnt ihr als *RPG Maker MV* Projekt hier laden:

<http://gamealchemists.com/?wpdmdl=588>