

Das Stachelfallenrätzel- Tutorial

(Version 1.0 – 12.05.2018)

für

RPG Maker MV & RPG Maker VX Ace

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 Urhebernamentens ist nicht gestattet.



COPYRIGHT BY GAME ALCHEMISTS

Vorwort

Das Tutorial ist für den *RPG Maker VX Ace* geschrieben, kann aber ohne Probleme für den *RPG Maker MV* 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



Wo ist Laurenz denn jetzt nur wieder gelandet?
Merkwürdige Löcher im Boden ... eigenartig!

Was nun?

Schauen wir uns die Sache doch einmal genauer an.



Hier scheint ein gefährlicher Weg anzufangen – spitze Stacheln schießen aus dem Boden, bereit, Laurenz zu Schaschlik zu verarbeiten! Ein spannendes Geschicklichkeitsspiel.

Um was geht es in diesem Tutorial?

In diesem Tutorial befassen wir uns mit einem Fallen-Rätsel. Wir haben eine Fläche vor uns, gespickt mit Speeren die aus dem Boden kommen. Jedoch tun sie dies nicht willkürlich! Wer den Weg hindurch findet, hat das Rätsel gelöst und überlebt diese Aufgabe.

Wir benötigen

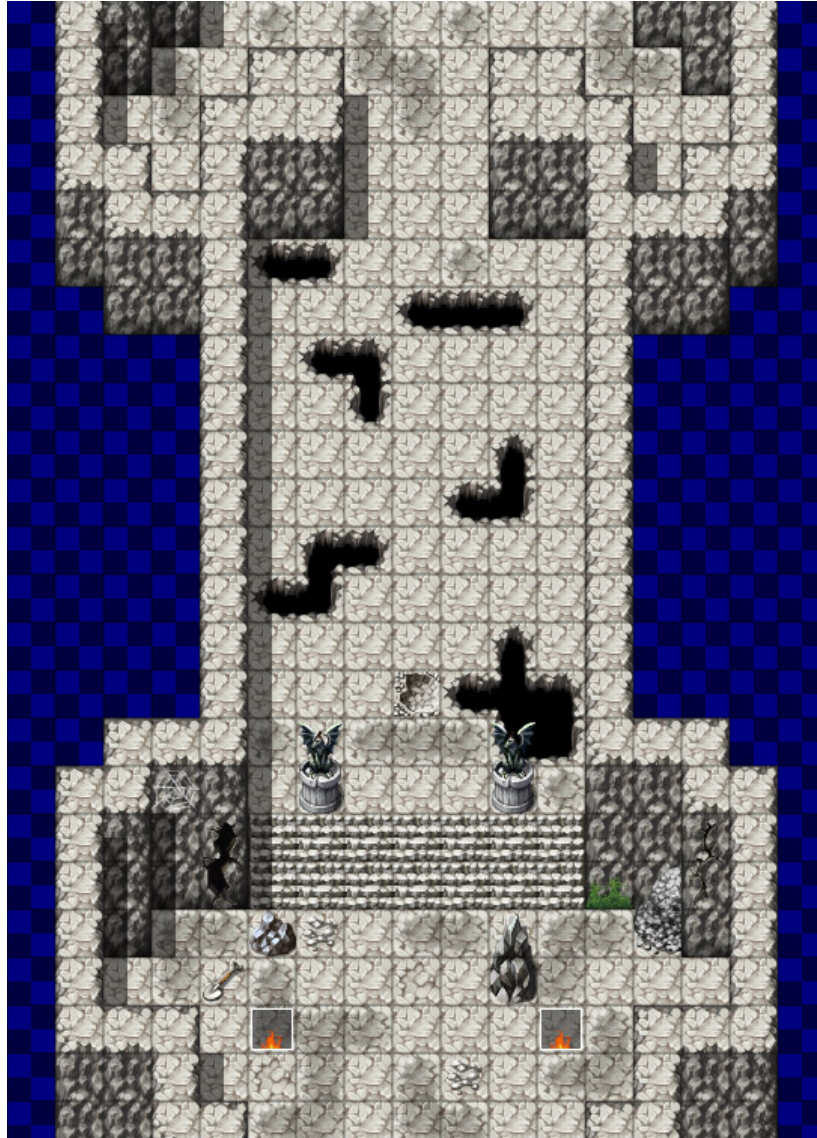
1. Gute Laune
2. Ein paar Events
3. Ein paar Event-Kentnisse

Schwierigkeitsgrad: ★★☐☐☐

Die Stacheln

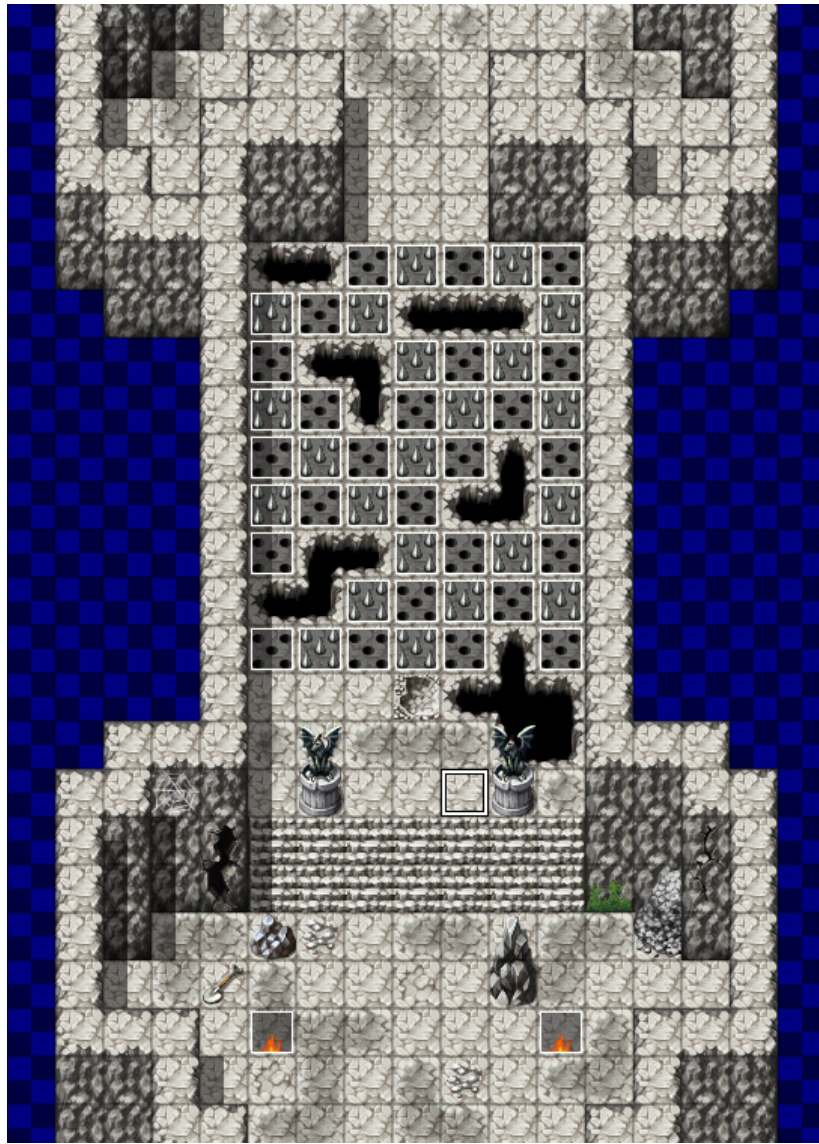
Kommen wir zum ersten Punkt des Tutorials – den Stacheln. Zuerst brauchen wir unsere Grundkarte, bei mir sieht sie so aus.

❗ Beachtet bitte, dass der Durchgang für unser Rätsel immer eine ungerade Anzahl an Feldern breit sein muss! Bei mir ist der Durchgang sieben Felder breit.



Im Bereich des Durchgangs werden wir nun unsere Stacheln einbringen.

⚠ Schon einmal zum Vormerken: Die Stacheln werden in einem Raster aufgelegt. Es gibt zwei Gruppen. Eine Stachelgruppe, die gerade eingefahren ist (nennen wir sie schwarze Gruppe) und eine Gruppe, die gerade ausgefahren ist (nennen wir sie weiße Gruppe). Zwischen dem Wechsel der Gruppen besteht eine Pause, in der sich der Spieler sicher bewegen kann. Hier eine Veranschaulichung, damit ihr seht, auf was es hinausläuft.



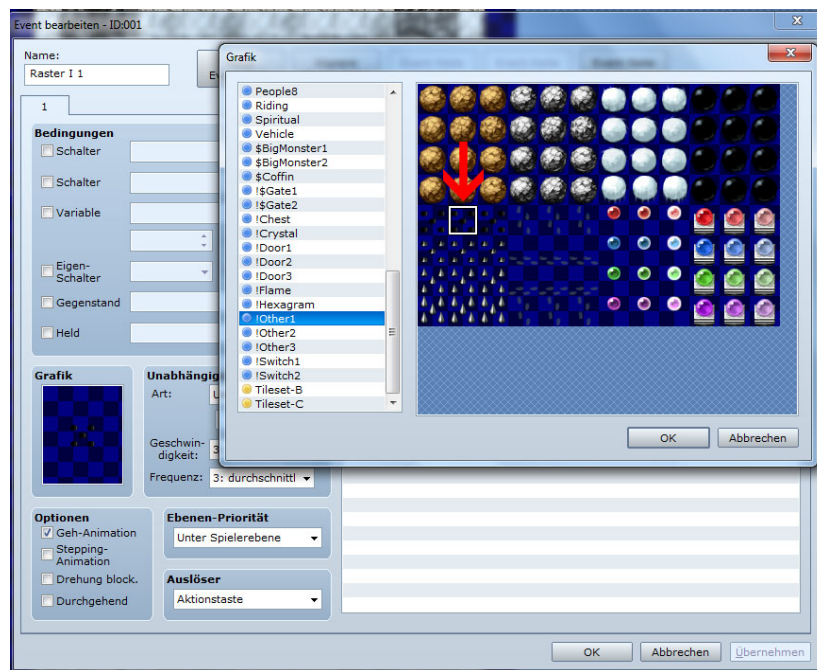
Aber nun der Reihe nach. Wir legen unser erstes Event an. Wie ihr seht, sind die eigentlichen Events, die man sieht, sehr schlicht gehalten. Wir erzeugen uns also ein Event und nennen es „**Raster I 1**“.

⚠ Zur Erklärung: **Römisch 1** ist die Gruppenzugehörigkeit. **Römisch 1** ist die schwarze Gruppe, **römisch 2**, zu der wir gleich kommen, ist die weiße Gruppe. Die Zahl dahinter symbolisiert die Nummer des Events in der Gruppe.

Die Einstellungen:

- ➡ Ebenen-Priorität: **Unter Spielerebene**
- ➡ Auslöser: **Aktionstaste**
- ➡ Als Grafik nutzen wir die vorhandenen Stacheln, die es schon im Maker gibt. Fündig werdet ihr unter „!Other1“.

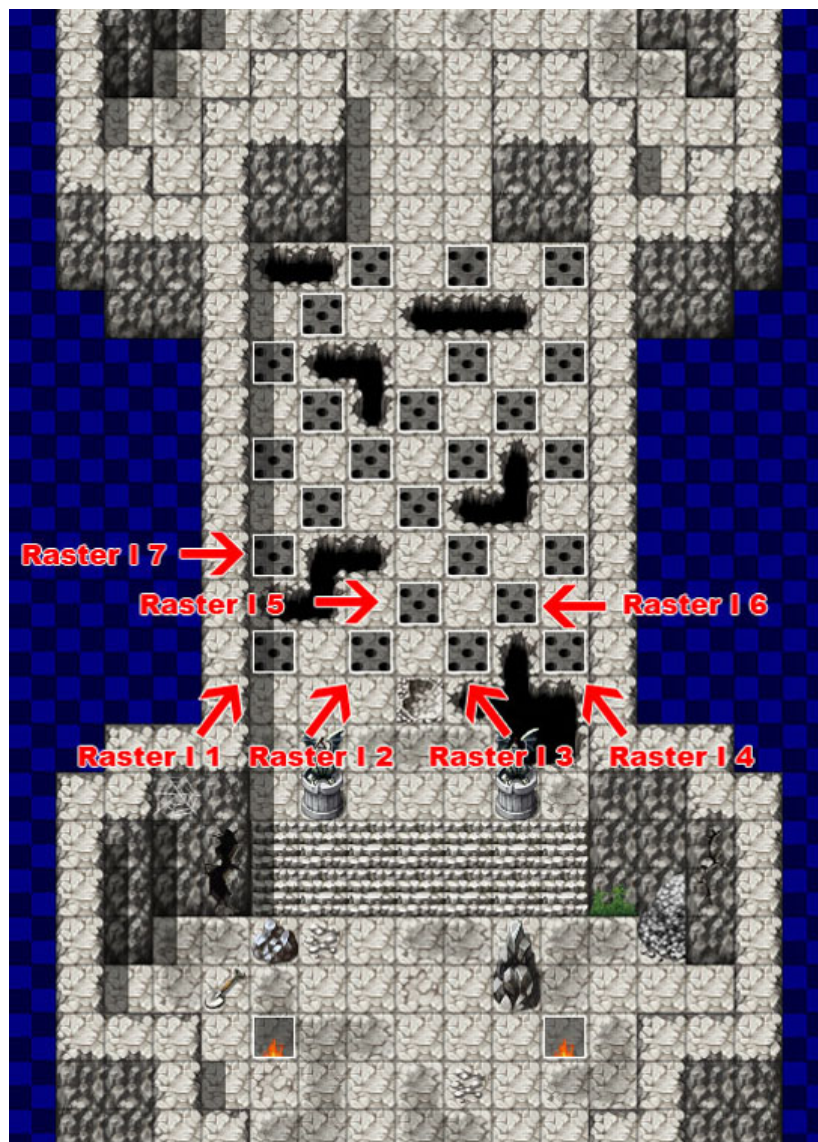
Ich habe euch auf dem Bild hervorgehoben, welche Grafik ihr nutzen müsst, nämlich die eingefahrenen Stacheln.



Klickt auf **OK** , dann sind wir hier schon fertig!

Nun müssen wir die Stacheln im Gang immer abwechselnd verteilen. Eine Stachelfalle, ein Platz frei, eine Stachelfalle, wieder ein Platz frei... Dabei müsst ihr die Löcher im Boden nicht mit Stacheln belegen. Ich hoffe, ihr könnt das Raster und den Sinn des Rasters erkennen.

❗ Wichtig: Ihr müsst die Nummern bei den Stacheln ebenfalls umbenennen, um den Überblick zu behalten. Wie ich schon erklärte, ist **römisch 1** die Gruppe und die Zahl dahinter die Nummer des Events in dieser Gruppe. Zur Erklärung habe ich euch die ersten Events im Bild benannt.

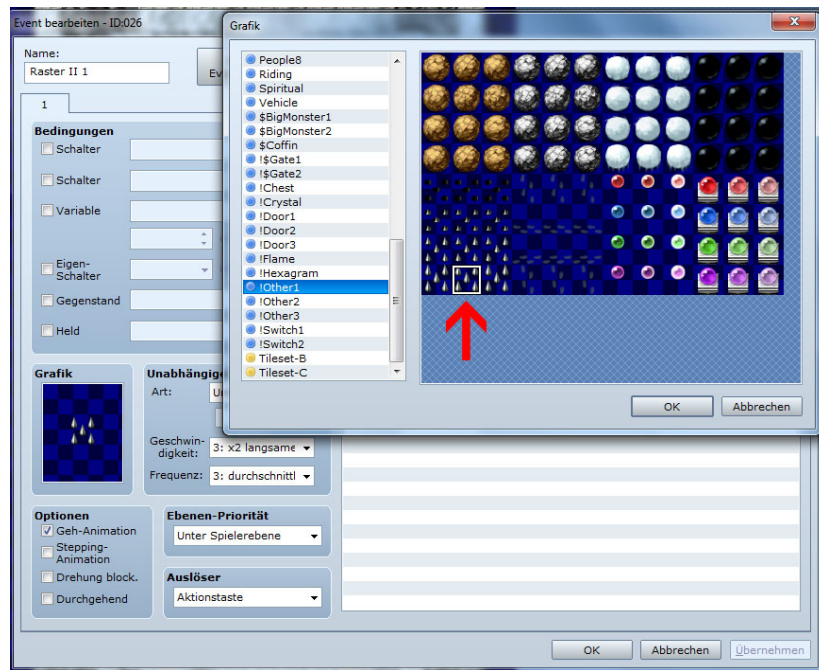


Fertig? Gut dann wären wir schon mit der schwarzen Gruppe fertig, kommen wir zur weißen Gruppe.

Am leichtesten ist es, ihr kopiert euch ein schwarzes Event und nennt es um in „**Raster II 1**“.

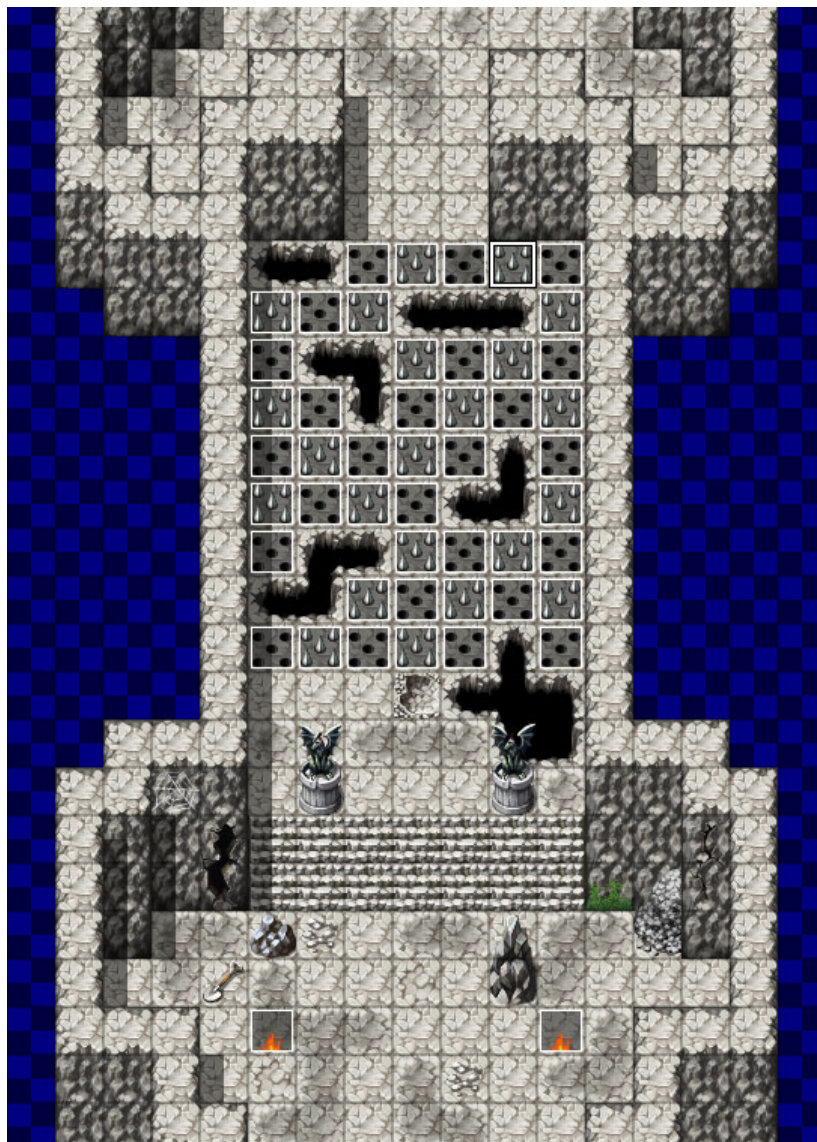
Änderung:

➡ Die Grafik wird geändert, denn die weiße Gruppe hat schon die ausgefahrenen Stacheln.



Nun machen wir das Gleiche noch einmal, wie mit der schwarzen Gruppe. Abwechselnd werden nun die Events der weißen Gruppe zwischen die schon anwesenden schwarzen Events eingefügt. Dabei wieder die Bodenöffnungen auslassen. Nun müsste es so bei euch aussehen.

❗ Nicht das Umbenennen der Events vergessen. Wir machen dies wie bei der weißen Gruppe. Das römisch II bleibt erhalten und die Zahl dahinter steigt mit jedem Event an.



Damit wäre dieser Teil geschafft. Die Stacheln sind platziert, nun kommt die Steuerung der Stacheln an die Reihe.

Die Stachelfallensteuerung

Wir brauchen ein Event, welches als paralleler Prozess fungiert und die Stacheln steuert. Ein neues Event muss also her, welches wir „**Stachelsteuerung**“ nennen.

Einzige Einstellung:

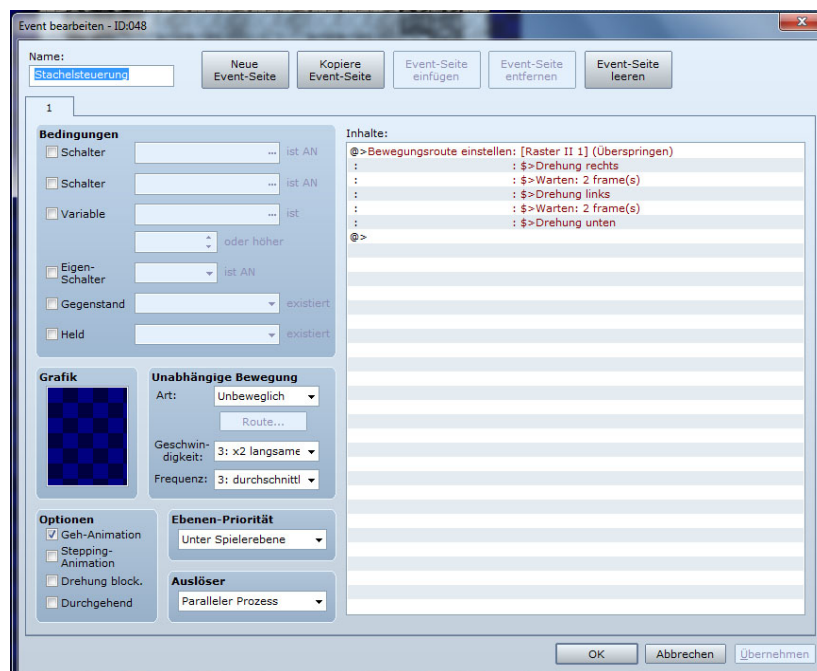
➡ Auslöser: **Paralleler Prozess**

Die Stacheln werden also dauerhaft ein und ausfahren. Fangen wir mit dem ersten Schritt an. Als erstes werden wir die Stacheln, die ausgefahren sind, sprich, die weiße Gruppe, einfahren. Dies regeln wir über Bewegungsrouten.

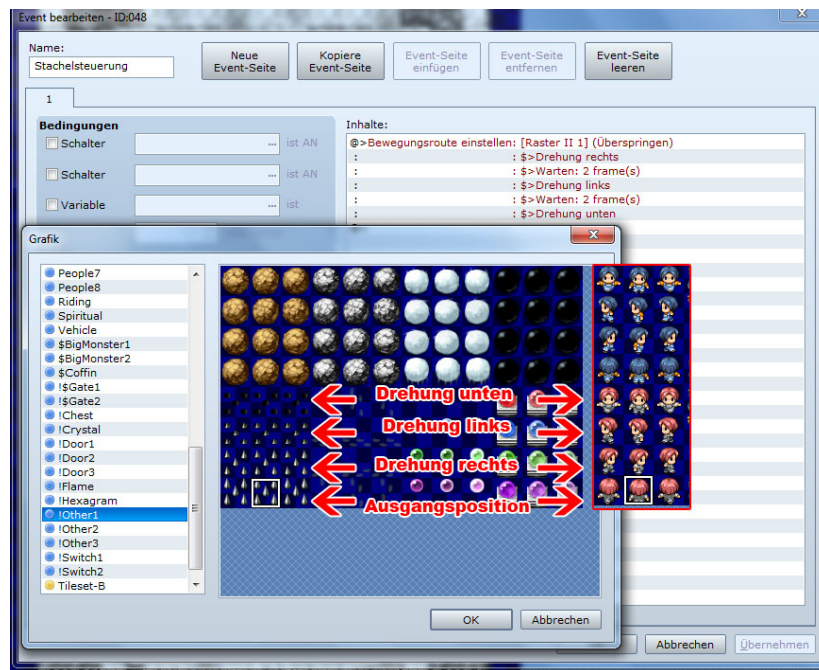
Diese lauten:

- ➡ Drehung rechts
- ➡ Warte 2 Frames
- ➡ Drehung links
- ➡ Warte 2 Frames
- ➡ Drehung unten

Hier kommt nun die gute und kluge Vorsicht zum Vorschein. Gut das wir die Stacheln mit Zahlen benannt haben, nun können wir der Reihe nach durchgehen und fangen mit „**Raster II 1**“ an.



Um besser zu verstehen, wie die Bewegungsrouten funktionieren, hier mal die Veranschaulichung. Die Grafiken der Stacheln funktionieren wie die der Charaktere.



Damit hätten wir aber erst den Stachel „Raster II 1“ eingefahren. Wir wollen aber alle der Gruppe römisch II einfahren. Darum die Bewegungsroute auch für die anderen Events der Gruppe einstellen.

Event bearbeiten - ID:049

Name:
Stachelstausung

Neue
Event-Seite

Kopiere
Event-Seite

Event-Seite
einfügen

Event-Seite
entfernen

Event-Seite
leeren

1

Bedingungen

☐ Schalter

...

ist AN

☐ Schalter

...

ist AN

☐ Variable

...

ist

☐ Eigen-Schalter

oder höher

ist AN

☐ Gegenstand

existiert

☐ Held

existiert

Grafik



Unabhängige Bewegung

Art: Unbeweglich

Route...

Geschwindigkeit: 3: x2 langsame

Frequenz: 3: durchschnittl

Optionen

☒ Geh-Animation

☐ Stepping-Animation

☐ Drehung block.

☐ Durchgehend

Ebenen-Priorität

Unter Spielerebene

Auslöser

Paralleler Prozess

Inhalte:

@>Bewegungsroute einstellen: [Raster II 1] (Überspringen)

:

:

:

:

:

@>Bewegungsroute einstellen: [Raster II 2] (Überspringen)

:

:

:

:

:

@>Bewegungsroute einstellen: [Raster II 3] (Überspringen)

:

:

:

:

:

@>Bewegungsroute einstellen: [Raster II 4] (Überspringen)

:

:

:

:

:

@>Bewegungsroute einstellen: [Raster II 5] (Überspringen)

:

:

:

:

:

@>Bewegungsroute einstellen: [Raster II 6] (Überspringen)

:

:

:

:

:

@>Bewegungsroute einstellen: [Raster II 7] (Überspringen)

:

:

:

:

:

@>Bewegungsroute einstellen: [Raster II 8] (Überspringen)

:

:

:

:

:

@>Bewegungsroute einstellen: [Raster II 9] (Überspringen)

:

:

:

:

:

@>Bewegungsroute einstellen: [Raster II 10] (Überspringen)

:

:

:

:

:

@>Bewegungsroute einstellen: [Raster II 11] (Überspringen)

:

:

:

:

:

@>Bewegungsroute einstellen: [Raster II 12] (Überspringen)

:

:

:

:

:

@>Bewegungsroute einstellen: [Raster II 13] (Überspringen)

:

:

:

:

:

@>Bewegungsroute einstellen: [Raster II 14] (Überspringen)

:

:

:

:

:

@>Bewegungsroute einstellen: [Raster II 15] (Überspringen)

:

:

:

:

:

@>Bewegungsroute einstellen: [Raster II 16] (Überspringen)

:

:

:

:

:

@>Bewegungsroute einstellen: [Raster II 17] (Überspringen)

:

:

:

:

:

@>Bewegungsroute einstellen: [Raster II 18] (Überspringen)

:

:

:

:

:

@>Bewegungsroute einstellen: [Raster II 19] (Überspringen)

:

:

:

:

:

@>Bewegungsroute einstellen: [Raster II 20] (Überspringen)

:

:

:

:

:

@>Bewegungsroute einstellen: [Raster II 21] (Überspringen)

:

:

:

:

:

@>Bewegungsroute einstellen: [Raster II 22] (Überspringen, Warten)

:

:

:

:

:

❗ Damit die Speere alle gleichzeitig aus dem Boden kommen, wenden wir einen kleinen Trick an. Alle Bewegungsrouten bekommen nur die Option „**Überspringen, wenn Bewegung nicht möglich**“, außer die letzte Bewegungsrouten, diese bekommt als Zusatz noch den Befehl „**Warten bis abgeschlossen**“. Würde dieser Wartebefehl nicht da sein, würde der Maker das gesamte Event sprichwörtlich runter rasseln, ohne zu warten, ob die Speere ein- oder ausgefahren sind. Es würde nur ein sinnloses Flackern der Speere entstehen und das wollen wir ja nicht.

So, sieht ja schon recht komplex aus oder?

Aus diesem Grund greifen wir beim nächsten Schritt in diesem Event schon ein wenig vor. Wir setzen einen Schalter mit dem Namen „**Raster II**“ und stellen diesen auf „**AUS**“.

Bei der Begründung zu diesem Schalter greife ich auch schon ein wenig vor, aber habe mich entschieden, es hier schon einzubauen, da Anfänger vielleicht bei der Länge des Events eh schon genug damit zu tun haben, den Überblick zu behalten.

Wir werden später abfragen, welche Gruppe aktiv ist – die weiße oder die schwarze Gruppe.

➡ Ist die weiße Gruppe aktiv, sprich der Schalter „**Raster I**“ AN, sind deren Speere ausgefahren. Sind die Speere eingefahren, ist die Gruppe inaktiv, der Schalter „**Raster I**“ ist **AUS**.

➡ Ist die schwarze Gruppe aktiv, sprich der Schalter „**Raster II**“ AN, sind deren Speere ausgefahren. Sind die Speere eingefahren, ist die Gruppe inaktiv, der Schalter „**Raster II**“ ist **AUS**.

Da wir die Speere gerade eingefahren haben, ist die Gruppe „**Raster II**“ also ab sofort inaktiv, der Schalter muss also **AUS** sein.

Ich hoffe, dies war einigermaßen verständlich.

➡ Als nächstes erzeugen wir uns eine Trennlinie mit der Kommentarfunktion, diese dient nur der Optik und der Auflockerung. Sie soll uns als Markierung dienen, die uns sagt, eine Rastergruppe hat gerade etwas getan.

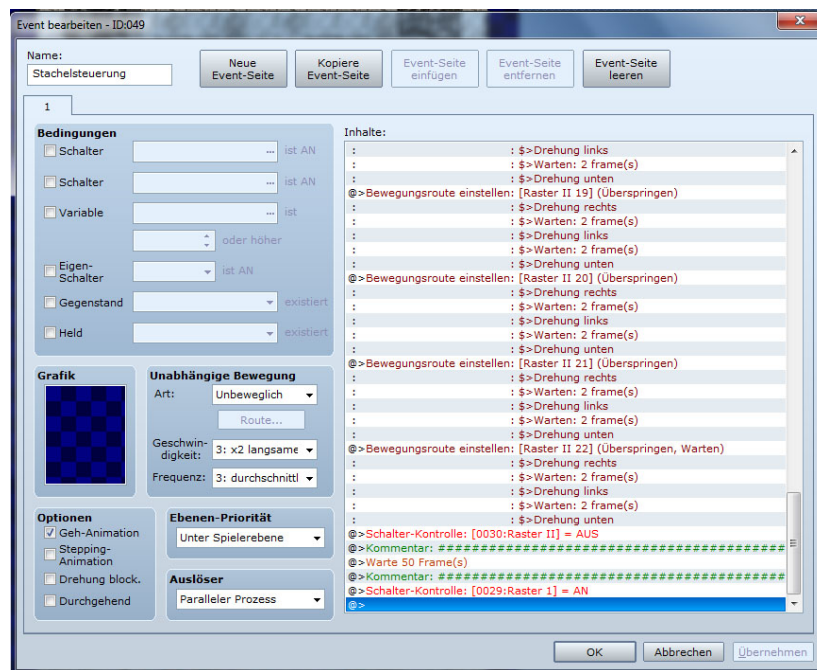
➡ Wait 50 Frames – dies ist die Wartezeit, in der der Spieler sich bewegen kann, ohne dass ihm etwas passiert, beide Stachelgruppen sind inaktiv. Über diesen Wartebefehl kann man übrigens auch den Schwierigkeitsgrad steuern. Um so höher die Framezahl, um so länger wird gewartet bis der nächste Befehl beginnt.

⚠ Es gilt = 60 Frames = 1 Sekunde

➡ Noch einmal eine Trennlinie

➡ Als letztes ein neuer Schalter nämlich „**Raster I**“ ist AN

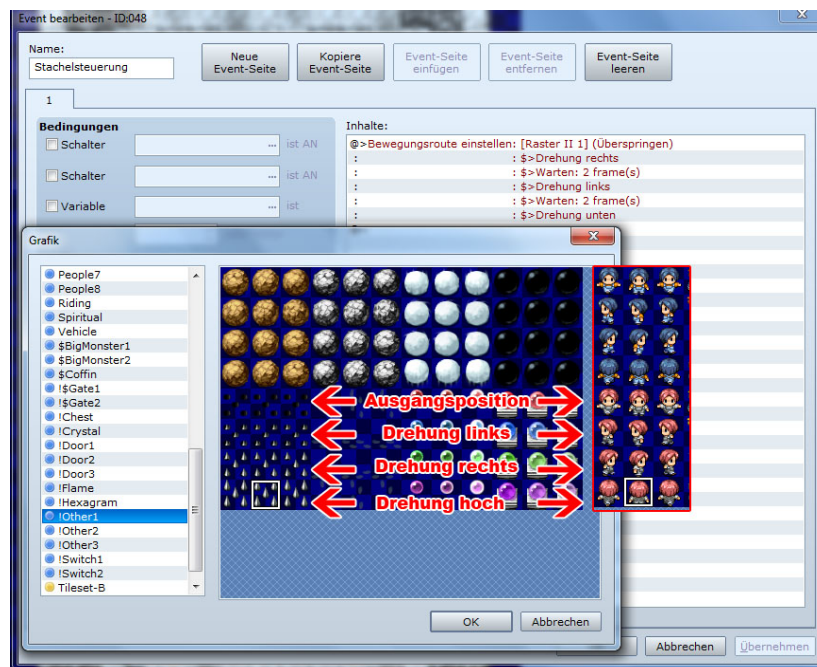
Der nächste Punkt ist nun, dass alle Stacheln von „**Raster I**“ aktiv werden, sprich aus dem Boden kommen und der Spieler verletzt werden kann.



Aber wir müssen nicht nur über den Schalter sagen, dass die „**Raster I**“ aktiv wird, wir müssen auch die Stacheln dieser Gruppe ausfahren. Dies machen wir wieder mit Bewegungsroute.

Die Bewegungsroute würde lauten:

- ➡ Drehung links
- ➡ Warte 2 Frames
- ➡ Drehung rechts
- ➡ Warte 2 Frames
- ➡ Drehung hoch



Also genau das umgekehrte, was wir vorhin machten. Wir fangen dieses Mal mit dem Event „**Raster I 1**“ an.

Wiederholt die Bewegungsroute nun bei allen Events der „**Raster I**“ und ihr müsstet dieses Ergebnis erhalten.

! Achtung: Beachtet bitte, dass wieder überall nur „Überspringen“ eingetragen ist und erst beim letzten Event noch der Zusatz „Warte“ hinzukommt!

Event bearbeiten - ID:049

Name: Racheldsteuerung

Neue Event-Seite | Kopiere Event-Seite | Event-Seite einfügen | Event-Seite entfernen | Event-Seite leeren

1

Bedingungen

☐ Schalter ist AN

☐ Schalter ist AN

☐ Variable ist


☐ oder höher

☐ Eigen-Schalter ist AN

☐ Gegenstand existiert

☐ Held existiert

Grafik



Unabhängige Bewegung

Art: Unbeweglich

Route...

Geschwindigkeit: 3: x2 langsame

Frequenz: 3: durchschnittl

Optionen

☒ Geh-Animation

☐ Stepping-Animation

☐ Drehung block.

☐ Durchgehend

Ebenen-Priorität

Unter Spielerebene

Auslöser

Paralleler Prozess

Inhalt:

③>Schalter-Kontrolle: [0030:Raster 1] = AUS

③>Kommentar: #####

③>Warte 50 Frame(s)

③>Schalter-Kontrolle: [0029:Raster 1] = AN

③>Bewegungsroute einstellen: [Raster 1 1] (Überspringen)

: :>Drehung links

: :>Warten: 2 frame(s)

: :>Drehung rechts

: :>Warten: 2 frame(s)

: :>Drehung hoch

③>Bewegungsroute einstellen: [Raster 1 2] (Überspringen)

: :>Drehung links

: :>Warten: 2 frame(s)

: :>Drehung rechts

: :>Warten: 2 frame(s)

: :>Drehung hoch

③>Bewegungsroute einstellen: [Raster 1 3] (Überspringen)

: :>Drehung links

: :>Warten: 2 frame(s)

: :>Drehung rechts

: :>Warten: 2 frame(s)

: :>Drehung hoch

③>Bewegungsroute einstellen: [Raster 1 4] (Überspringen)

: :>Drehung links

: :>Warten: 2 frame(s)

: :>Drehung rechts

: :>Warten: 2 frame(s)

: :>Drehung hoch

③>Bewegungsroute einstellen: [Raster 1 5] (Überspringen)

: :>Drehung links

: :>Warten: 2 frame(s)

: :>Drehung rechts

: :>Drehung hoch

③>Bewegungsroute einstellen: [Raster 1 6] (Überspringen)

: :>Drehung links

: :>Warten: 2 frame(s)

: :>Drehung rechts

: :>Warten: 2 frame(s)

: :>Drehung hoch

③>Bewegungsroute einstellen: [Raster 1 7] (Überspringen)

: :>Drehung links

: :>Warten: 2 frame(s)

: :>Drehung rechts

: :>Warten: 2 frame(s)

: :>Drehung hoch

③>Bewegungsroute einstellen: [Raster 1 8] (Überspringen)

: :>Drehung links

: :>Warten: 2 frame(s)

: :>Drehung rechts

: :>Warten: 2 frame(s)

: :>Drehung hoch

③>Bewegungsroute einstellen: [Raster 1 9] (Überspringen)

: :>Drehung links

: :>Warten: 2 frame(s)

: :>Drehung rechts

: :>Warten: 2 frame(s)

: :>Drehung hoch

③>Bewegungsroute einstellen: [Raster 1 10] (Überspringen)

: :>Drehung links

: :>Warten: 2 frame(s)

: :>Drehung rechts

: :>Warten: 2 frame(s)

: :>Drehung hoch

③>Bewegungsroute einstellen: [Raster 1 11] (Überspringen)

: :>Drehung links

: :>Warten: 2 frame(s)

: :>Drehung rechts

: :>Warten: 2 frame(s)

: :>Drehung hoch

③>Bewegungsroute einstellen: [Raster 1 12] (Überspringen)

: :>Drehung links

: :>Warten: 2 frame(s)

: :>Drehung rechts

: :>Warten: 2 frame(s)

: :>Drehung hoch

③>Bewegungsroute einstellen: [Raster 1 13] (Überspringen)

: :>Drehung links

: :>Warten: 2 frame(s)

: :>Drehung rechts

: :>Warten: 2 frame(s)

: :>Drehung hoch

③>Bewegungsroute einstellen: [Raster 1 14] (Überspringen)

: :>Drehung links

: :>Warten: 2 frame(s)

: :>Drehung rechts

: :>Warten: 2 frame(s)

: :>Drehung hoch

③>Bewegungsroute einstellen: [Raster 1 15] (Überspringen)

: :>Drehung links

: :>Warten: 2 frame(s)

: :>Drehung rechts

: :>Warten: 2 frame(s)

: :>Drehung hoch

③>Bewegungsroute einstellen: [Raster 1 16] (Überspringen)

: :>Drehung links

: :>Warten: 2 frame(s)

: :>Drehung rechts

: :>Warten: 2 frame(s)

: :>Drehung hoch

③>Bewegungsroute einstellen: [Raster 1 17] (Überspringen)

: :>Drehung links

: :>Warten: 2 frame(s)

: :>Drehung rechts

: :>Warten: 2 frame(s)

: :>Drehung hoch

③>Bewegungsroute einstellen: [Raster 1 18] (Überspringen)

: :>Drehung links

: :>Warten: 2 frame(s)

: :>Drehung rechts

: :>Warten: 2 frame(s)

: :>Drehung hoch

③>Bewegungsroute einstellen: [Raster 1 19] (Überspringen)

: :>Drehung links

: :>Warten: 2 frame(s)

: :>Drehung rechts

: :>Warten: 2 frame(s)

: :>Drehung hoch

③>Bewegungsroute einstellen: [Raster 1 20] (Überspringen)

: :>Drehung links

: :>Warten: 2 frame(s)

: :>Drehung rechts

: :>Warten: 2 frame(s)

: :>Drehung hoch

③>Bewegungsroute einstellen: [Raster 1 21] (Überspringen)

: :>Drehung links

: :>Warten: 2 frame(s)

: :>Drehung rechts

: :>Warten: 2 frame(s)

: :>Drehung hoch

③>Bewegungsroute einstellen: [Raster 1 22] (Überspringen)

: :>Drehung links

: :>Warten: 2 frame(s)

: :>Drehung rechts

: :>Warten: 2 frame(s)

: :>Drehung hoch

③>Bewegungsroute einstellen: [Raster 1 23] (Überspringen)

: :>Drehung links

: :>Warten: 2 frame(s)

: :>Drehung rechts

: :>Warten: 2 frame(s)

: :>Drehung hoch

③>Bewegungsroute einstellen: [Raster 1 24] (Überspringen)

: :>Drehung links

: :>Warten: 2 frame(s)

: :>Drehung rechts

: :>Warten: 2 frame(s)

: :>Drehung hoch

③>Bewegungsroute einstellen: [Raster 1 25] (Überspringen, Warten)

: :>Drehung links

: :>Warten: 2 frame(s)

: :>Drehung rechts

: :>Warten: 2 frame(s)

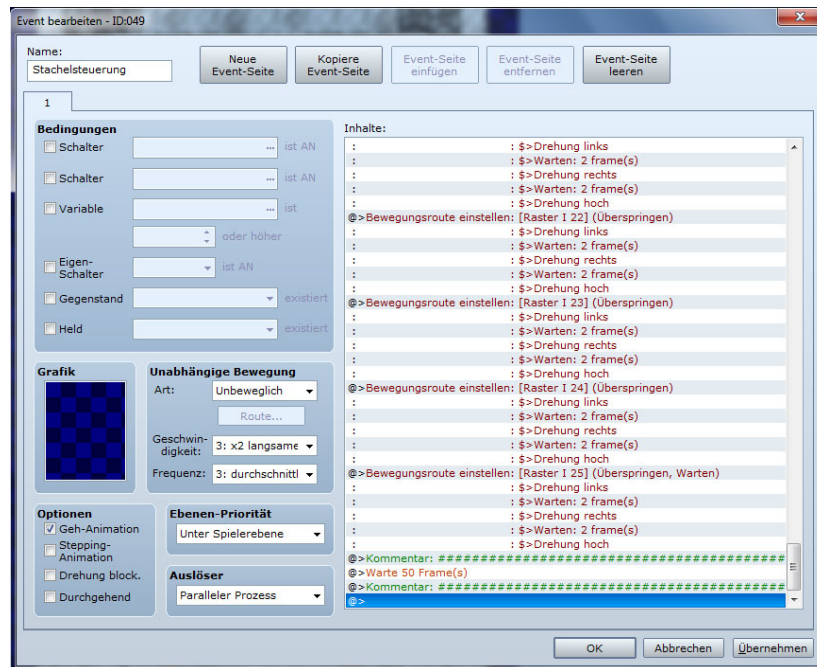
: :>Drehung hoch

③>

Die Hälfte haben wir schon!

Die Stacheln der schwarzen Gruppe mit dem Namen „**Raster I**“ sind nun ausgefahren. Hier verweilen sie auch kurz.

- ➡ Ein kleiner Kommentar mit einer Trennlinie für die bessere Übersicht
- ➡ Warte 50 Frames – so lange bleiben die Stacheln draußen
- ➡ Noch ein Kommentar mit Trennlinie



Aber irgendwann müssen die Stacheln ja mal wieder zurück dahin, wo sie her kamen. Wir schieben also die Stacheln zurück in die Erde – und wie machen wir dies? Mit Bewegungsrouten! Dazu nutzen wir wieder die Bewegungsrouten, die wir als erstes genutzt haben.

- ➡ Drehung rechts
- ➡ Warte 2 Frames
- ➡ Drehung links
- ➡ Warte 2 Frames
- ➡ Drehung unten

Dieses Mal wenden wir diese Bewegungsroute aber auf alle Events des „**Rasters I**“ an.

Event bearbeiten - ID049

Name:
Stachelsteuerung

Neue
Event-Seite

Kopiere
Event-Seite

Event-Seite
einfügen

Event-Seite
entfernen

Event-Seite
leeren

1

Bedingungen

☐ Schalter

ist AN

☐ Schalter

ist AN

☐ Variable

ist

oder höher

☐ Eigen-Schalter

ist AN

☐ Gegenstand

existiert

☐ Held

existiert

Grafik

Unabhängige Bewegung

Art: Unbeweglich

Route...

Geschwindigkeit: 3: x2 langsame

Frequenz: 3: durchschnittl

Optionen

☒ Geh-Animation

☐ Stepping-Animation

☐ Drehung block.

☐ Durchgehend

Ebenen-Priorität

Unter Spielerebene

Auslöser

Paralleler Prozess

Inhalte:

➤>Kommentar: #####

➤>Warte 50 Frame(s)

➤>Kommentar: #####

➤>Bewegungsroute einstellen: [Raster I 1] (Überspringen)

: :>Drehung rechts

: :>Warten: 2 frame(s)

: :>Drehung links

: :>Warten: 2 frame(s)

: :>Drehung unten

➤>Bewegungsroute einstellen: [Raster I 2] (Überspringen)

: :>Drehung rechts

: :>Warten: 2 frame(s)

: :>Drehung links

: :>Warten: 2 frame(s)

: :>Drehung unten

➤>Bewegungsroute einstellen: [Raster I 3] (Überspringen)

: :>Drehung rechts

: :>Warten: 2 frame(s)

: :>Drehung links

: :>Warten: 2 frame(s)

: :>Drehung unten

➤>Bewegungsroute einstellen: [Raster I 4] (Überspringen)

: :>Drehung rechts

: :>Warten: 2 frame(s)

: :>Drehung links

: :>Warten: 2 frame(s)

: :>Drehung unten

➤>Bewegungsroute einstellen: [Raster I 5] (Überspringen)

: :>Drehung rechts

: :>Warten: 2 frame(s)

: :>Drehung links

: :>Warten: 2 frame(s)

: :>Drehung unten

➤>Bewegungsroute einstellen: [Raster I 6] (Überspringen)

: :>Drehung rechts

: :>Warten: 2 frame(s)

: :>Drehung links

: :>Warten: 2 frame(s)

: :>Drehung unten

➤>Bewegungsroute einstellen: [Raster I 7] (Überspringen)

: :>Drehung rechts

: :>Warten: 2 frame(s)

: :>Drehung links

: :>Warten: 2 frame(s)

: :>Drehung unten

➤>Bewegungsroute einstellen: [Raster I 8] (Überspringen)

: :>Drehung rechts

: :>Warten: 2 frame(s)

: :>Drehung links

: :>Warten: 2 frame(s)

: :>Drehung unten

➤>Bewegungsroute einstellen: [Raster I 9] (Überspringen)

: :>Drehung rechts

: :>Warten: 2 frame(s)

: :>Drehung links

: :>Warten: 2 frame(s)

: :>Drehung unten

➤>Bewegungsroute einstellen: [Raster I 10] (Überspringen)

: :>Drehung rechts

: :>Warten: 2 frame(s)

: :>Drehung links

: :>Warten: 2 frame(s)

: :>Drehung unten

➤>Bewegungsroute einstellen: [Raster I 11] (Überspringen)

: :>Drehung rechts

: :>Warten: 2 frame(s)

: :>Drehung links

: :>Warten: 2 frame(s)

: :>Drehung unten

➤>Bewegungsroute einstellen: [Raster I 12] (Überspringen)

: :>Drehung rechts

: :>Warten: 2 frame(s)

: :>Drehung links

: :>Warten: 2 frame(s)

: :>Drehung unten

➤>Bewegungsroute einstellen: [Raster I 13] (Überspringen)

: :>Drehung rechts

: :>Warten: 2 frame(s)

: :>Drehung links

: :>Warten: 2 frame(s)

: :>Drehung unten

➤>Bewegungsroute einstellen: [Raster I 14] (Überspringen)

: :>Drehung rechts

: :>Warten: 2 frame(s)

: :>Drehung links

: :>Warten: 2 frame(s)

: :>Drehung unten

➤>Bewegungsroute einstellen: [Raster I 15] (Überspringen)

: :>Drehung rechts

: :>Warten: 2 frame(s)

: :>Drehung links

: :>Warten: 2 frame(s)

: :>Drehung unten

➤>Bewegungsroute einstellen: [Raster I 16] (Überspringen)

: :>Drehung rechts

: :>Warten: 2 frame(s)

: :>Drehung links

: :>Warten: 2 frame(s)

: :>Drehung unten

➤>Bewegungsroute einstellen: [Raster I 17] (Überspringen)

: :>Drehung rechts

: :>Warten: 2 frame(s)

: :>Drehung links

: :>Warten: 2 frame(s)

: :>Drehung unten

➤>Bewegungsroute einstellen: [Raster I 18] (Überspringen)

: :>Drehung rechts

: :>Warten: 2 frame(s)

: :>Drehung links

: :>Warten: 2 frame(s)

: :>Drehung unten

➤>Bewegungsroute einstellen: [Raster I 19] (Überspringen)

: :>Drehung rechts

: :>Warten: 2 frame(s)

: :>Drehung links

: :>Warten: 2 frame(s)

: :>Drehung unten

➤>Bewegungsroute einstellen: [Raster I 20] (Überspringen)

: :>Drehung rechts

: :>Warten: 2 frame(s)

: :>Drehung links

: :>Warten: 2 frame(s)

: :>Drehung unten

➤>Bewegungsroute einstellen: [Raster I 21] (Überspringen)

: :>Drehung rechts

: :>Warten: 2 frame(s)

: :>Drehung links

: :>Warten: 2 frame(s)

: :>Drehung unten

➤>Bewegungsroute einstellen: [Raster I 22] (Überspringen)

: :>Drehung rechts

: :>Warten: 2 frame(s)

: :>Drehung links

: :>Warten: 2 frame(s)

: :>Drehung unten

➤>Bewegungsroute einstellen: [Raster I 23] (Überspringen)

: :>Drehung rechts

: :>Warten: 2 frame(s)

: :>Drehung links

: :>Warten: 2 frame(s)

: :>Drehung unten

➤>Bewegungsroute einstellen: [Raster I 24] (Überspringen)

: :>Drehung rechts

: :>Warten: 2 frame(s)

: :>Drehung links

: :>Warten: 2 frame(s)

: :>Drehung unten

➤>Bewegungsroute einstellen: [Raster I 25] (Überspringen, Warten)

: :>Drehung rechts

: :>Warten: 2 frame(s)

: :>Drehung links

: :>Warten: 2 frame(s)

: :>Drehung unten

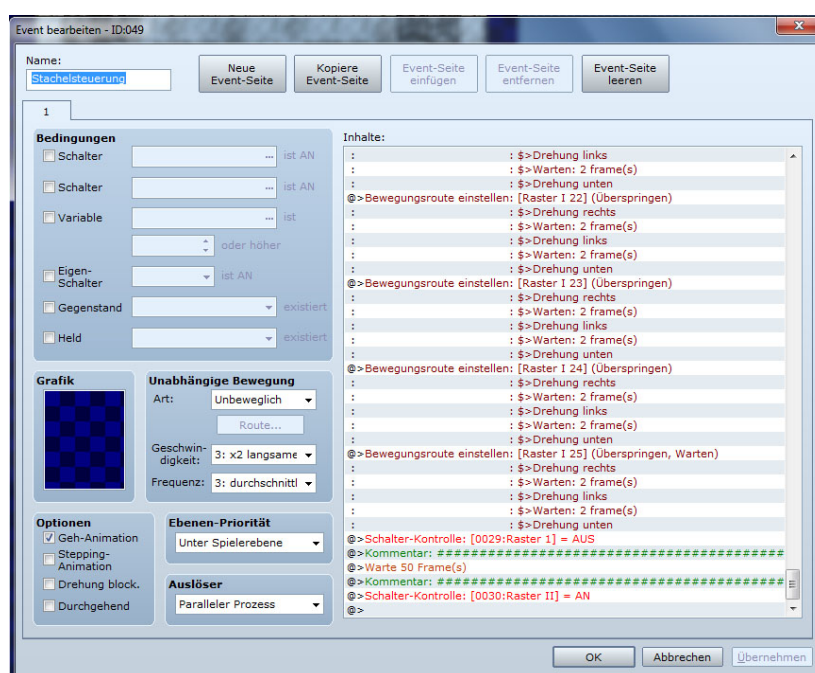
➤>

! Achtung: Beachtet bitte, dass wieder überall nur „Überspringen“ eingetragen ist und erst beim letzten Event noch der Zusatz „Warte“ hinzukommt!

Damit wären wir fast fertig, ihr habt es gleich geschafft!

Da die Stacheln eingefahren sind, geht für den Spieler auch keine Gefahr mehr von ihnen aus, darum kommen nun wieder ein paar Schalter auf uns zu.

- ➔ Schalter „**Raster I**“ wird ausgeschaltet, da die Speere ja eingefahren, sprich inaktiv sind
- ➔ Wieder mal eine Trennlinie
- ➔ Warte 50 Frames – hier kann sich der Spieler wieder sicher bewegen
- ➔ Noch eine Trennlinie
- ➔ Schalter „**Raster II**“ wird angeschaltet, da diese Stacheln nun wieder ausgefahren werden.



Event bearbeiten - ID:049

Name:
Stachelsteuerung

Neue
Event-Seite

Kopiere
Event-Seite

Event-Seite
einfügen

Event-Seite
entfernen

Event-Seite
leeren

1

Bedingungen

☐ Schalter

...

ist AN

☐ Schalter

...

ist AN

☐ Variable

...

ist

oder höher

...

ist AN

☐ Eigen-Schalter

...

ist AN

☐ Gegenstand

...

existiert

☐ Held

...

existiert

Grafik

Unabhängige Bewegung

Art: Unbeweglich

Route...

Geschwindigkeit: 3: x2 langsame

Frequenz: 3: durchschnittl

Optionen

☒ Geh-Animation

☐ Stepping-Animation

☐ Drehung block.

☐ Durchgehend

Ebenen-Priorität

Unter Spielerebene

Auslöser

Paralleler Prozess

Inhalte:

@>Schalter-Kontrolle: [0029:Raster 1] = AUS

@>Kommentar: #####

@>Warte 50 Frame(s)

@>Kommentar: #####

@>Schalter-Kontrolle: [0030:Raster II] = AN

@>Bewegungsroute einstellen: [Raster II 1] (Überspringen)

: \$>Drehung links

: \$>Warten: 2 frame(s)

: \$>Drehung rechts

: \$>Warten: 2 frame(s)

: \$>Drehung hoch

@>Bewegungsroute einstellen: [Raster II 2] (Überspringen)

: \$>Drehung links

: \$>Warten: 2 frame(s)

: \$>Drehung rechts

: \$>Warten: 2 frame(s)

: \$>Drehung hoch

@>Bewegungsroute einstellen: [Raster II 3] (Überspringen)

: \$>Drehung links

: \$>Warten: 2 frame(s)

: \$>Drehung rechts

: \$>Warten: 2 frame(s)

: \$>Drehung hoch

@>Bewegungsroute einstellen: [Raster II 4] (Überspringen)

: \$>Drehung links

: \$>Warten: 2 frame(s)

: \$>Drehung rechts

: \$>Warten: 2 frame(s)

: \$>Drehung hoch

@>Bewegungsroute einstellen: [Raster II 5] (Überspringen)

: \$>Drehung links

: \$>Warten: 2 frame(s)

: \$>Drehung rechts

: \$>Warten: 2 frame(s)

: \$>Drehung hoch

@>Bewegungsroute einstellen: [Raster II 6] (Überspringen)

: \$>Drehung links

: \$>Warten: 2 frame(s)

: \$>Drehung rechts

: \$>Warten: 2 frame(s)

: \$>Drehung hoch

@>Bewegungsroute einstellen: [Raster II 7] (Überspringen)

: \$>Drehung links

: \$>Warten: 2 frame(s)

: \$>Drehung rechts

: \$>Warten: 2 frame(s)

: \$>Drehung hoch

@>Bewegungsroute einstellen: [Raster II 8] (Überspringen)

: \$>Drehung links

: \$>Warten: 2 frame(s)

: \$>Drehung rechts

: \$>Warten: 2 frame(s)

: \$>Drehung hoch

@>Bewegungsroute einstellen: [Raster II 9] (Überspringen)

: \$>Drehung links

: \$>Warten: 2 frame(s)

: \$>Drehung rechts

: \$>Warten: 2 frame(s)

: \$>Drehung hoch

@>Bewegungsroute einstellen: [Raster II 10] (Überspringen)

: \$>Drehung links

: \$>Warten: 2 frame(s)

: \$>Drehung rechts

: \$>Warten: 2 frame(s)

: \$>Drehung hoch

@>Bewegungsroute einstellen: [Raster II 11] (Überspringen)

: \$>Drehung links

: \$>Warten: 2 frame(s)

: \$>Drehung rechts

: \$>Warten: 2 frame(s)

: \$>Drehung hoch

@>Bewegungsroute einstellen: [Raster II 12] (Überspringen)

: \$>Drehung links

: \$>Warten: 2 frame(s)

: \$>Drehung rechts

: \$>Warten: 2 frame(s)

: \$>Drehung hoch

@>Bewegungsroute einstellen: [Raster II 13] (Überspringen)

: \$>Drehung links

: \$>Warten: 2 frame(s)

: \$>Drehung rechts

: \$>Warten: 2 frame(s)

: \$>Drehung hoch

@>Bewegungsroute einstellen: [Raster II 14] (Überspringen)

: \$>Drehung links

: \$>Warten: 2 frame(s)

: \$>Drehung rechts

: \$>Warten: 2 frame(s)

: \$>Drehung hoch

@>Bewegungsroute einstellen: [Raster II 15] (Überspringen)

: \$>Drehung links

: \$>Warten: 2 frame(s)

: \$>Drehung rechts

: \$>Warten: 2 frame(s)

: \$>Drehung hoch

@>Bewegungsroute einstellen: [Raster II 16] (Überspringen)

: \$>Drehung links

: \$>Warten: 2 frame(s)

: \$>Drehung rechts

: \$>Warten: 2 frame(s)

: \$>Drehung hoch

@>Bewegungsroute einstellen: [Raster II 17] (Überspringen)

: \$>Drehung links

: \$>Warten: 2 frame(s)

: \$>Drehung rechts

: \$>Warten: 2 frame(s)

: \$>Drehung hoch

@>Bewegungsroute einstellen: [Raster II 18] (Überspringen)

: \$>Drehung links

: \$>Warten: 2 frame(s)

: \$>Drehung rechts

: \$>Warten: 2 frame(s)

: \$>Drehung hoch

@>Bewegungsroute einstellen: [Raster II 19] (Überspringen)

: \$>Drehung links

: \$>Warten: 2 frame(s)

: \$>Drehung rechts

: \$>Warten: 2 frame(s)

: \$>Drehung hoch

@>Bewegungsroute einstellen: [Raster II 20] (Überspringen)

: \$>Drehung links

: \$>Warten: 2 frame(s)

: \$>Drehung rechts

: \$>Warten: 2 frame(s)

: \$>Drehung hoch

@>Bewegungsroute einstellen: [Raster II 21] (Überspringen)

: \$>Drehung links

: \$>Warten: 2 frame(s)

: \$>Drehung rechts

: \$>Warten: 2 frame(s)

: \$>Drehung hoch

@>Bewegungsroute einstellen: [Raster II 22] (Überspringen, Warten)

: \$>Drehung links

: \$>Warten: 2 frame(s)

: \$>Drehung rechts

: \$>Warten: 2 frame(s)

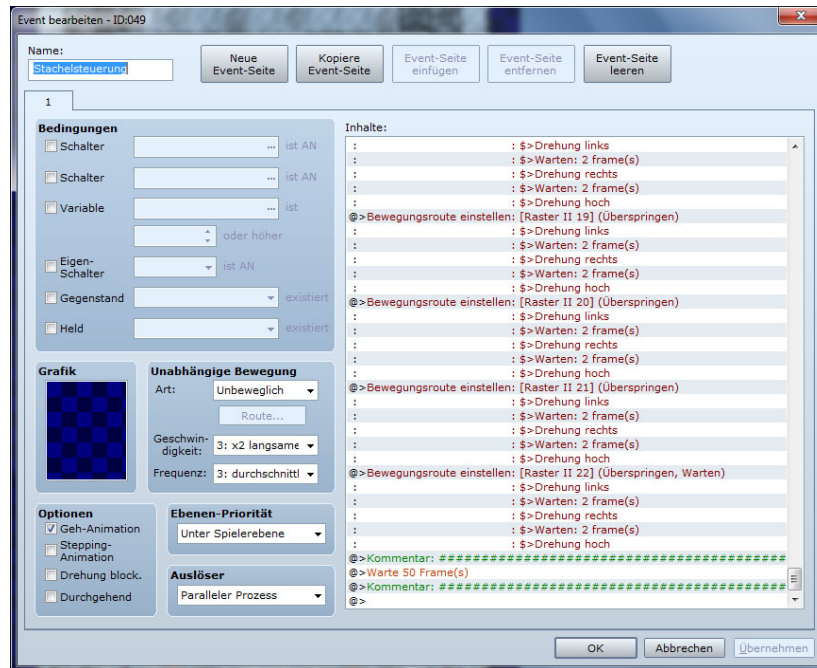
: \$>Drehung hoch

@>

II (17)

Damit wären wir bei diesem Event fast fertig. Das Einzige, was nun noch kommt, ist:

- ➡ Wieder mal eine Trennlinie
- ➡ Warte 50 Frames – hier kann sich der Spieler wieder bewegen
- ➡ Noch eine Trennlinie



Fertig! Nach dem Warte-Befehl fängt das Event bedingt durch den parallelen Prozess wieder von vorn an.

Was wir jetzt getan haben, ist kurz zusammengefasst:

- ➡ „Raster II“ eingefahren
- ➡ 50 Frames Wartezeit eingesetzt in der sich der Spieler bewegen kann
- ➡ „Raster I“ ausgefahren
- ➡ 50 Frames Wartezeit in der die Stacheln von „Raster I“ draußen verweilen
- ➡ „Raster I“ eingefahren
- ➡ 50 Frames Wartezeit eingesetzt in der sich der Spieler bewegen kann
- ➡ „Raster II“ ausgefahren
- ➡ 50 Frames Wartezeit in der die Stacheln von „Raster 2“ draußen verweilen

Wir haben also für die **Rastergruppe 1 und 2** einen kompletten Durchgang erstellt und dem Maker gesagt, wie sie sich bewegen sollen. Dank der Wartebefehle am Ende jedes Durchgangs haben wir die Garantie, dass die Gruppe immer gemeinsam ein- und ausfährt.

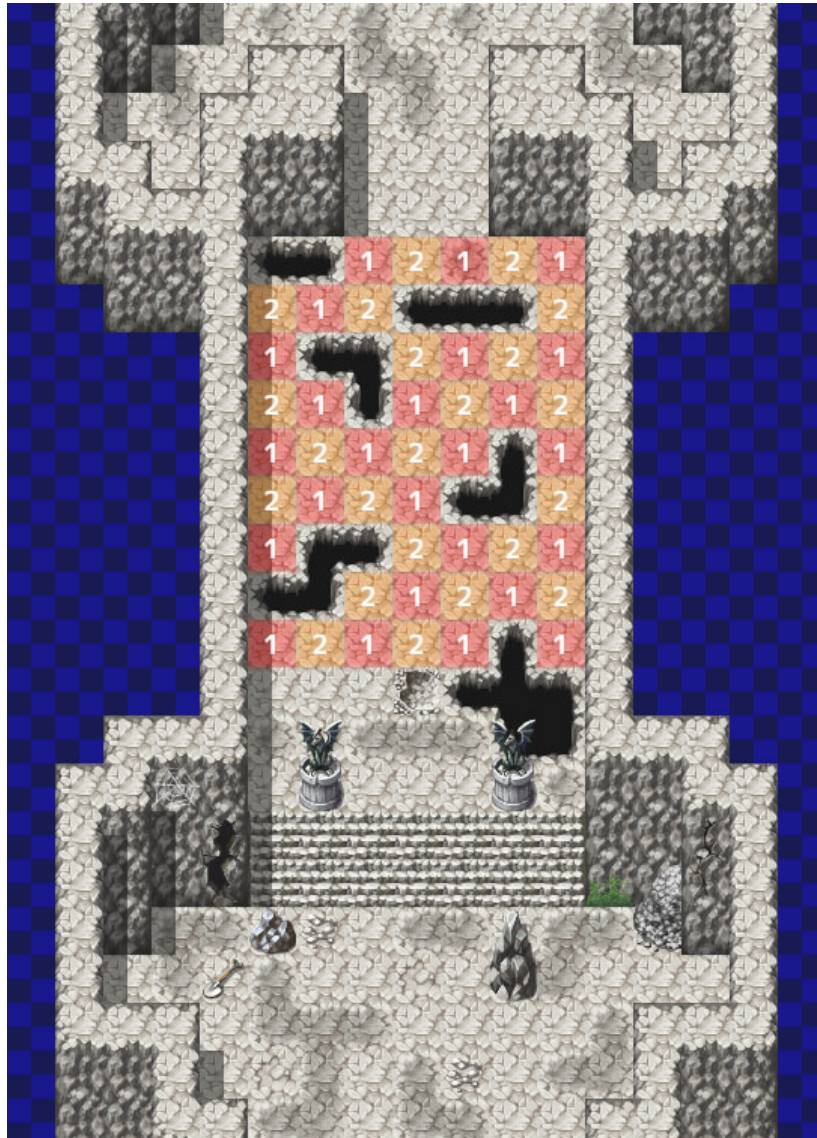
Der Schadensmanager

Tja, jetzt fahren zwar unsere Stacheln schön ein und aus, das bringt uns aber nicht viel. Wir müssen ja noch festlegen, wann der Spieler wo hin laufen kann, ohne das ihn etwas piekst und tötet bzw. er verletzt wird. Aus diesem Grund nutzen wir Regions IDs. Einige werden sich nun fragen: Was sind Region IDs? Das ist recht einfach zu erklären.

Zwei der eingekreisten Symbole dürften euch bekannt vor kommen – „Karte“, „Events“ und gleich daneben ist „Regions“. Klickt mal drauf. Die Karte wird nun etwas blass und die Tilesets sind weg. Stattdessen sind da Nummern. Dies sind die Regions IDs. Klickt man sie an und „zeichnet“ damit auf der Karte herum, legt man unsichtbare Markierungen aus. Diese nutzen wir nun zur Markierung, damit der Maker weiß, wo unser kleiner Protagonist steht.

- ➡ Auf alle Flächen, wo Events des „Rasters I“ drauf sind, legen wir die **Regions ID 1**.
- ➡ Auf alle Flächen, wo Events des „Rasters II“ drauf sind, legen wir die **Regions ID 2**.

Nun müsste es so bei euch aussehen:



Regions IDs gesetzt? Dann geht es nun zu den Events zurück.

Wir brauchen ein neues Event mit dem Namen „**Schadensmanager**“.

Einzige Einstellung:

➡ Auslöser: **Paralleler Prozess**

Nun zum Inhalt:

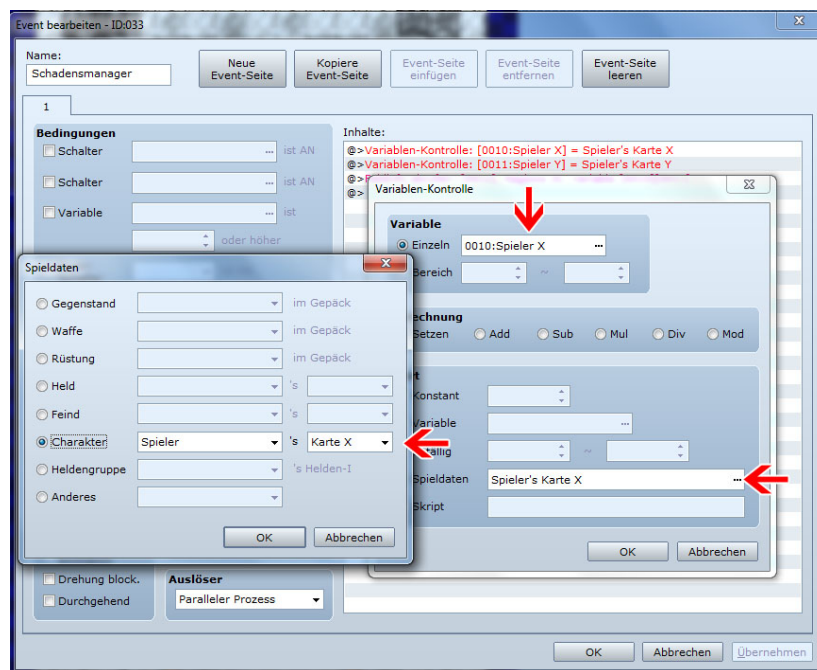
➡ Warte = **2 Frames** (wieder eine Sicherung gegen Lags)

➡ Variable Koordinate **Spieler X**

➡ Variable Koordinate **Spieler Y**

➡ Variable **Feld ID**

Dies sind die Variablen, die wir brauchen. Spieler X und Y einzustellen ist recht einfach. Am besten zu erklären mit einem Bild, hier als Beispiel für die X Koordinate. Y Funktioniert genauso:



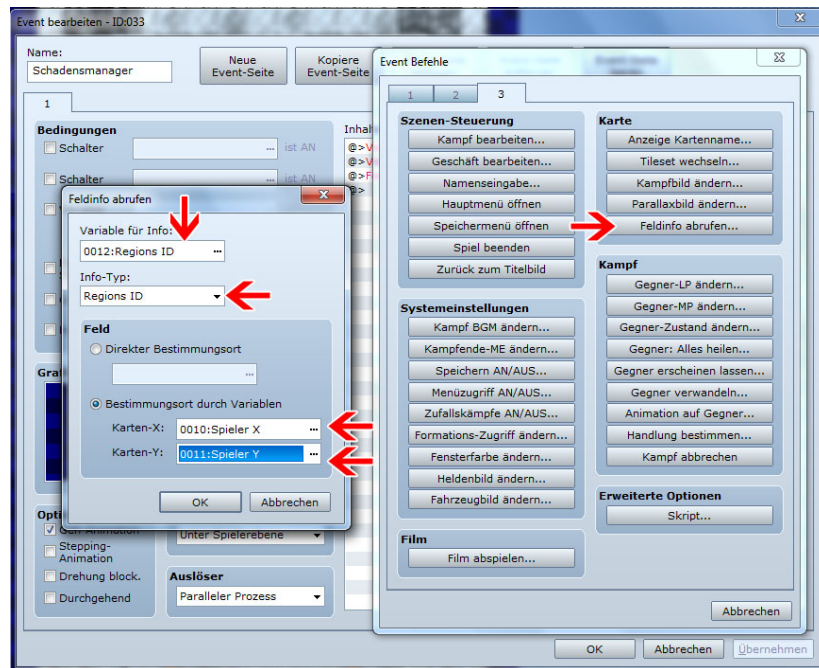
Feld ID Infos ruft man etwas anders ab. Feld ID Info ist auf Seite drei der Eventbefehle zu finden.

➡ Wir setzen den Variablennamen ein

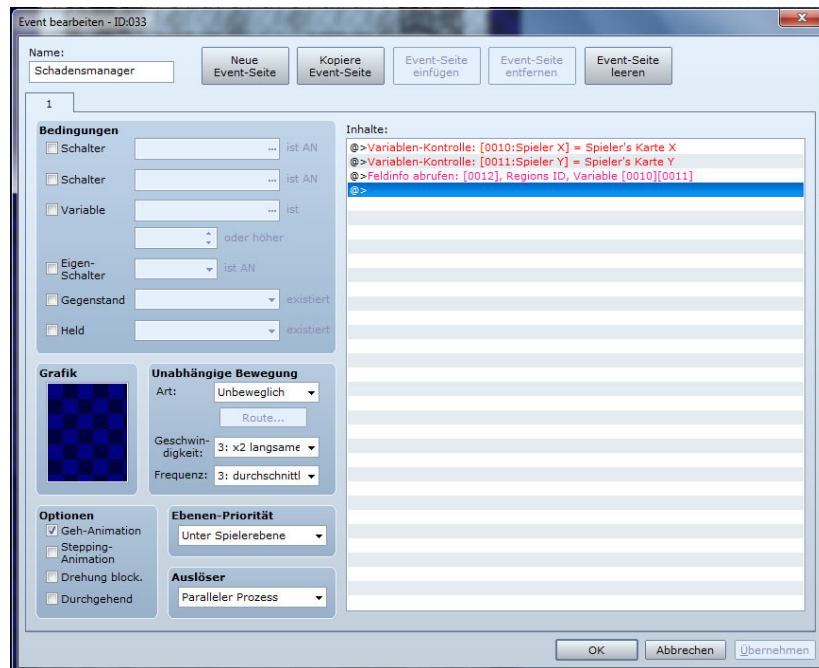
➡ Info-Typ ist Region ID

➡ Bestimmungsort wird durch die Variablen X und Y des Spielers ermittelt

Wir fragen also ab, auf welchem Regions-ID-Feld der Spieler ist.



Fertig sieht der ganze Spaß nun so aus:



An dieser Stelle kommen wir nun zu den beiden Schaltern, die wir vorhin gesetzt haben. Ihr erinnert euch sicherlich an die Schalter „**Raster I**“ und „**Raster II**“.

Wir hatten es so gemacht, dass der Schalter „**Raster I**“ an ist, wenn die Stacheln ausgefahren sind, dasselbe bei „**Raster II**“.

Darum fragen wir nun ab:

➡ Bedingung Schalter „**Raster I**“ AN?

Wenn nein, passiert nichts wenn ja, beginnt die nächste Bedingung.

➡ Bedingung **Regions ID = 1**?

Genau hier für brauchen wir die Regions IDs! Ist die Regions ID nicht 1, passiert gar nichts. Wäre aber Regions ID so groß wie 1, wissen wir also, der Spieler steht auf Regions ID 1.

Was bedeutet das?

⚠ Durch den Schalter „**Raster I**“ wissen wir bereits, dass die Speere ausgefahren sind. Dadurch, dass die Regions ID die Größe 1 hat, wissen wir auch, dass der Spieler auf einem Feld stehen muss, der zu „**Raster I**“ gehört, dessen Speere ja ausgefahren sind. Was passiert, wenn der Protagonist Bekanntschaft mit ein paar Speeren macht?

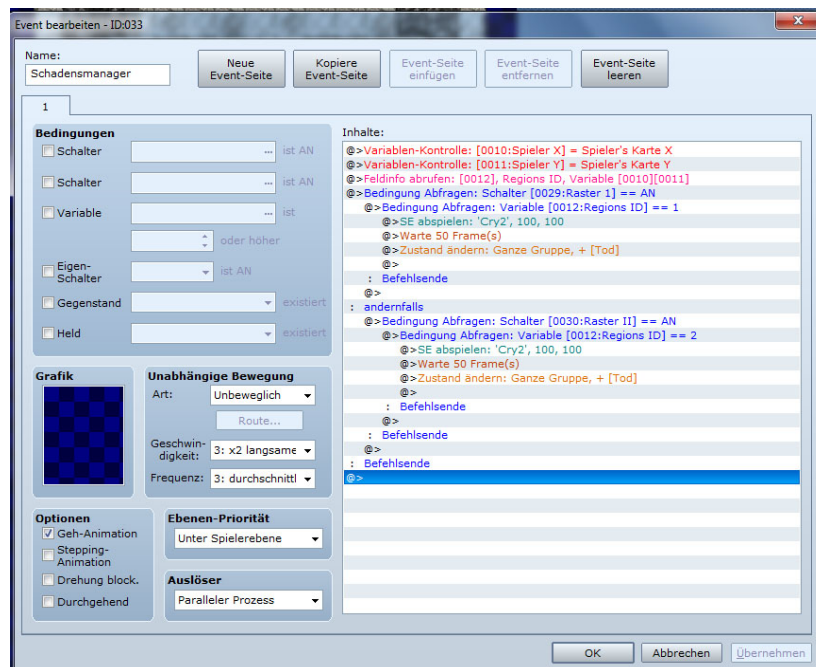
Bei mir passiert folgendes:

➡ Der SE Sound „**Cry2**“ wird abgespielt

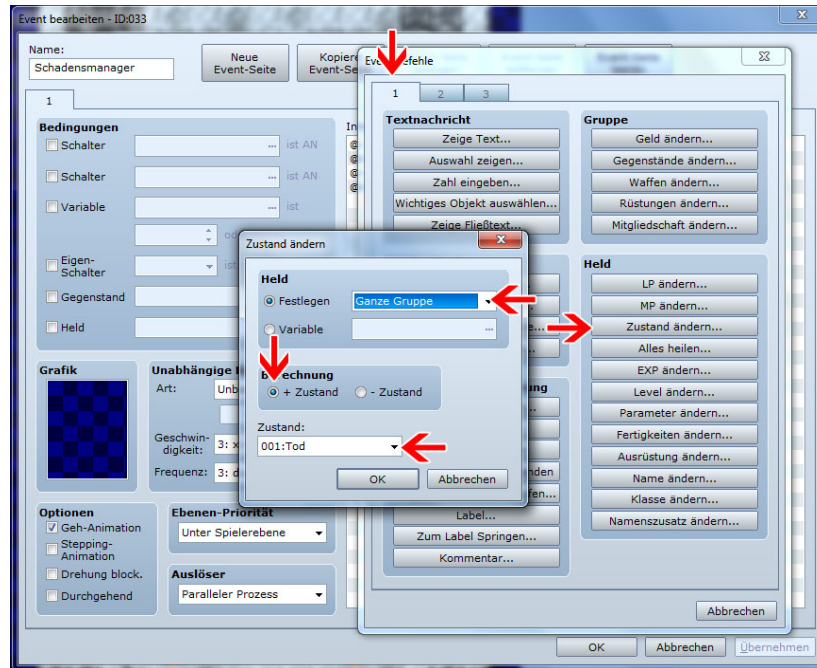
➡ Warte **50 Frames** – muss hier sein, weil der Sound gar nicht abgespielt werden würde und der Maker weiter hetzt

➡ Der Zustand des Spielers wird geändert in Tod – bedeutet Game Over!

Das Selbe wiederholen wir bei „**Regions ID = 2**“ und Schalter „**Raster 2**“:



Für die, die nicht wissen, wie man den Zustand des Spielers ändert:



⚠ Im übrigen kann man außer der gesamten Gruppe auch einstellen, dass nur ein Bestimmter aus der Gruppe stirbt, ich habe ich es jetzt so gelassen, da es egal ist.

Speichert noch zum Schluss mit **OK** ab.

Damit sind wir am Ende des Tutorials angelangt. Ich wünsche euch viel Spaß beim Nachbau!
Für alle Detailverliebten, geht es im nächsten Kapitel noch ein wenig weiter.

Ein klein wenig Detailsverliebtheit?

Willkommen im Bereich für diejenigen, die detailverliebt sind!

Messerscharfe Speere schießen aus dem Boden aber ... Moment mal! Die machen keine Geräusche? Unglaublich, darum spielen wir noch einmal mit Regions IDs herum und erweitern unsere Karte damit noch etwas.

⚠ Mit den Regions IDs steuern wir unsere Geräusche, dabei gilt: Um so weiter weg, um so größer die Regions ID.

Der nächste Schritt ist also ein neues Event, welches wir „**Klingengeräusch**“ nennen.

➡ Bedingung: Ein neuer Schalter namens „**Klingengeräusch 1**“ – wir arbeiten hier mit zwei Geräuschen, einmal, wenn die Klingen ausgefahren werden (Schalter „**Klingengeräusch 2**“) und einmal, wenn die Klingen eingefahren werden (Schalter „**Klingengeräusch 1**“). Wo wir die Schalter setzen, erkläre ich gleich.

➡ Auslöser: **Paralleler Prozess**

Die Regions ID, die wir brauchen, wird ja schon im Event „**Schadensmanager**“ abgefragt, darauf können wir in diesem Event gleich aufbauen.

Zum Inhalt:

➡ Wir fragen ab: ist die **Regions ID 1**, also steht der Spieler auf einem **Regions ID 1 Feld**?

Wenn nein, passiert nichts.

Wenn ja, passiert folgendes:

➡ Der SE Sound „**Sword3**“ wird abgespielt, **Lautstärke 80%**

➡ Schalter „**Klingengeräusch 1**“ wird **ausgeschaltet**

Aber warum?

⚠ Ganz einfach: Die Speere fahren zusammen aus, da brauchen wir nur ein Geräusch. Schalten wir nun diesen Schalter nicht aus, würde der SE durch den parallelen Prozess in jedem Frame neu abgespielt werden. Hört sich nicht sehr schön an und wir brauchen den Sound ja nur einmal.

Andernfalls:

➡ Wir fragen ab: ist die **Regions ID 2**, also steht der Spieler auf einem **Regions ID 2 Feld**?

Wenn nein, passiert nichts.

Wenn ja, passiert folgendes:

➡ Der SE Sound „**Sword3**“ wird abgespielt, **Lautstärke 80%**

➡ Schalter „**Klingengeräusch 1**“ wird **ausgeschaltet**

Andernfalls:

→ Wir fragen ab: ist die **Regions ID 3**, also steht der Spieler auf einem **Regions ID 3 Feld**?
Wenn nein, passiert nichts.

Wenn ja, passiert folgendes:

- Der SE Sound „**Sword3**“ wird abgespielt, **Lautstärke 70%**
- Schalter „**Klingengeräusch 1**“ wird ausgeschaltet

Andernfalls:

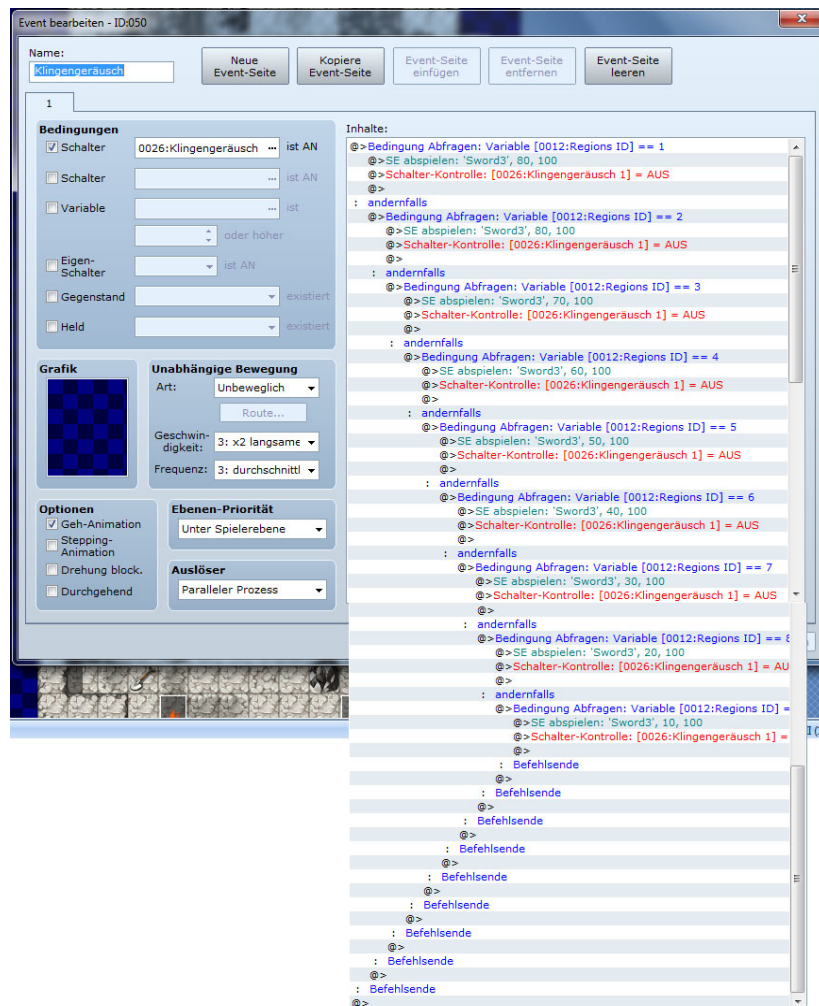
→ Wir fragen ab: ist die **Regions ID 4**, also steht der Spieler auf einem **Regions ID 4 Feld**?
Wenn nein, passiert nichts.

Wenn ja, passiert folgendes:

- Der SE Sound „**Sword3**“ wird abgespielt, **Lautstärke 60%**
- Schalter „**Klingengeräusch 1**“ wird ausgeschaltet

etc.

Danach müsste es bei euch so aussehen:



Damit haben wir die erste Reiterseite fertig. Diese kopieren wir uns jetzt.

Änderungen an der zweiten Reiterseite:

- ➡ Bedingung: Ein neuer Schalter Namens „Klingengeräusch 2“
- ➡ Die Schalter im Inhalt die deaktiviert werden auf „Klingengeräusch 2“ ändern
- ➡ Das SE Geräusch auf „Sword4“ ändern

Fertig! Damit kann das Event mit **OK** geschlossen und gespeichert werden.

Als letztes öffnen wir uns noch einmal das Event „Schadensmanager“.

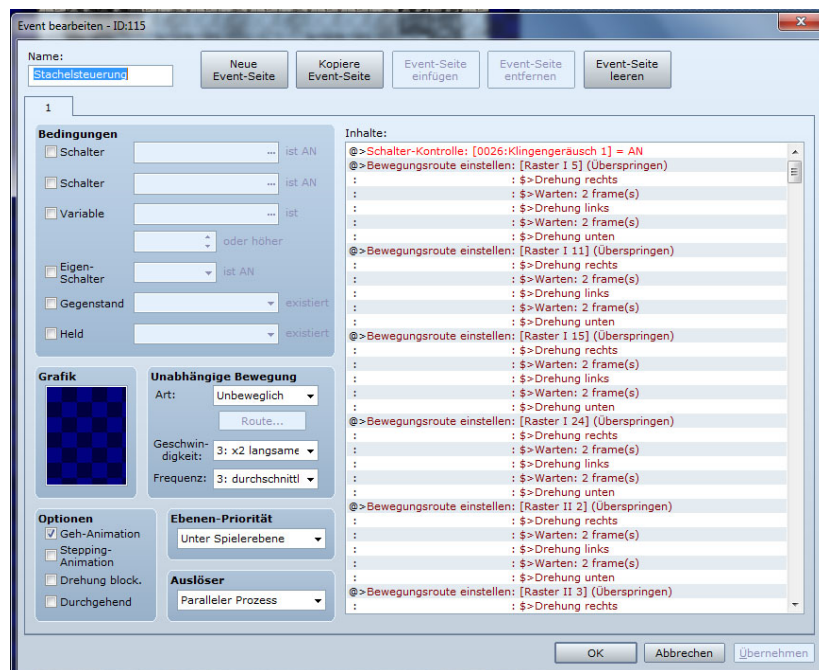
Hier werden wir nun nämlich festlegen, wann die Schalter „Klingengeräusch 1“ und „Klingengeräusch 2“ angeschaltet werden.

Wir rufen uns kurz ins Gedächtnis:

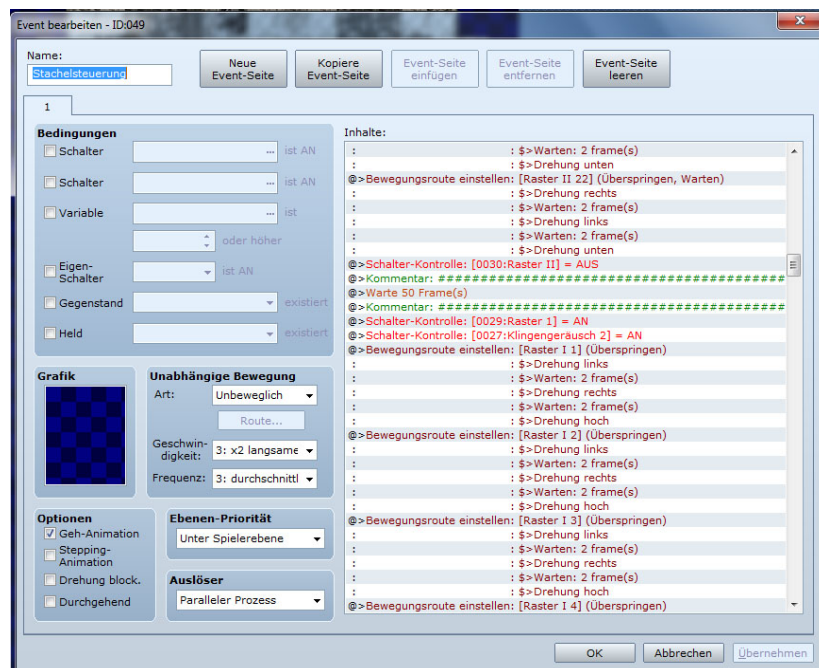
⚠ Schalter „Klingengeräusch 1“ für das Geräusch, wenn die Speere in den Boden gehen

⚠ Schalter „Klingengeräusch 2“ für das Geräusch, wenn die Speere aus dem Boden kommen

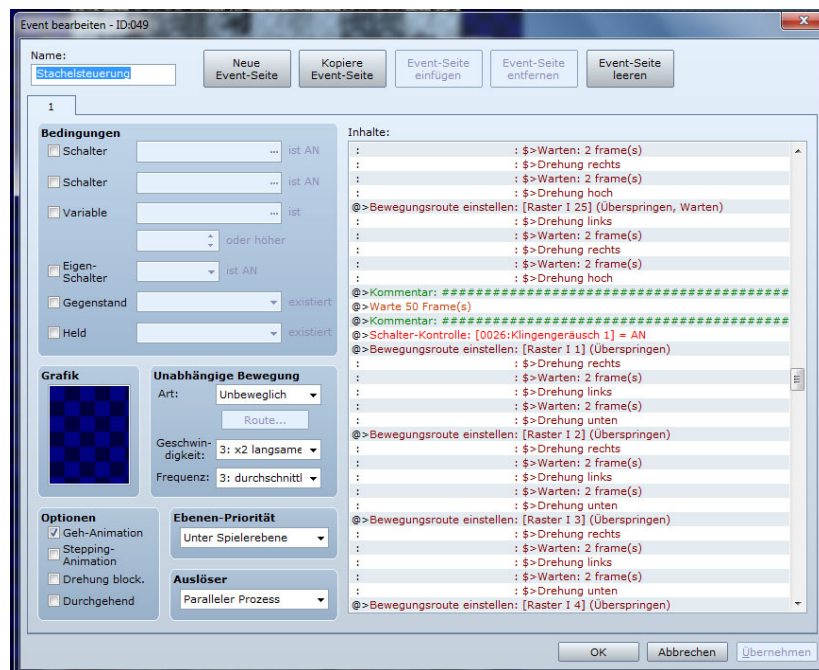
Schalter „Klingengeräusch 1“ gleich am Anfang anschalten, da als erstes die Speere eingefahren werden.



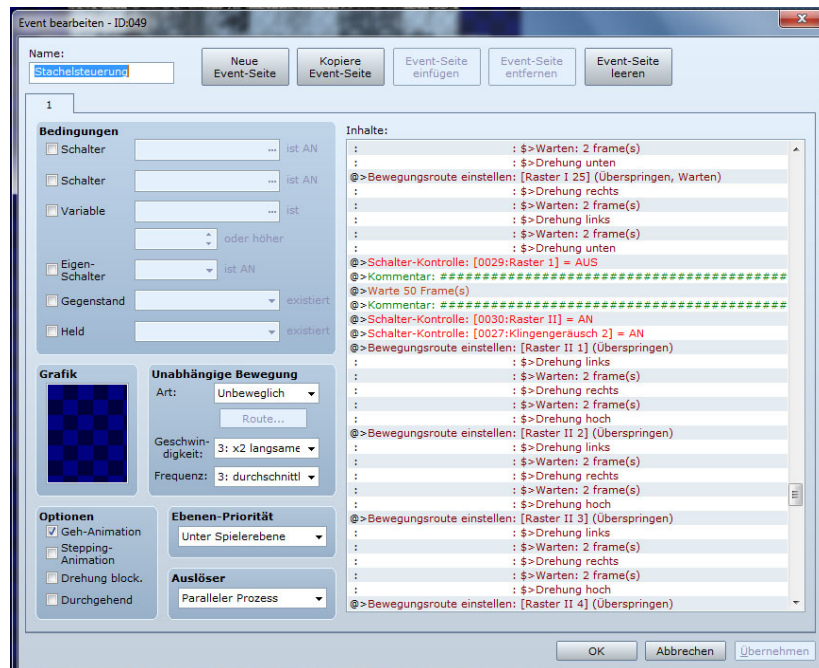
Etwas weiter unten Schalter „Klingengeräusch 2“ anschalten, da die Speere aus dem Boden kommen.



Noch viel weiter unten Schalter „Klingengeräusch 1“ gleich anschalten, da die Speere wieder eingefahren werden.



Fast am Ende noch einmal Schalter „**Klingengeräusch 2**“ anschalten, da die Speere aus dem Boden kommen.



Das war es schon. Von nun an gibt es auch Geräusche, wenn die Speere ein- und ausfahren.

Demo-Tutorial

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

<http://gamealchemists.com/?wpdmdl=183>