

RPG MAKER XP

RPG Maker XP Kurs Teil 1

Inhaltsverzeichnis

1	Vorwort.....	2
2	RPG Maker XP Grundkurs Teil 1.....	2
	2.1 Erste Schritte.....	2
	2.2 Die Arbeitsoberfläche.....	3
	2.3 Das Erstellen von Maps.....	4
	2.4 Ereignisse.....	7
	2.5 Die Datenbank.....	12
	2.5.1 Actors.....	13
	2.5.2 Classes.....	17
	2.5.3 Skills.....	19
	2.5.4 Items.....	21
	2.5.5 Weapons.....	24
	2.5.6 Armors.....	25
	2.5.7 Enemies.....	27
	2.5.8 Troops.....	30
	2.5.9 States.....	33
	2.5.10 Systems.....	35
3	Schlusswort.....	37

Vorwort

Herzlich Willkommen zum ersten Teil dieses Kurses, bei dem es darum gehen soll das Grundwissen zum Gebrauch des RPG Maker XPs zu vermitteln. Im ersten Teil soll es zunächst einmal allgemein um die Arbeitsumgebung gehen, über das Erstellen von Karten, Erste Worte zu Ereignissen und um die Datenbank. (zu Deutsch: die Grundlagen)

Dabei sei gesagt, dass dieser Kurs sich vor allem an Anfänger richtet die noch nichts mit dem Maker zu tun hatten. Wer bereits etwas mit den älteren Makern gearbeitet hat kann die meisten Kapitel hier überspringen und auch im Allgemeinen werden hier viele Dinge erklärt die einigen vielleicht trivial erscheinen.

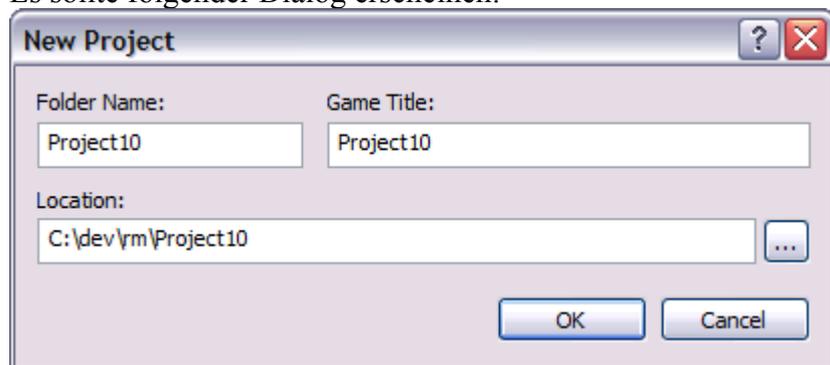
Dieser Kurs wird zudem laufend aktualisiert und verbessert. Dies ist jedoch nur eine Offline-Fassung, in der ihr diese Verbesserungen natürlich nicht zu sehen bekommt. Deshalb schaut euch doch zu gegebener Zeit die aktualisierte Online-Fassung unter dieser URL an: http://www.rpga.info/RPG_Maker_XP_Kurs.html

Wir hoffen dass dieser Kurs verständlich ist und euch den Einstieg ins „makern“ erleichtert.

Erste Schritte

Nachdem ihr also den RPG Maker XP (im folgenden ‚Maker‘ genannt) erfolgreich installiert und geöffnet habt, solltet ihr erstmal eine überwältigend leere Arbeitsoberfläche vor euch sehen. Bevor wir etwas damit anfangen können müssen wir zuerst mal ein neues Projekt erstellen, was man entweder über die Schaltfläche File -> New Project, oder über die Tastenkombination Strg+N erreicht.

Es sollte folgender Dialog erscheinen:



Unter ‚Folder Name‘ könnt ihr den Namen des Ordners einstellen, unter ‚Game Title‘ den Namen eures Spiels. (letzterer wird während des Spielens als Titel des Fensters angezeigt)

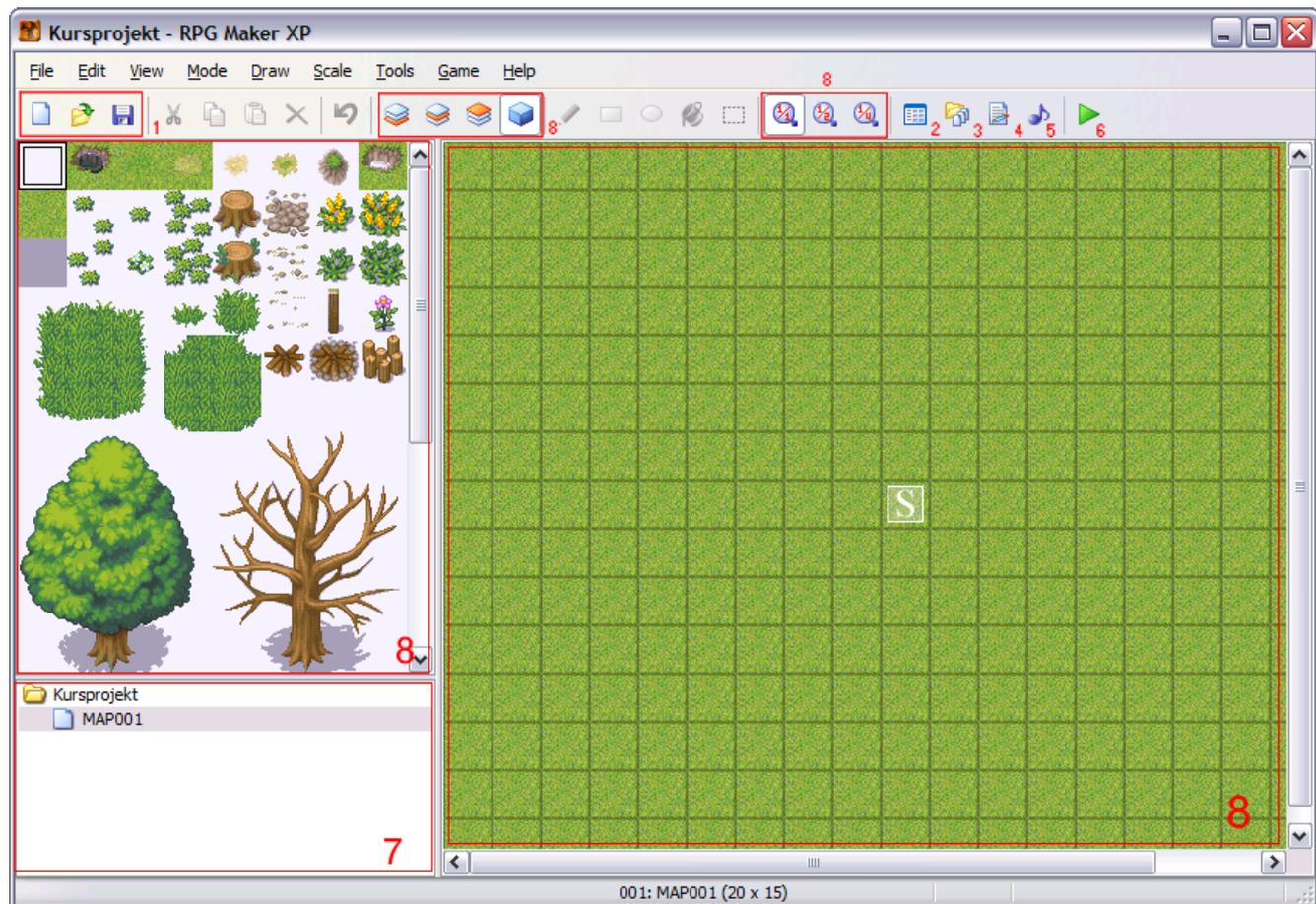
Bei dem Punkt ‚Location‘ könnt ihr bestimmen wo genau der Ordner erstellt wird – am besten an einem Ort den ihr später auch wieder finden könnt.

Wir werden dieses Projekt mal im Ordner „Kursprojekt“ erstellen, und ihm auch genau diesen Namen geben.

Nachdem diese ersten Einstellungen stehen könnt ihr das ganze mit einem graziösen Klick auf ‚OK‘ bestätigen und damit habt ihr so eben ein Spiel begonnen!

Die Arbeitsoberfläche

Jetzt sieht auch die Arbeitsoberfläche gleich nicht mehr so leer aus... genau genommen, sieht sie ziemlich voll aus:



Wir haben uns mal die Freiheit genommen, das ganze mit Zahlen zu versehen, denn in dieser Reihenfolge wollen wir es abhandeln.

Bei 1. findet ihr 3 nützliche Icons die von links nach rechts folgendes tun: Ein neues Projekt erstellen, ein altes Projekt öffnen und das derzeitige Projekt speichern. Besonders von dem dritten Button solltet ihr oft gebrauch machen.

2. Ist ein Verweis auf die sog. „Datenbank“. Dort könnt ihr viele Einstellungen bzgl. eures Projektes vornehmen, von Helden und Monstern bis hin zu neuen Animationen. Wenn ihr einen kurzen Blick darauf werft, werdet ihr sehen, dass das doch ein größerer Themenkreis ist, den wir später noch genauer betrachten wollen.

3. Unter diesem Punkt könnt ihr sehr bequem eigene Ressourcen (Grafiken und Musik) in euer Projekt importieren, welche ihr dann hier benutzen könnt. An dieser Stelle noch eine Notiz an alle die bereits mit früheren Makern gearbeitet haben: Es ist bei dem XP nicht mehr strikt notwendig die Grafiken zu importieren, man kann sie auch einfach in die entsprechenden Ordner kopieren. Im Gegensatz zu früheren makern wird das nicht euer Projekt kaputt machen.

4. Der Skripteditor, eine der Neuerungen beim RPG Maker XP, und ein Thema das man eigentlich kaum in ein paar wenige Zeilen fassen kann. Erfahrene Nutzer können sich auf jeden Fall über diese Neuerung freuen, da sie im Vergleich zu älteren Makern wirklich ungeahnte Möglichkeiten eröffnet.

Thema dieses Kurses ist diese Funktion jedoch noch nicht, wer sich dennoch mal informieren möchte sollte sich den Artikel zu Ruby durchlesen, welcher sich ebenfalls auf dieser CD befindet.

5. Hier findet ihr den ‚Sound Test‘ bei dem ihr euch einen Überblick über die Musik eures Spiels verschaffen könnt. Standardmäßig sind bereits einige Lieder darin zu finden, welche Teil des RTP des Makers sind. (RTP = Run Time Package, Standardressourcen für den Maker)

6. Das Testspiel. Im Moment könnt ihr mal Probeweise darauf klicken, ihr werdet bereits das Grundgerüst eines RPGs vorfinden, auch wenn das auf Anhieb nicht viel hermacht.

Im Verlauf dieses Kurses werden wir sicher noch oft genug von diesem Punkt gebrauch machen, und dann hoffentlich mit interessanteren Ergebnissen.

7. Der sog. ‚Map Tree‘. Hier werden alle Karten eures Spiels gespeichert und organisiert, bzw. neue Karten erstellt. Und wem das jetzt nichts sagt: genau darum geht es im nächsten Punkt.

8. Der ‚Map Editor‘ in (nahezu) allen seinen Facetten.

Als ‚Map‘ (zu Deutsch: Karte) bezeichnet man ein Rechteckiges Bild welches aus vorgefertigten 32x32 Pixel großen Feldern zusammengesetzt ist. ‚Maps‘ sind ein gängiges Konzept in 2D Spielen, die Umgebungen in praktisch allen 2D Spielen werden auf diese Art und Weise erstellt. (wenn auch die einzelnen Felder vielleicht andere Größen haben)

Das Erstellen von Maps (mappen)

Maps im Maker bestehen aus insgesamt 4 Ebenen: den ersten drei, die wirklich nur die angesprochenen Grafiken enthalten, und der Ereignisebene auf die wir später NPCs und andere Objekte setzen. Standardmäßig startet man auf der Ereignisebene, da wir aber zunächst etwas mappen wollen müssen wir auf eine andere Ebene (Layer) wechseln. Das erreicht man, in dem man auf ein entsprechende Icon oben in der Leiste klickt: [BILD EINFÜGEN]

F5, F6 oder F7 drückt (F8 würde euch wieder auf die Ereignisebene bringen)

Oder ganz einfach über die Schaltfläche Mode -> Layer 1 (bis 3)



Sobald ihr euch auf einer editierbaren Ebene befindet könnt ihr zunächst mal links bei dem Tileset eines der angesprochenen Felder auswählen.

Und dazu noch oben bei den Icons eines der vorhandenen Zeichenwerkzeuge:



Und schließlich in dem großen Fenster auf der rechten Seite drauf los pinseln.

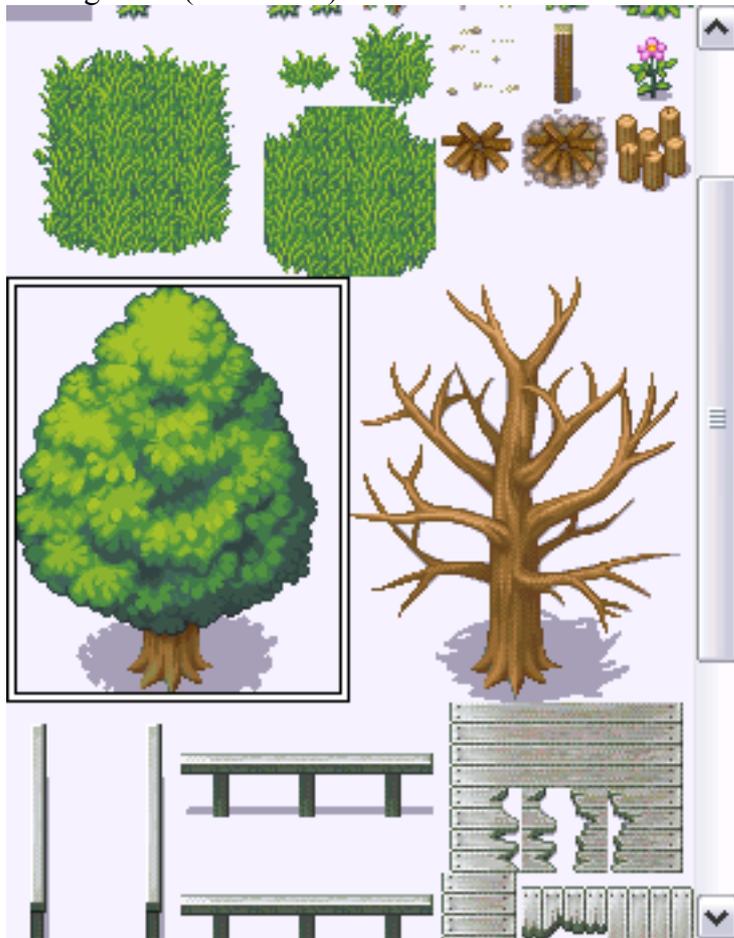
Hier gibt es noch ein paar Dinge zu beachten:

Auf einem Chipset sind die oberen 8 Felder immer die sog. ‚Autotiles‘ alle Felder darunter zeichnen exakt das, was ihr bereits in der Vorschau seht. Die Autotiles verhalten sich etwas anders, und versuchen euch weichere Übergänge zwischen den einzelnen Feldern zu geben, ohne das ihr selbst viel dazu tun müsst. Probiert es einfach mal aus, denn hierbei geht nichts über Probieren.

Felder mit weißem Hintergrund sind wiederum transparent. Solltet ihr euch auf der untersten Ebene befinden, ist es nicht sehr ratsam mit solchen Feldern zu pinseln, da an solchen Stellen der Hintergrund

durchscheinen würde. Um diese richtig zu benutzen wechselt einfach auf die Ebenen 2 oder 3, da werdet ihr mehr Erfolg haben.

Ihr müsst natürlich nicht immer nur einzelne Felder auswählen, haltet einfach die Maustaste gedrückt und wählt einen ganzen (sinnvollen) Abschnitt aus:

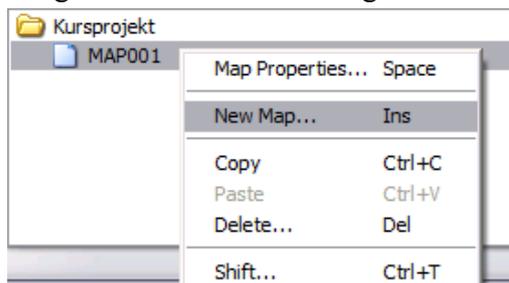


Das sollte die Arbeit etwas beschleunigen.

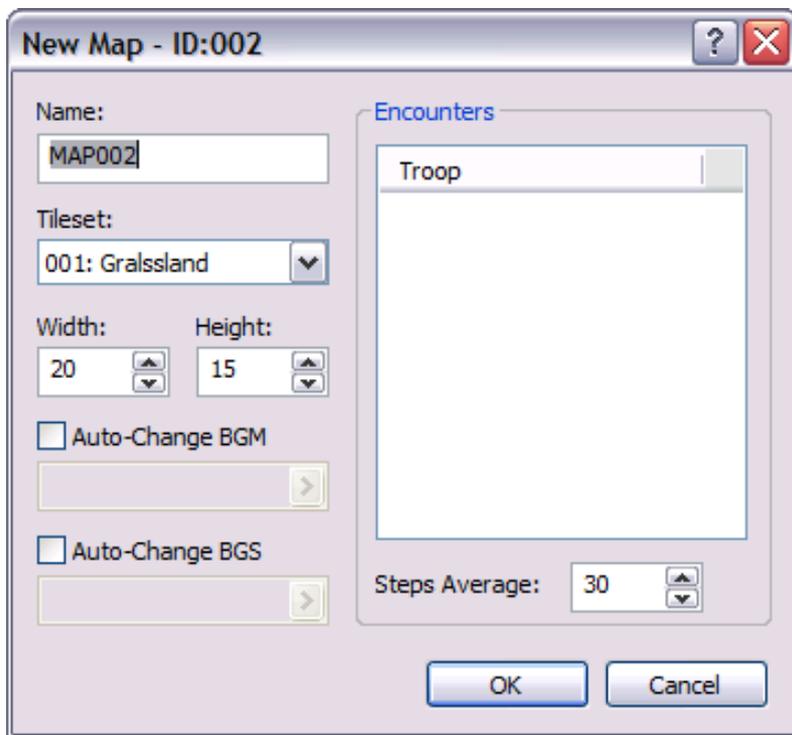
An dieser Stelle probiert es einfach einmal aus. Es ist wirklich ziemlich selbst erklärend, sobald man mal den Dreh raus hat, und liefert vor allem einfach ganz anschauliche Ergebnisse. Maps wie man sie aus Topspielen wie Velsarbor kennt wurden auf die gleiche Art und Weise erstellt, alles was es dazu braucht ist Übung und Zeit. In einem schnellen Testspiel (F12, oder Icon s.o.) können wir unser Werk begutachten. So sieht's zum Beispiel bei mir aus:



Bevor wir uns nun selbstsicher an die Ereignisse wagen, erstellen wir doch zuerst mal eine weitere Map, um das auch mal gemacht zu haben. Durch einen Rechtsklick auf den eben angesprochenen Map Tree links unten im Programm bekommt ihr folgende Schaltfläche zu Gesicht:



Wir klicken wie hier angedeutet auf ‚New Map...‘ und gelangen so in folgenden Dialog:



Den gleichen Dialog hätten wir durch einen Klick auf ‚Map Properties‘ bekommen, bloß hätte er sich auf die derzeitige Map bezogen.

Nun gut, was kann man hier alles einstellen? Unter ‚Name‘ kann man der Map zuerst mal einen Namen zuweisen, was wir am besten gleich mal tun, da ‚MAP002‘ nicht besonders Deskriptiv ist. Wir nennen unsere Map mal ‚Stadt‘.

Unter ‚Tileset‘ kann man die Sammlung an Feldern bestimmen mit der man die Map gestalten will. Da wir in diesem Falle eine Stadt mappen wollen, ist ‚Grassland‘ nicht das ideale Tileset dazu, weshalb wir stattdessen ‚Port Town‘ (Hafenstadt) gewählt haben.

Nebenbei seid ihr natürlich nicht auf diese paar Tilesets beschränkt, man kann sich auch seine eigenen Erstellen, wozu wir aber erst später kommen.

‚Width‘ und ‚Height‘ (Breite und Höhe) beziehen sich auf die Maße der Map in Feldern. 20x15 ist genau ein Bildschirm, und damit die minimale Größe für eine Map. Es ist allerdings keine gute Idee zu große Maps zu erstellen, da der Aufwand sehr schnell sehr groß werden kann.

Unter ‚Auto-Change BGM‘ (Automatische Veränderung der Hintergrundmusik) kann man die Musik für diese Map einstellen. Klickt erstmal das Kästchen an und wählt dann darunter eine Musik aus die euch gefällt, bzw. passt.

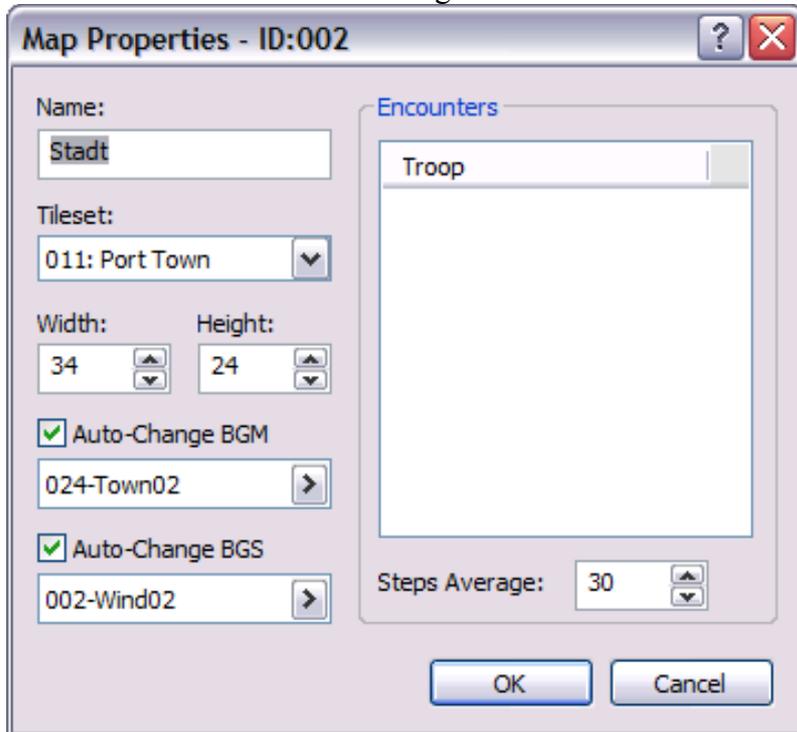
‚Auto-Change BGS‘ ist genauso zu handhaben wie ‚Auto-Change BGM‘ mit dem Unterschied, dass man hier ein Hintergrundgeräusch einstellen kann. (Wind, Regen, Meeresrauschen, etc.)

Unter ‚Encounters‘ kann man Zufallskämpfe auf dieser Map einstellen. Durch einen Doppelklick auf das weiße Feld unter ‚Troop‘ kann man die einzelnen Gegnertruppen auswählen die einem im Spiel gegenüber stehen soll. Spezifische Einstellungen dazu kann man in der Datenbank vornehmen, wozu wir aber erst später kommen.

‚Steps Average‘ gibt die Anzahl an Schritten an die im Durchschnitt zwischen den Kämpfen vergehen. Man sollte gerade hier vorsichtig sein, denn zu niedrige Zahlen werden einige Spieler schnell frustrieren, bzw. langweilen.

Nachdem ihr alle Einstellungen vorgenommen habt könnt ihr das ganze mit ‚OK‘ bestätigen und habt damit eine neue Map erstellt die ihr in Angriff nehmen könnt. Solltet ihr die Einstellungen noch einmal überarbeiten wollen, müsst ihr die Map markieren und auf ‚Map Properties‘ klicken, das bringt euch wieder zu dem Dialog von oben zurück.

Am Ende sahen meine Einstellungen in etwa so aus:



An dieser Stelle könnt ihr nun wieder etwas mappen, aber da es nicht sonderlich wichtig ist werden wir nicht weiter darauf eingehen. Was nun aber wichtig ist, ist die Frage wie man die Spielfigur auf diese Map bekommt. Dazu wechseln wir nun (endlich) auf die Ereignisebene (F8).

Ereignisse

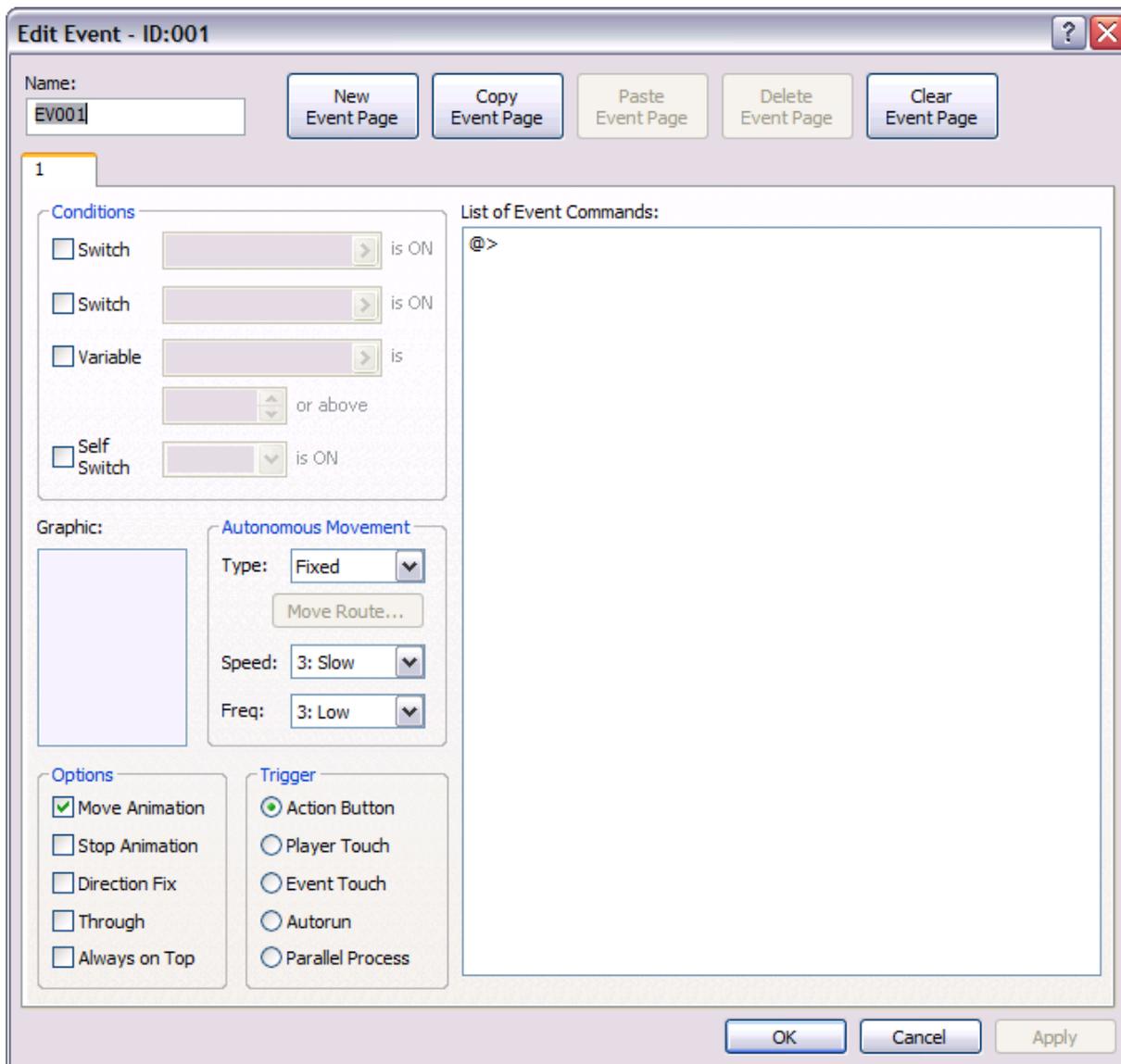
Den Spieler auf diese Map zu bekommen ist nicht sonderlich schwer: Einfach einen Rechtsklick auf das Feld auf dem er stehen soll und dann ‚Player’s Starting Position‘ auswählen. Jetzt startet das Spiel auf dieser Map, und an dieser Stelle...

Auf der anderen Seite, muss man zugeben, dass es bisher noch kein großartiges Spiel ist, was wir da produziert haben. Der Spieler kann sich etwas bewegen, und unter Umständen auch schon etwas kämpfen, aber es bringt nichts da keine wirkliche Interaktion vorhanden ist.

Genau an dieser Stelle kommen die ‚Events‘ (zu Deutsch: Ereignisse) ins Spiel.

Events treiben das ganze Spiel voran. Ein NPC ist genau so ein Event, wie eines das zum Beispiel einen Intro-Text anzeigt. Wir fangen am besten mal mit einem NPC an.

Nach einem Doppelklick auf das gewünschte Feld sollte sich ein etwas größerer Dialog öffnen in dem man sich erneut erstmal zurecht finden muss:



Unter ‚Name‘ könnt ihr zuerst das Event betiteln. Das ist nicht wirklich notwendig, aber genau wie bei einer Map ist es auf Dauer guter Stil, da man sich so besser zurecht findet.

Die Kontrollen daneben ignorieren wir für den Moment, genauso wie das als ‚Conditions‘ betitelte Kästchen. Unter ‚Graphic‘ kann man dem Event eine Grafik zuweisen, daneben und darunter hat man noch ein paar Einstellungen wie sich das Event bewegt und wie es animiert wird.

Da dieses Event wie eben bereits erwähnt ein NPC werden soll, weisen wir ihm bereits eine entsprechende Grafik zu. Würde man sich das ganze jetzt in einem Testspiel anschauen würde der NPC bereits da stehen, jedoch absolut nichts tun.

Als erstes sollte er sich etwas bewegen können, dass erreicht man mit den Einstellungen unter ‚Autonomous Movement‘ (Selbstständige Bewegung).

Unter ‚Type‘ kann man die Art der Bewegung einstellen:

- Fixed: Das Event bewegt sich gar nicht und bleibt auf der Stelle stehen.
- Random: Die Eventbewegung wird zufällig bestimmt.
- Approach: Das Event bewegt sich auf den Spieler zu
- Custom: Hierbei muss man unter ‚Move Route‘ die gesamte Bewegung selbst bestimmen.

Bei ‚Speed‘ und ‚Freq.‘ kann man nun noch die Bewegungsgeschwindigkeit und die Zeit die zwischen zwei Schritten vergeht bestimmen.

Unter ‚Options‘ könnte man noch etwas an der Darstellung verändern. ‚Move Animation‘ legt fest, dass das Event beim bewegen animiert ist, ‚Stop Animation‘ sagt dagegen aus, dass das immer der Fall ist.

Ist ‚Direction Fix‘ aktiviert wird das Event seinen Gesichtsrichtung nicht verändern, egal in welche Richtung es sich bewegt. ‚Through‘ bedeutet, dass das Event auch durch Wände und allgemein durch an für sich unpassierbare Gegenstände gehen kann.

‚Always on Top‘ sagt aus, dass das Event über allen anderen Gegenständen auf der Map ist. (zum Beispiel, ein Vogel der lediglich über die Map hinweg fliegt und nicht wirklich darin steht)

Wenn wir das richtig gemacht haben sollten wir in dem Spiel bereits eine Änderung beobachten können, der NPC wird sich bewegen.

Davon abgesehen wird er jedoch nichts tun, und genau das wollen wir an dieser Stelle beheben. Mit einem Doppelklick auf das große weiße Feld rechts im Event Editor kommen wir zu den Event Befehlen, die eigentlich das ganze Spiel ausmachen:



Das, sowie die 2 anderen Seiten werden uns in nächster Zeit beschäftigen...

Zu Begin, klickt einfach einmal auf ‚Show Text...‘, was im Spiel ein Nachrichtenfenster öffnen wird, mit dem Text den ihr an dieser Stelle eintippt.

Da das eine sehr häufige Aktion ist, kann man gleich viele dieser Nachrichten auf einmal schreiben, indem man in dem weißen Fenster einfach Strg+N drückt.

An dieser Stelle schauen wir uns noch einmal das Kästchen mit der Beschriftung ‚Trigger‘ (Auslöser) an, dort kann man einstellen, wann genau das Event ausgeführt wird.

Das Event wird ausgeführt, wenn...

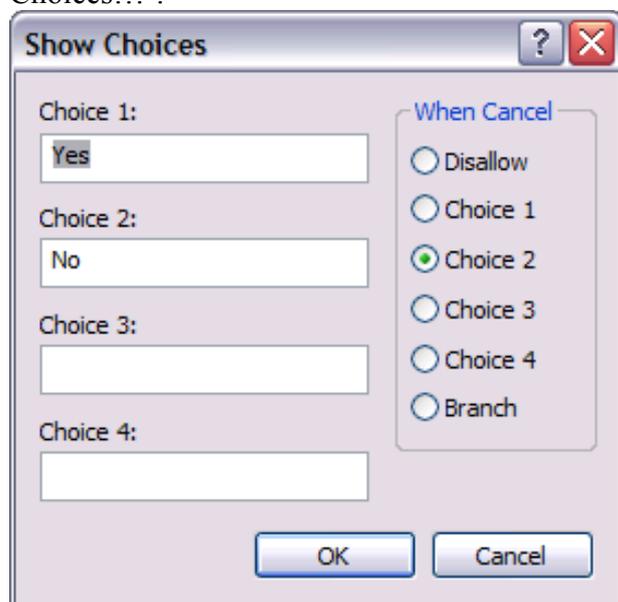
- Action Button: ...der Spieler davor steht und Enter drückt.
- Player Touch: ...wenn der Spieler es berührt.
- Event Touch: ...wenn das Event den Spieler berührt. (Unterschied zu Player Touch: man muss sich nicht in Richtung Event bewegen, das Event kann von sich aus ausgelöst werden)

Die letzten beiden Auslöser verhalten sich wieder etwas anders:

- Autorun: Das Event wird sofort ausgeführt, und hält das gesamte Geschehen an, bis das Kommando ‚Erase Event‘ ausgeführt wird.
- Parallel Process: Das Event wird wie bei Autorun sofort ausgeführt, läuft allerdings parallel zum Rest des Spieles.

Für einen NPC ist ‚Action Button‘ normalerweise am besten geeignet.

Das nächste Kommando das wir uns vornehmen werden, ist direkt darunter zu finden, und trägt den Titel ‚Show Choices...‘:



‚Show Choices‘, zu Deutsch: Entscheidungsmöglichkeiten anzeigen, stellt den Spieler innerhalb des Spieles vor eine Wahl zwischen bis zu 5 verschiedenen Möglichkeiten.

Diese Wahlmöglichkeiten kann man unter den Textkontrollen ‚Choice 1‘, bis ‚Choice 4‘ einstellen.

Zusätzlich dazu gibt es noch den Fall, dass der Spieler die Wahl abbricht (im Spiel durch Druck auf die Escape-Taste), dieser Fall wird im Feld auf der rechten Seite eingestellt.

- ‚Disallow‘ verbietet das Abbrechen der Wahl
- ‚Choice1‘ bis ‚Choice4‘ führen einfach die jeweilige Entscheidung aus, wenn der Spieler das ganze Abbricht. Wenn man zum Beispiel wie oben nach ‚Ja‘ oder ‚Nein‘ fragt, könnte man bei Abbruch ganz einfach den ‚Nein‘ Fall ausführen. (was man aus einigen kommerziellen RPGs gewöhnt sein sollte)
- ‚Branch‘ verarbeitet den Abbruch spezifisch, als eine eigene Auswahlmöglichkeit.

Nachdem ihr die Eingabe bestätigt habt solltet ihr im Event Editor in etwa folgenden Aufbau sehen:

```
@>Show Choices: Ja, Nein
: When [Ja]
@>
: When [Nein]
@>
: Branch End
```

Jetzt könnt ihr unter ‚When [...]‘ die einzelnen Fälle abhandeln, nach dem ‚Branch End‘ werden die einzelnen Stränge wieder zusammen geführt. Man kann diese Sorte Befehl auch verschachteln, und somit schnell komplexe Entscheidungen darstellen.

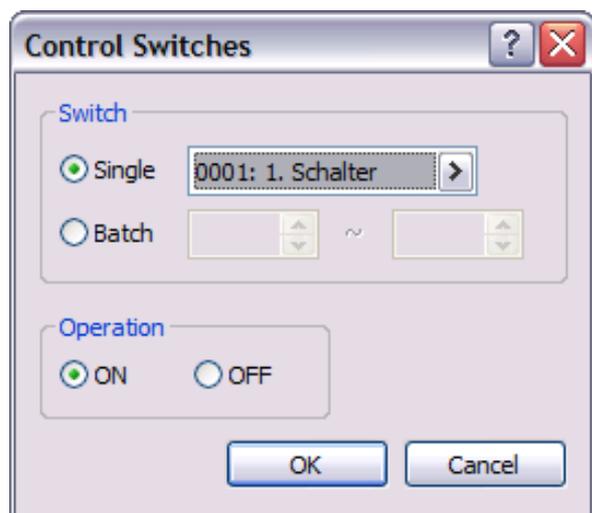
Aber all das nützt uns nicht viel, da wir bisher nicht wirklich etwas am Spiel verändern können, wir könnten nur verschiedene Nachrichten anzeigen, und das ist mit Sicherheit nicht der Sinn der Sache.

Gehen wir also vorerst zurück in den Event Editor. Diesmal betrachten wir das Kästchen ‚Conditions‘ (Bedingungen)

Hier kann man Bedingungen eingeben, wann ein Event aktiv (bzw. überhaupt sichtbar) sein soll. Das erste Bedingungsgefüge des makers sind die sog. ‚Switches‘ (Schalter).

Die Bezeichnung ist sehr treffend, denn Switches, haben genau wie die meisten ihrer Vorbilder in der realen Welt nur zwei Zustände, entweder sind sie an, oder sie sind aus. (ON und OFF)

Erstellen wir also ein neues Event und geben ihm die Bedingung, dass der Switch 1, nennen wir ihn mal ‚1. Schalter‘, an ist. Nun gehen wir in das andere Event zurück und schauen uns auf der ersten Seite der Event Befehle rechts, den Befehl ‚Control Switches‘ näher an.



Unter dem Kästchen Switch kann man sich zuerst einmal unter ‚Single‘ den entsprechenden Schalter auswählen, oder unter ‚Batch‘ eine ganze Reihe an Schaltern gleichzeitig bearbeiten. (letzteres geht über die Nummern der einzelnen Schalter, unser 1.Schalter hat zum Beispiel die Nummer 0001)

Im Kästchen darunter, ‚Operation‘, kann man nun einstellen was man mit diesem/diesen Schalter/n tun will. Entweder kann man sie anschalten (ON), oder ausschalten (OFF). Zu Beginn ist jeder Schalter ausgeschaltet, und sowieso wollten wir ihn anschalten... somit stimmen die Einstellungen oben.

Damit sieht das komplette Skript dieses NPCs nun so, oder so ähnlich, aus:

```
@>Text: Hallo ich bin ein NPC.  
:      : Wollt ihr Schalter 1 anschalten?  
@>Show Choices: Ja, Nein  
: When [Ja]  
  @>Text: Wird gemacht!  
  @>Control Switches: [0001: Schalter 1] = ON  
  @>  
: When [Nein]  
  @>Text: Nun, dann einander mal.  
  @>  
: Branch End
```

An dieser Stelle können wir uns noch ein weiteres Mal ein Testspiel zu Gemüte führen, um uns zu versichern, dass alles so funktioniert wie wir es beabsichtigt haben.

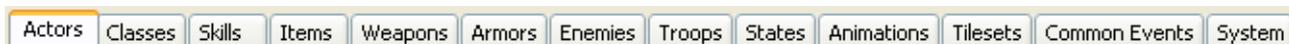
Events sind im Allgemeinen noch ein großes Thema, was an der Fülle von Befehlen die euch zur Verfügung stehen auch ersichtlich wird. Wir werden uns deshalb erst im nächsten Teil weiter mit ihnen befassen.

Als nächstes widmen wir uns der Datenbank.

Die Datenbank

In der Datenbank lassen sich viele grundlegende Einstellungen zu eurem Projekt tätigen. Ihr könnt die Datenbank über das Database-Icon auf der Makeroberfläche, oder mit der F9 Taste öffnen.

Nun öffnet sich ein großes Dialogfenster, welches in 13 Teilabschnitte gegliedert ist.



Sobald ihr einen der Abschnitte (zum Beispiel „Classes“) anklickt, verändert sich das Dialogfeld. Zuallererst wollen wir uns einen kurzen Überblick über diese Abschnitte verschaffen.

Unter „**Actors**“ (zu Deutsch: Akteure, Hauptcharaktere) könnt ihr die Einstellungen der Helden eures Projektes ändern, sowie neue Helden erstellen.

Im Abschnitt „**Classes**“ werden Einstellungen zu den Klassen eurer Helden vorgenommen und neue Klassen entworfen. Jeder Held verfügt über eine spezielle Klasse, z.B. Magier, Barbar oder Bogenschütze. Die Klassen entscheiden über die Fertigkeiten eines Chars, sowie über die Waffen und Rüstungen die er tragen kann.

Bei „**Skills**“ lassen sich Zaubersprüche und spezielle Fertigkeiten für eure Helden und deren Feinde anlegen. Gegenstände wie Heiltränke werden im Menüpunkt „**Items**“ erstellt.

„**Weapons**“ ist für das Erschaffen von Waffen zuständig, während ihr bei „**Armors**“ alle Arten von Rüstungen erstellen könnt.

Unter „**Enemies**“ könnt ihr die Werte und das Aussehen der Feinde verändern und neue Feinde anlegen. Damit diese Feinde allerdings im Kampf auftreten können, müssen sie in eine Gegnertruppe gesetzt werden. Eine Gegnertruppe erstellt ihr im Abschnitt „**Troop**“.

Damit ein Kampfteilnehmer vergiftet, verwirrt oder gar sterben kann, muss man diese Zustände zuerst in „**States**“ erstellen.

„**Animations**“ ist für die Kampfanimationen zuständig. Damit der Einsatz von Zaubern, Items und der Angriff mit Waffen durch grafische Animationen belegt wird, müssen diese Animationen zuerst hier erstellt werden.

Der Abschnitt „**Tilesets**“ wurde ja bereits besprochen, als wir über das Erstellen von Maps gesprochen habt. Ein Tileset ist ein Grafikpaket aus vielen einzelnen Bausteinen. Mit diesen Bausteinen werden die Maps ausgestattet. Dabei gibt es Tilesets für Städte, Wälder, Höhlen und vielen anderen Ortschaften. Unter „**Tilesets**“ könnt ihr die Eigenschaften der Tilesets verändern. Aber ihr könnt auch neue Tilesets erstellen, und somit eigene Mapgrafiken in euer Spiel einbinden.

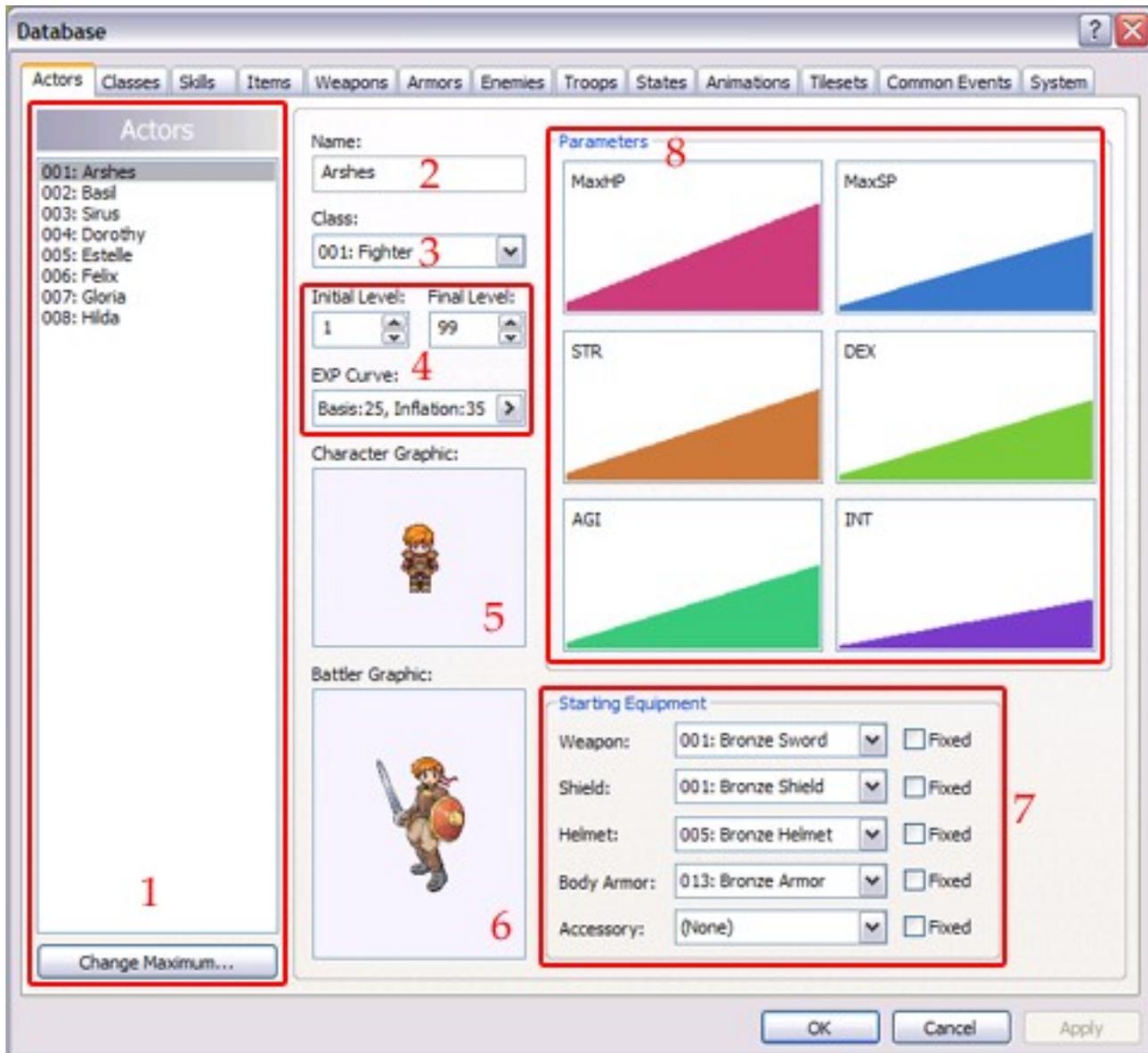
Der Abschnitt „**Common Events**“ ist etwas komplizierter. Wir haben ja bereits über Events gesprochen. Dies sind Ereignisse, denen der Spieler auf einer Map begegnet. Common Events sind jedoch Ereignisse, welche im gesamten Projekt wirken, statt auf eine Map begrenzt zu sein. Dies können zum Beispiel Ereignisse sein, welche die Tageszeit oder das Wetter auf allen Maps steuern.

Der letzte Abschnitt „**System**“ regelt einige grundlegende Eigenschaften eures Projektes. Zum Beispiel werden hier das Titelbild, die Kampfmusik und dergleichen eingestellt.

Wir werden uns nun die einzelnen Abschnitte genauer anschauen.

Actors

Als erstes werden wir uns mit den „Actors“ befassen. Sobald ihr auf den Abschnitt Actors klickt, sollte dieses Dialogfenster erscheinen.



Die Spalte, welche mit der Ziffer 1 gekennzeichnet wurde, enthält eine Liste mit allen Helden, die in eurem Projekt vorkommen. Aufgelistet seht ihr die Nummer, sowie den Namen des Helden. Um einen neuen Helden dieser Liste hinzuzufügen, klickt ihr auf den Button „Change Maximum...“. Nun erscheint ein neues Fenster, in dem ihr die Anzahl der im Spiel vorkommenden Helden einstellen könnt. Ihr habt in eurem neuen Projekt momentan Acht Helden. Um einen neuen Helden hinzuzufügen, gebt also die Zahl Neun ein. Ihr könnt auch einen bestehenden Helden löschen. Klickt hierfür auf den gewünschten Heldenamen in der Liste, und klickt dann auf die Entfernen Taste eurer Tastatur.

Wenn ihr einen neuen Helden erstellt habt, müsst ihr diesem natürlich einen Namen, sowie Aussehen und spezielle Eigenschaften verpassen. Sobald ihr auf einen Zeile in der Heldenliste klickt, erscheinen rechts die Informationen zu diesem Helden. Wählt ihr euren neuen Helden aus, so sind die meisten dieser Felder noch leer.

In dem Feld „Name:“ (2) könnt ihr den Namen eures neuen Helden eingeben. Wir werden unserem Helden den Namen „Alex“ verpassen.

Im Feld „Class“ (3) wählt ihr die Klasse des Helden aus. Die Klassen werden im nächsten Abschnitt der Datenbank festgelegt. Ihr habt jedoch schon zu Beginn einige Klassen vorgegeben. Die Klasse des Helden kann während des Spiels noch geändert werden. In der Datenbank legt ihr also jediglich fest, mit welcher Klasse der Held das Spiel beginnt. Wir geben unserem Helden vorerst die Klasse „Fighter“.

Die nächsten drei Felder (4) regeln das Aufsteigen des Heldenlevels. Standardmäßig gibt es 100 Levels, von Level 1 bis 99. Unter „Initial Level:“ legt ihr fest mit welchem Level der Held beginnt. Bei „Final Level:“ wird das maximale Level, welches der Held erreichen kann, eingegeben.

„EXP Curve:“ (auf Deutsch: Erfahrungspunktekurve) gibt an, wie viele Erfahrungspunkte ein Held benötigt, um ein Level aufzusteigen. Durch einen Doppelklick auf diesen Button öffnet sich ein neues Fenster, in dem alle Level des Helden und deren benötigten Erfahrungspunkte aufgelistet sind.

To Next Level		Total	
L 1:	25	L21:	2089
L 2:	41	L22:	2327
L 3:	63	L23:	2582
L 4:	91	L24:	2854
L 5:	125	L25:	3143
L 6:	168	L26:	3450
L 7:	218	L27:	3775
L 8:	277	L28:	4120
L 9:	346	L29:	4484
L10:	424	L30:	4867
L11:	512	L31:	5271
L12:	612	L32:	5696
L13:	723	L33:	6142
L14:	846	L34:	6609
L15:	982	L35:	7099
L16:	1131	L36:	7610
L17:	1293	L37:	8145
L18:	1470	L38:	8703
L19:	1661	L39:	9285
L20:	1867	L40:	9891
		L41:	10522
		L42:	11177
		L43:	11858
		L44:	12565
		L45:	13298
		L46:	14058
		L47:	14845
		L48:	15659
		L49:	16501
		L50:	17372
		L51:	18271
		L52:	19199
		L53:	20157
		L54:	21144
		L55:	22162
		L56:	23210
		L57:	24290
		L58:	25400
		L59:	26543
		L60:	27718
		L61:	28925
		L62:	30166
		L63:	31439
		L64:	32747
		L65:	34088
		L66:	35464
		L67:	36875
		L68:	38321
		L69:	39802
		L70:	41319
		L71:	42873
		L72:	44464
		L73:	46091
		L74:	47756
		L75:	49459
		L76:	51200
		L77:	52979
		L78:	54797
		L79:	56654
		L80:	58551

Basis: 25 Inflation: 35

Ihr könnt zwischen der „To Next Level“- und der „Total“-Liste wählen. Erstere gibt die Erfahrungspunkte an, welche der Held für den Levelaufstieg benötigt. Zweitere gibt an, wie viele Erfahrungspunkte der Held in dem jeweiligen Level inne hat.

Außerdem könnt ihr zwei Werte verändern: Die „Basis“ und die „Inflation“. Diese sind für die Berechnung der benötigten Erfahrungspunkte notwendig. Der Basiswert ist der grundlegende Wert für die Erfahrungspunkte-Berechnung. Seine Erhöhung resultiert in einen sehr hohen Anstieg der benötigten Erfahrungspunkte pro Level des Helden. Der Inflationswert sorgt für einen exponentiellen Anstieg, welcher stärker wird, je höher das Level des Helden ist. Die genaue Formel für die Berechnung der Erfahrungspunkte pro Level des Helden lautet:

$$\text{benötigte Erfahrungspunkte} = \text{Basiswert} * ((\text{nächstes Level} + 3)^{(2,4 + \text{Inflationswert} / 100}) / (5^{(2,4 + \text{Inflationswert} / 100)}))$$

Doch ihr braucht die Formel nicht zu verstehen, denn die benötigten

Erfahrungspunkte sind ja bereits in dem Fenster angegeben.

Im Prinzip ist ein umstellen der Erfahrungspunktekurve ohnehin nicht notwendig. Wir lassen für unseren Helden Alex die Standardeinstellung Basis: 30 und Inflation: 30.

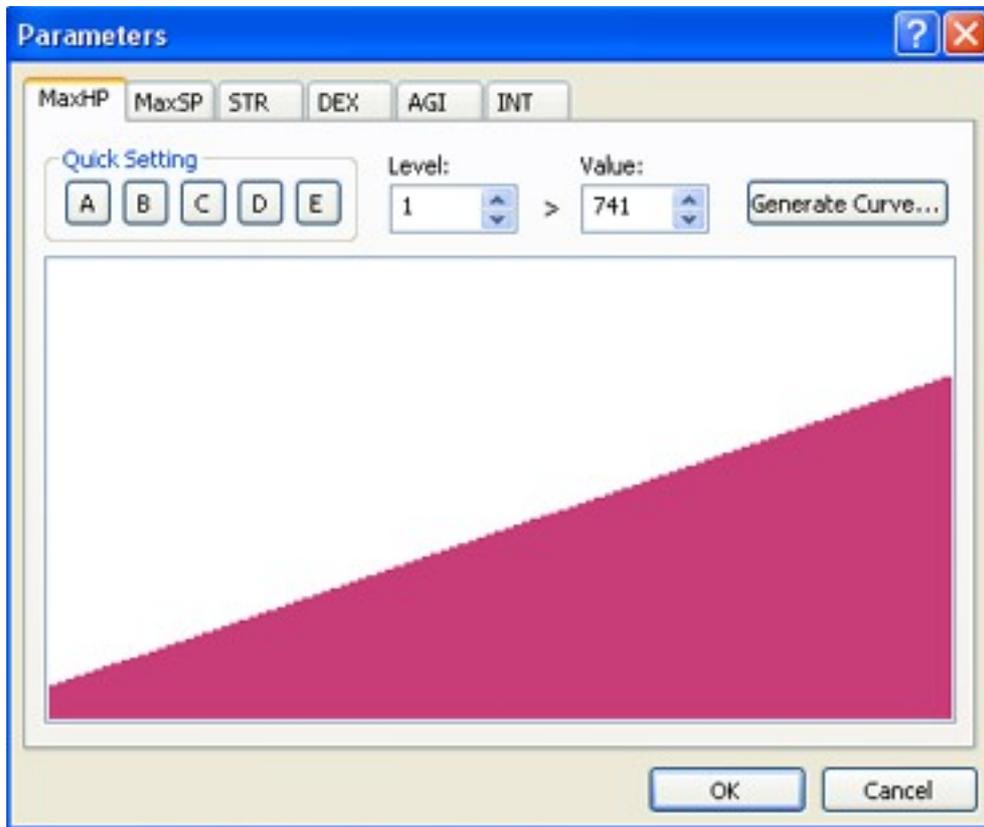
Der nächste wichtige Punkt ist die „Character Graphic“, also die Heldengrafik (5). Hier legt ihr das Aussehen des Helden auf der Map und im Menü. Klickt doppelt auf das große Fenster. Es wird sich nun eine Liste mit allen vorhandenen Characters-Grafiken öffnen. Wählt euch die gewünschte Grafik aus, und klickt auf OK. Für Alex verwenden wir die Heldengrafik „013-Warrior01“.

„Battler Graphic“, also die Kämpfergrafik (6) legt das Aussehen des Helden während des Kampfes fest. Auch hier könnt ihr euch per Doppelklick auf die große, leere Fläche aus einer Liste von Battlers-Grafiken eine Beliebige aussuchen. Damit sie passend zu unserer Heldengrafik ist, wählen wir auch hier die Grafik „013-Warrior01“.

Der nächste Abschnitt des Actors-Fensters trägt die Bezeichnung „Parameters“ (7). Hier wird festgelegt, welche Attribute der Held besitzt, und wie sich diese bei einem Levelanstieg verändern.

Die Helden besitzen sechs verschiedene Attribute: HP (Lebenspunkte), SP (Fertigkeitenpunkte), STR (Stärke), DEX (Geschicklichkeit), AGI (Wendigkeit) und INT (Intelligenz).

Diese Attribute beeinflussen die Stärke des Helden im Kampf. Damit der Held im Laufe des Spiels immer stärker wird, müssen diese Werte beim Levelanstieg ebenfalls aufsteigen. Nur wird ein Magier mehr Intelligenz besitzen als ein Barbar. Während der Dieb schneller und geschickter ist als ein Mönch. Wenn ihr doppelt auf einen der Attribute draufklickt, erscheint ein neues Fenster.

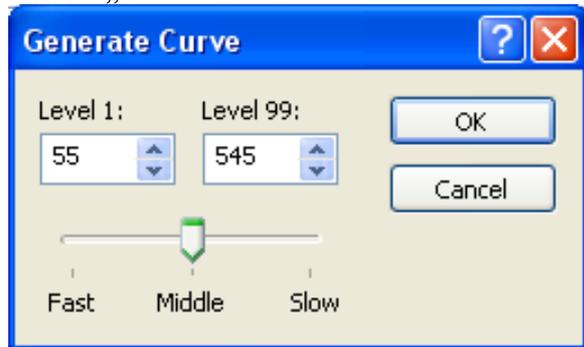


In diesem Fenster ist die Kurve des ausgewählten Attributs noch einmal in vergrößerter Ausführung zu sehen. Die Kurve zeigt die Höhe des jeweiligen Attributs pro Level. Sie beginnt links mit Level 1 und endet rechts mit Level 99. Je höher der Graph geht, desto höher ist das Attribut. Ihr könnt nun den Graphen beliebig verändern. Klickt einfach mit der Maus auf eine Stelle des Graphen, und lasst die Maustaste gedrückt. So lässt sich die Kurve eigenhändig umzeichnen.

Deutlich einfacher und bequemer geht es jedoch, wenn ihr dies dem Computer überlasst. Dies funktioniert auf zweierlei Weise:

Oben links befindet sich ein Feld mit der Bezeichnung „Quick Setting“. Darunter befinden sich Buttons mit den Buchstaben A bis E. Sobald ihr auf einen dieser Buchstabenbuttons klickt, verändert sich die Kurve. Der Button A generiert eine besonders hohe Kurve, während der Button E eine sehr niedrige Kurve generiert. Der Button C ist ein Mittelmaß zwischen beiden Extremen. Unser Held soll ein starker Barbar sein. Deshalb solltet ihr ihm bei dem Attribut *STR* einen Wert mit A generieren, und bei Intelligenz einen Wert durch den E Button. Die anderen Werte lassen wir bei C.

Durch diese Methode steigen die Attribute jedoch nur linear. Das heißt, der Held bekommt nach jedem Levelanstieg dieselbe Menge am jeweiligen Attribut. Die Kurve lässt sich jedoch auch exponentiell generieren. In diesem Fall steigt oder fällt die Menge des Attributzuwachses je höher das Heldenlevel ist. Klickt hierfür auf den Button „Generate Curve“.



Es öffnet sich nun ein neues Fenster. Ihr könnt in dem Feld „Level 1“ das Startattribut beim ersten Level des Helden, und beim Feld „Level 99“ das Endattribut beim höchsten Level des Helden einstellen. Darunter befindet sich ein Regler mit den Bezeichnungen „Fast“ (schnell), „Middle“ (mittel) und „Slow“ (langsam). Befindet sich der Regler auf „Middle“, so verläuft die Kurve linear. Nähert er sich der rechten Seite, so steigt das Attribut stärker, je höher das Heldenlevel ist. Nähert der Regler sich der linken Seite, so steigen die Attribute in den unteren Leveln

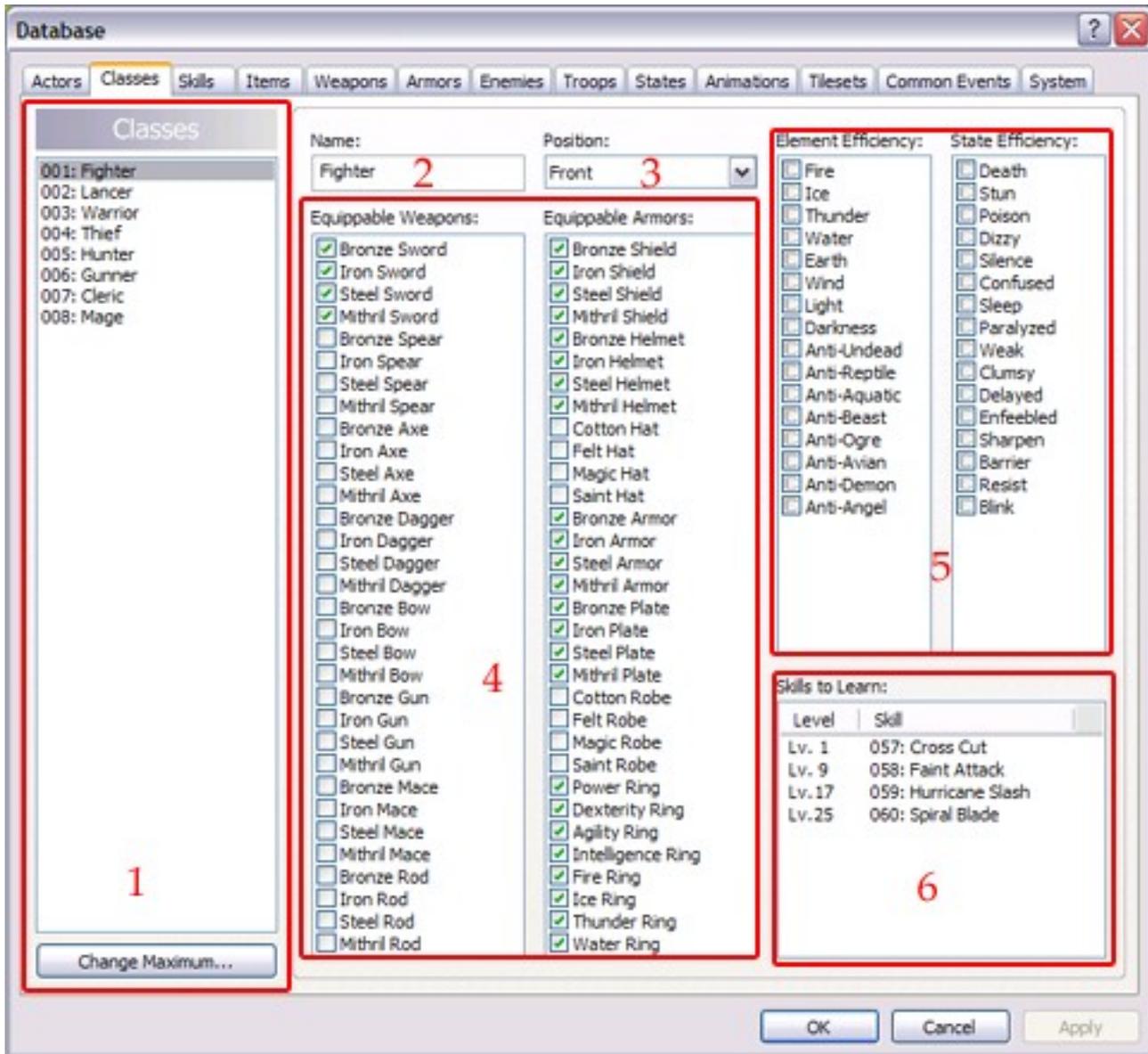
besonders stark, und steigen dann schwächer, je höher das Heldenlevel ist. Sobald ihr die gewünschten Einstellungen vorgenommen habt, klickt auf OK.

Als letztes könnt ihr noch festlegen, mit welchen Waffen und Rüstungen der Held in das Spiel einsteigt. Dies lässt sich in dem Bereich, welcher mit „Starting Equipment“ (7) beschrieben ist, einstellen. Sobald ihr auf die beschrifteten Balken neben *Weapon* (Waffe), *Shield* (Schild), *Helmet* (Helm), *Body Armor* (Rüstung) oder *Accessory* (Zubehör) klickt, erscheint eine Liste mit den jeweiligen Ausrüstungsgegenständen, die der Held tragen kann. Welche Ausrüstungsgegenstände existieren könnt ihr in den Teilabschnitten „Armors“ und „Weapons“ festlegen. Welche Ausrüstungsgegenstände der Held tragen kann, ist von seiner Klasse abhängig, und wird bei „Classes“ eingestellt. Doch dazu kommen wir später noch.

Neben den ausgewählten Ausrüstungsgegenständen befindet sich noch eine kleine Box, neben der die Bezeichnung „Fixed“ (festgesetzt) steht. Wenn ihr die Box anklickt, so erscheint ein Häkchen darin. Damit habt ihr den jeweiligen Ausrüstungsgegenstand festgesetzt. Das heißt, der Spieler kann diese Ausrüstung nicht mehr ablegen. Dies eignet sich besonders bei Charakteren, die nur kurz im Spiel vorkommen. Damit der Spieler ihnen nicht ihre Ausrüstungsgegenstände abnehmen kann, müssen sie mit einem Häkchen bei „Fixed“ belegt werden.

Classes

Jeder Held gehört einer Klasse an. Damit ihr eurem Helden eine eigene Klasse erstellen könnt, müsst ihr den Abschnitt „Classes“ aufsuchen. Nach einem Klick auf „Classes“ erscheint folgendes Dialogfenster.



In der linken Spalte (1) befindet sich wieder eine Liste, der Heldenliste bei „Actors“ nicht unähnlich. Diesmal sind jedoch die Klassen aufgezählt. Das Hinzufügen und Löschen von Klassen funktioniert aber auf die gleiche Weise: Legt mit einem Klick auf „Change Maximum“ eine höhere Klassenzahl als zuvor fest, und wählt dann eine neue, unbeschriebene Klasse aus. Oder wählt eine existierende Klasse aus, und klickt die Entfernen-Taste.

Rechts neben der Spalte befinden sich die Informationen dieser Klasse. Zuerst legt ihr in dem Feld „Name:“ (2) den Namen für eure Klasse fest. Wir nennen sie in diesem Tutorial mal „Barbar“. In dem Feld „Position“ (3) könnt ihr einstellen, ob der Held eher ein Frontkämpfer ist, oder sich lieber im Hintergrund hält. Steht der Wert auf „Front“ (Vorne), so wird der Charakter häufiger getroffen, als wenn er auf „Middle“ (Mitte) oder „Rear“ (Hinten) steht. Magier, Bogenschützen und Kleriker sollten eher im Hintergrund, also bei „Rear“ stehen. Unser Barbar jedoch, sollte auf „Front“ gesetzt werden, denn er ist ein Nahkämpfer.

In den nächsten zwei Listen seht ihr wieder die Boxen, in denen ihr mit einem Klick ein Häkchen hinzufügen, und mit einem erneuten Klick das Häkchen wieder entfernen könnt.

Die beiden Listen (4) zählen die vorhandenen (in dem Abschnitt „Weapons“ und „Armors“ erstellten) Waffen und Rüstungen auf. Setzt ihr neben dem Namen einer Waffe, oder einer Rüstung ein Häkchen, so erhalten alle Helden, welche dieser Klasse angehören, die Fertigkeit diese Waffe oder Rüstung zu tragen.

Unser Barbar zum Beispiel sollte mit Axtwaffen kämpfen. Wir setzen deshalb bei den Einträgen „Bronze Axe“, „Iron Axe“, „Steel Axe“ und „Mythril Axe“ (Bronze-, Eisen-, Stahl- und Mythrilaxt) ein Häkchen. Bei den anderen Einträgen unter „Equippable Weapons:“ entfernen wir dafür die Häkchen. Somit ist unser Barbar nur in der Lage, Axtwaffen zu tragen. Wie man eigene Axtwaffen hinzufügt, oder die bestehenden Axtwaffen ins deutsche übersetzt, erfahren wir im nächsten Abschnitt. Bei den Rüstungen machen wir genau dieselbe Prozedur durch. Bei den Rüstungen, die der Barbar tragen darf, setzen wir ein Häkchen. Die Rüstungen, die er nicht tragen soll, bekommen kein Häkchen.

Die nächsten beiden Listen (5) sind anders aufgebaut. Zwar gibt es auch hier wieder eine kleine Box neben den aufgelisteten Begriffen, doch sind diese Boxen mit den Buchstaben A, B, C, D, E, F beschriftet. Sobald ihr eine Box anklickt, ändert sich ihr Buchstabe.

Diese Listen legen die Stärken und Schwächen eurer Klasse fest. Allgemein gilt: C ist der Mittelwert. Das heißt: Die Klasse ist gegenüber dem jeweiligen Wert weder schwach noch stark. B, und A kennzeichnen eine Schwäche; D, E und F kennzeichnen eine Stärke gegenüber dem jeweiligen Wert.

Die linke Spalte „Element Efficiency:“ legt die Stärken und Schwächen der Helden, welcher dieser Klasse angehören, gegenüber Elementen fest. Wie üblich habt ihr auch hier wieder eine Liste mit Elementen. Neue Elemente könnt ihr im Abschnitt „System“ erstellen.

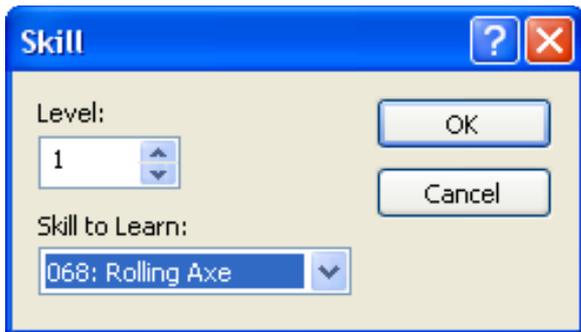
Fertigkeiten, Zaubersprüche sowie Waffenangriffe können einem (oder sogar mehreren) Element angehören. Damit unser Barbar sehr empfindlich gegenüber Feuerangriffen reagiert, klicken wir solange auf die Box neben „Fire“ (Feuer), bis ein A erscheint. Dann erhält der Barbar doppelten Schaden bei Feuerangriffen. Gegen Donnerangriffe soll der Held nur eine leichte Schwäche haben. Deshalb klicken wir bei der Box neben „Thunder“ (Donner) solange drauf, bis ein B erscheint. Der Barbar erhält bei Donnerangriffen nun 150% mehr Schaden. Ein Donnerangriff der vorher 50 Schaden verursacht hätte, würde nun 75 Schaden verursachen. Gegen Erde soll der Barbar keine Schwäche haben. Deshalb setzen wir in der Box bei „Earth“ ein C. Der Barbar erhält nun weder mehr, noch weniger Schaden bei Erdangriffen. Würden wir den Wert auf D setzen, so würde der Barbar nur noch die Hälfte des Schadens erleiden. Da unser Barbar ein guter Schwimmer ist, setzen wir neben „Water“ (Wasser) ein E. Nun ist der Barbar immun gegen Wasserangriffe. Er erleidet gar keinen Schaden mehr. Doch da der Barbar aus dem kalten Norden kommt, soll er Eisangriffe absorbieren können. Wird neben „Ice“ ein F gesetzt, so wird der Barbar durch Eisangriffe sogar geheilt, statt Schaden zu erleiden.

Die rechte Spalte ist für die Zustände, welche im „States“ Abschnitt eingestellt werden. Das Prinzip ist hier ähnlich: Wird ein Charakter zum Beispiel mit einem Giftzauber belegt, so ist seine Schwäche bzw. Stärke gegen das Gift entscheidend, ob der Charakter in den „Venom“-Zustand (Giftzustand) verfällt. Bei A ist diese Wahrscheinlichkeit bei 100%. Der Charakter wird sofort vergiftet. Bei B ist die Wahrscheinlichkeit nur 80%, bei C 60%, bei D 40% und bei E nur 20%. Damit der Charakter immun gegen einen Zustand wird, muss F ausgewählt werden.

Vielleicht wundert ihr euch, das sich bei den States auch der Zustand „Knockout“ (außer Gefecht gesetzt, also Tod/Bewusstlos) befindet. Was passiert wenn dieser Zustand auf F, also auf immun ist? Kann der Charakter dann nicht sterben? Doch, er kann. Fällt die HP des Helden unter 0, so stirbt er auf jeden Fall. Die Einstellungen bei „State Efficiency:“ bezieht sich nur auf spezielle Fertigkeiten und Zauber, die den Helden mit dem Zustand belegen. So gibt es in Spielen wie Final Fantasy zum Beispiel Todeszauber, welche einen Gegner mit einem Schlag vernichten. Die Wahrscheinlichkeit, mit der solche Zauber wirken, wird hier festgelegt. Des Weiteren gibt es Zustände, die unabhängig von Resistenzen wirken. Doch dazu später im Abschnitt „States“.

Das letzte Fenster trägt die Bezeichnung „Skills to learn“ (6). Hier wird festgelegt, welche Fertigkeiten und Zauber eine Klasse erlernen kann. Unser Barbar sollte einige Kampffertigkeiten erlernen. Sobald ihr doppelt auf die weiße

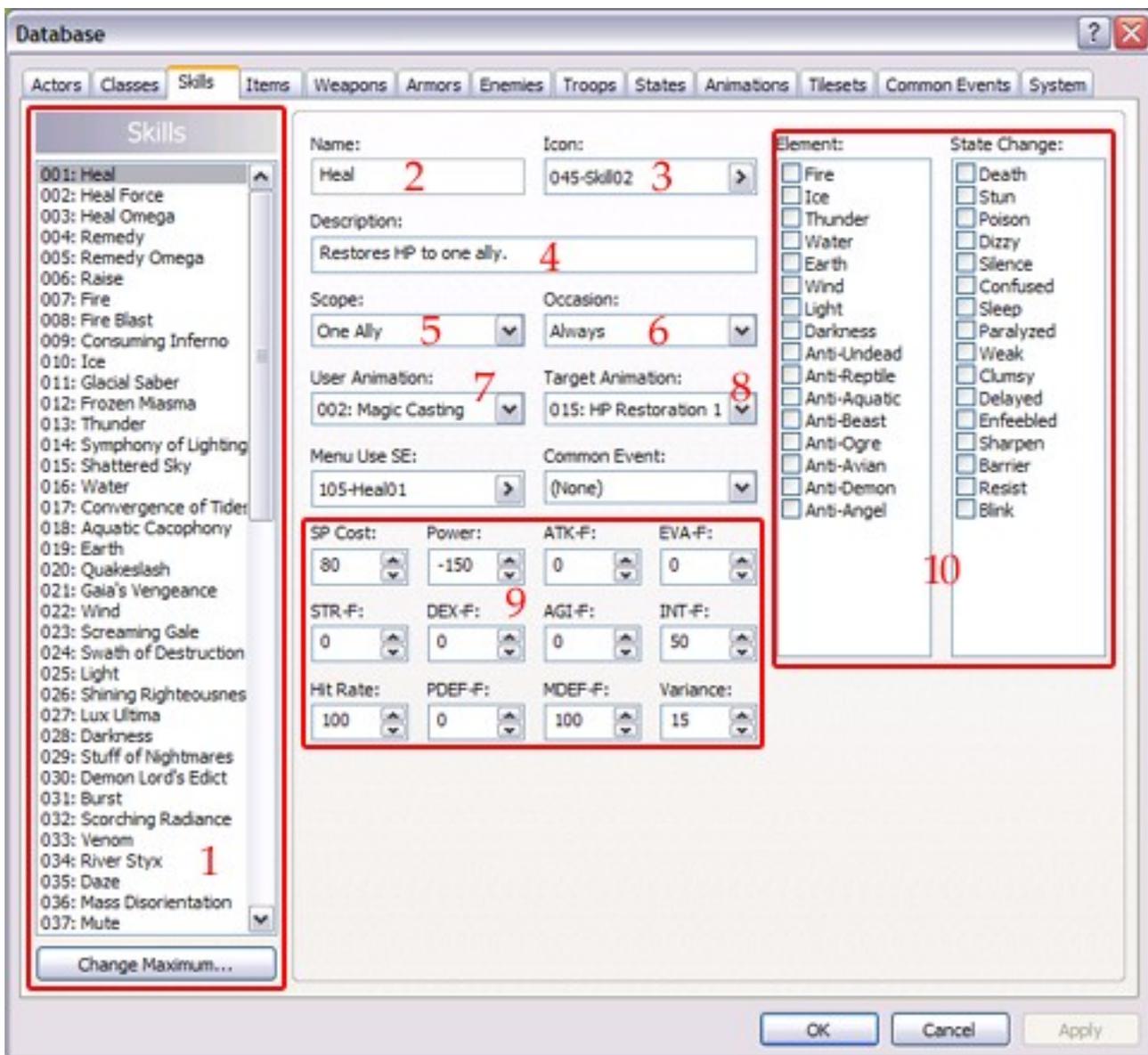
Fläche dieses Fenster klickt, erscheint ein neues Dialogfenster.



Durch einen Klick auf „Skill to Learn“ erscheint eine Liste mit allen Fertigkeiten, welche im Abschnitt „Skills“ erstellt wurden. Wir wählen uns nun die 68te Fertigkeit mit dem Namen „Rolling Axe“ (rollende Axt) aus. Bei „Level:“ geben wir die Levelnummer ein, in der die Klasse die Fertigkeit erlernen soll. Damit unser Barbar mit Level 10 die rollende Axt erlernt, geben wir also 10 ein. Ihr könnt einer Klasse beliebig viele Fertigkeiten hinzufügen. Doch um neue Fertigkeiten zu erstellen, müsst ihr den Abschnitt „Skills“ aufsuchen.

Skills

Klickt also auf den Abschnitt „Skills“ und ihr gelangt in das folgende Dialogfenster:



Links befindet sich wieder die Liste (1) der bestehenden Skills (Fertigkeiten). Um eine neue Fertigkeit hinzuzufügen, durchlaufen wir dieselbe Prozedur wie vorhin: Ein Klick auf Change Maximum, eine hohe Zahl (z.B. 85) eingeben, nach unten scrollen und eine leere Fertigkeit anwählen. Um der Fertigkeit einen Namen zu geben, nutzen wir das Feld „Name:“ (2). Wir wollen für unseren Barbaren eine neue Axttechnik entwerfen.

Wir nennen sie „Axtwurf“. In dem Feld „Icons:“ (3) können wir uns nun eine Grafik auswählen, die im Menü und im Kampf für diese Fertigkeit angezeigt wird. Passend zu einer Axttechnik wählen wir die Grafik „003-Weapon03“. Bei „Description“ (4) schreiben wir einen kurzen Informationstext, der erklärt, was diese Technik bewirkt. Er wird ebenfalls im Menü und im Kampf angezeigt. Wir schreiben in das Feld den Text „Die Axtwurftechnik. Effektiv gegen fliegende Gegner!“.

„Skope“ (5) legt fest, wen die Fertigkeit betrifft. Man kann hier zwischen none (Niemanden), one enemy (einen Gegner), all enemies (alle Gegner), one Ally (einem Teammitglied), all Allies (allen Teammitgliedern), one ally HP 0 (einem besiegten Teammitglied), all allies HP 0 (allen besiegten Teammitgliedern), the user (den Anwender der Fertigkeit) wählen. Da der Axtwurf einen Gegner treffen soll, wählen wir „one enemy“. Übrigens können auch Gegner Fertigkeiten erlernen. Würde ein Gegner die Fertigkeit Axtwurf einsetzen, so würde sich das „one enemy“ auf einen Helden der Heldentruppe beziehen. Das Ziel der Fertigkeit ist also stets aus dem Blickwinkel des Anwenders zu sehen.

Bei Occasion stellen wir ein, wann man die Fertigkeit einsetzen darf. „Always“ (immer) kennzeichnet Fertigkeiten, welche jederzeit eingesetzt werden dürfen. Dies wären zum Beispiel Heilzauber, welche sowohl vom Menü aus, wie auch während des Kampfes eingesetzt werden dürfen. Eine Fertigkeit die auf „Only in Battle“ (nur im Kampf) gestellt ist, darf nur während eines Kampfes eingesetzt werden. Dies trifft auf die normalen Kampfzauber und Kampftechniken zu. Unser Axtwurf ist eine solche Kampftechnik. Also wählen wir in diesem Dialog „Only in Battle“ aus. Es gibt auch noch die Auswahlmöglichkeiten „Only from the menu“ (nur im Menü), welche eine Fertigkeit nur während des Menüs einsetzbar macht, und „never“, bei dem eine Fertigkeit nie eingesetzt werden kann.

Bei „User Animation“ (7) wählen wir die Animation aus, welche am Helden beim Einsetzen der Fertigkeit eingeblendet werden soll. Ein Axtwurf bedarf keiner besonderen Animation, deshalb lassen wir dieses Feld auf „None“.

Bei „Target Animation“ (8) suchen wir uns dagegen eine Animation aus, die am Gegner, den die Technik trifft, abgespielt wird. Glücklicherweise gibt es bereits eine große Auswahl an Animationen. Wir verwenden die Animation „075: Axe-skill 1“. Wie man neue Animationen erstellt, erfährt ihr im Abschnitt „Animations“.

Bei „Menu Use SE“ könnten wir einstellen, welches Geräusch ertönt, wenn die Fertigkeit im Menü eingesetzt wird. Da wir aber eingestellt haben, dass der Axtwurf nur im Kampf benutzt werden darf, lassen wir dieses Feld frei.

Unter „Common Event“ kann man ein CommonEvent auswählen, welches beim benutzen der Fertigkeit aufgerufen wird. Dadurch kann man einer Fertigkeit zusätzliche Funktionen verschaffen, die in dem Abschnitt „Skills“ nicht einzustellen sind. Zum Beispiel das sich beim Verwenden der Fertigkeit die Heldengrafik verändert, oder das der Held seine Klasse wechselt usw. Durch diese Funktion lassen sich Fertigkeiten quasi endlos ausbauen. Allerdings wollen wir nur eine einfache Kampftechnik entwerfen, bei der es keiner Common Events bedarf. Deshalb lassen wir auch dieses Feld frei.

Der nächste große Bereich, der mit der Ziffer 9 gekennzeichnet ist, gibt Aufschluss über die Werte und Parameter der Fertigkeit. Bei SP Cost (Fertigkeitspunkte-Kosten) wird die benötigte Anzahl der Fertigkeitspunkte eingegeben. Da unser Barbar wohl eher wenige Fertigkeitspunkte besitzt, werden wir diese Zahl niedrig halten. Wir geben also 50 ein.

„Power“ gibt die Stärke der Fertigkeit an. Wird eine positive Zahl eingegeben (z.B. 150), so ist die Fertigkeit eine Kampftechnik die Schaden zufügt. Bei einer negativen Zahl (z.B. -100) jedoch, wirkt die Fertigkeit heilend. Ein Axtwurf fügt jedoch eindeutig Schaden zu. Wir geben also die Zahl 25 ein.

Bei „ATK-F“ (Einfluss des Angriffsschadens) können wir angeben, wie stark die Fertigkeit vom Angriffsschaden des Anwenders beeinflusst wird. Der Angriffsschaden ist stets von der Waffe, nicht von den Attributen des Charakters abhängig. Da ein Axtwurf definitiv von der Qualität der geworfenen Axt abhängt, geben wir die Zahl 100 ein. Bei einem Zauber (z.B.. Feuerball), der ja unabhängig von der getragenen Waffe ist, muss immer die Zahl 0 eingegeben werden!

Gleich daneben befindet sich ein Fenster mit der Bezeichnung „*EVA-F*“ (Ausweichsbeeinflussung). Hier wird eingestellt, inwiefern die Trefferwahrscheinlichkeit der Fertigkeit von der Geschicklichkeit des Anwenders und der Wendigkeit des Gegners abhängt. Bei Zaubern ist dieser Wert normalerweise 0. Bei Kampftechniken wie unserem Axtwurf geben wir jedoch die Zahl 100 ein. Denn somit trifft unser Barbar mit seinem Axtwurf häufiger, wenn seine Geschicklichkeit hoch, und der Gegner nicht besonders wendig ist.

Die nächsten vier Felder in der zweiten Zeile verhalten sich allesamt gleich. Sie geben an, wie sehr die Stärke der Fertigkeit von dem jeweiligen Attribut des Anwenders abhängig ist. Bei Zaubern müssten wir den Wert „*INT.-F*“ auf 100 setzen. Der Schaden unseres Axtwurfes hingegen sollte von der Stärke und Geschicklichkeit des Barbaren abhängen. Wir setzen daher bei „*STR-F*“ den Wert 100, und bei „*DEX-F*“ den Wert 50.

Nun haben wir die dritte und letzte Zeile an Parametern vor uns. „Hit-Rate“ (Trefferwahrscheinlichkeit) gibt die prozentuale Wahrscheinlichkeit an, mit der die Fertigkeit gelingt bzw. ihr Ziel trifft. Wir ihr euch sicher erinnert, haben wir uns ja schon bei dem Parameter „*EVA-F*“ mit der Trefferwahrscheinlichkeit auseinandergesetzt. Was ist nun der Unterschied zwischen „*EVA-F*“ und „Hit-Rate“? Beide Werte entscheiden über die Trefferwahrscheinlichkeit der Fertigkeit. „Hit-Rate“ gibt direkt an, mit welcher Wahrscheinlichkeit eine Fertigkeit gelingt. Doch dieser Wert wird von der Geschicklichkeit des Anwenders, und der Wendigkeit des Feindes beeinflusst, je höher der Wert bei „*EVA-F*“ ist. Für unseren Axtwurf ist nur die Geschicklichkeit bzw. Wendigkeit wichtig. Deshalb setzen wir Hit-Rate auf 100. Die Attacke trifft normalerweise immer, doch diese 100% können durch die Geschicklichkeit/Wendigkeit beeinflusst werden. Unser Held wird also bei besonders schnellen Gegnern eine geringere Trefferchance haben.

Der Wert „*PDEF-F*“ gibt an, wie stark die physische Verteidigung des Ziels die Wirkung der Fertigkeit verringert, während „*MDEF-F*“ angibt, wie stark die magische Verteidigung des Ziels die Fertigkeit mindert. Da ein Axtwurf sich wohl kaum durch eine starke Zauberessistenz abschwächen lässt, lassen wir den Wert bei „*MDEF-F*“ auf 0, und bei „*PDEF-F*“ bei 100.

Der letzte Wert trägt die Bezeichnung „Variance“. Dieser Wert gibt die maximale prozentuale Abweichung des Schadens vom Idealschaden an. Das heißt: Wenn unser Axtwurf bei einem Gegner 50 Schaden machen würde, und wir eine „Variance“ von 20 eingestellt haben, dann kann unser Schaden um bis zu 20% zum positiven oder negativen abweichen. Also würde die Technik letztendlich 40 bis 60 Schaden machen. Bei einer „Variance“ von 50 würde der Axtwurf einen Schaden von 25 bis 75 machen. Je größer die „Variance“, desto größer die Abweichung bei der Schadensberechnung. Wir lassen diesen Wert standardmäßig auf 15.

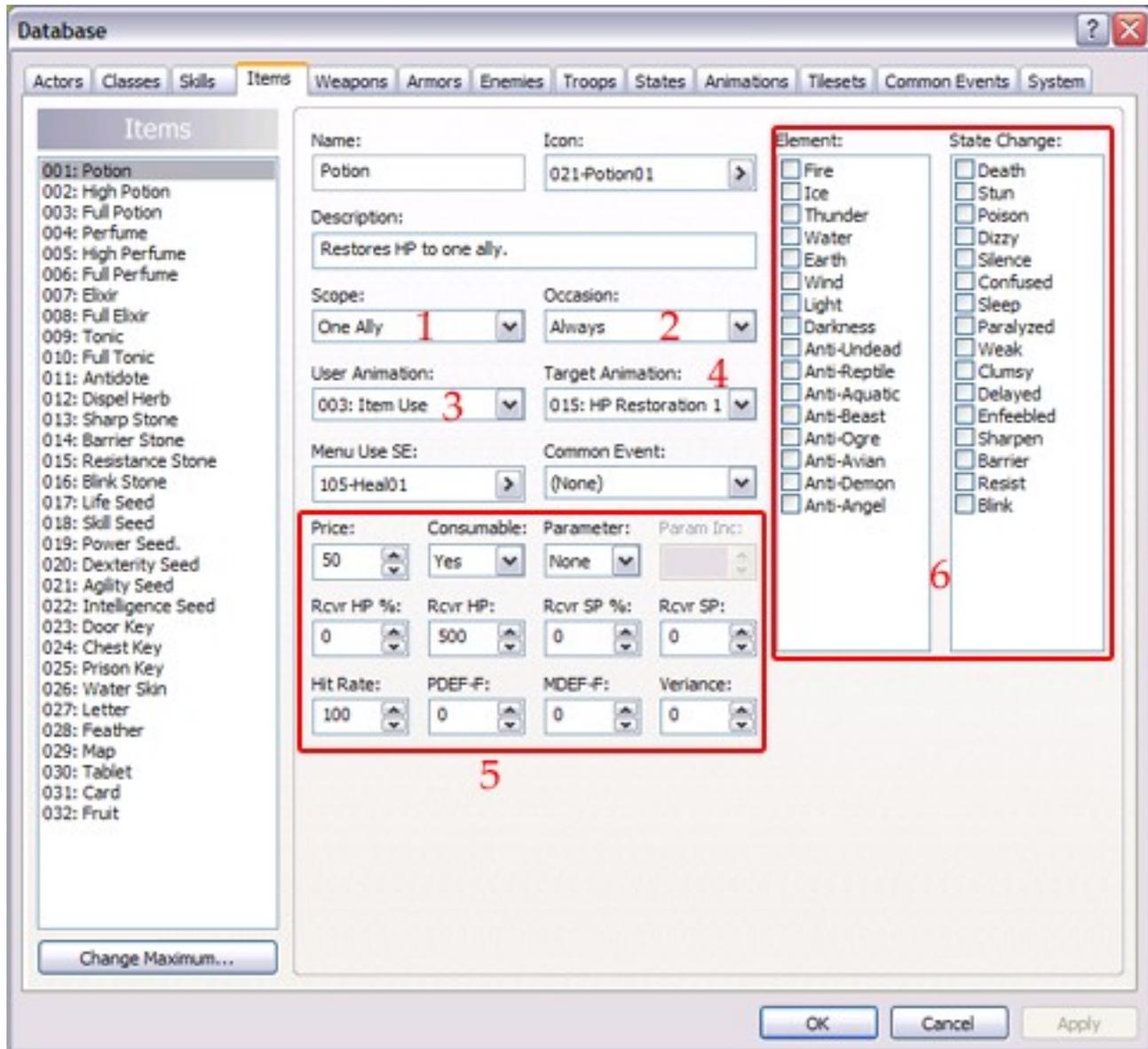
Nun haben wir alle Parameter ausgefüllt, und unser Fertigkeitenfenster ist nahezu vollständig ausgefüllt. Nun fehlen uns nur noch die beiden Spalten auf der rechten Seite (10). Sie tragen die Bezeichnung „Element:“ und „State Change“ (Zustandsänderung). Bei „Element“ sind sämtliche Elemente, welche im Abschnitt „System“ erstellt wurden, aufgelistet. Wir haben im Abschnitt „Classes“ bereits die Resistenz der Barbarenklasse gegenüber den Elementen eingestellt. Hier müssen wir auswählen, welchen Elementen die Technik angehört. Unsere Axtwurftechnik soll besonders gut gegen fliegende Gegner wirken. Deshalb setzen wir einen Haken bei dem Element „vs Bird“ (gegen Vögel). Nun fügt der Axtwurf bei allen Feinden, welche eine Schwäche gegen das Element „vs Bird“ besitzen, zusätzlichen Schaden zu.

Bei „State Change“ können wir einstellen, welche Zustandsänderungen die Waffe hervorruft. Die Zustandsänderungen können wir im Abschnitt „States“ erstellen. Doch wir haben ja bereits eine reichliche Auswahl an Zuständen. Wenn die Axt den Feind schwächen soll, dann setzen wir ein Kreuz bei „Weaken“. Ob der Feind durch den Axtwurf tatsächlich vom Schwäche-Zustand befallen wird, hängt von seiner Resistenz gegenüber diesem Zustand ab. Wenn wir ein zweites Mal auf diesen Zustand klicken, erscheint ein Minus. Das kennzeichnet, dass unsere Axttechnik den Gegner vom Schwäche-Zustand heilen würde. Bei einem Heilzauber gegen Gifte müssten wir also ein Minus auf den Zustand „Venom“ setzen. Da wir allerdings eine Axttechnik verwenden, belassen wir es bei dem Plus beim „Weaken“-Zustand.

Nun haben wir unsere Fertigkeit voll ausgefüllt. Damit unser Barbar diese Fertigkeit erlernt, müssen wir zurück in den Abschnitt „Classes“. Hier sehen wir unten rechts wieder das Fenster „Skills to learn“. Wenn wir doppelt auf die weiße Fläche klicken, erscheint eine Auswahl aller Fertigkeiten. Wenn wir etwas nach unten scrollen, sehen wir auch unsere neue Fertigkeit „Axtwurf“. Wählt diese Fertigkeit aus, und gebt bei Level den Wert 1 aus. Nun verfügt unser Barbarencharakter die Fertigkeit Axtwurf.

Items

Damit unser Barbarencharakter die Kämpfe gegen seine Feinde auch überlebt, müssen wir ihn mit starken Heiltränken und anderen nützlichen Items ausstatten. Dafür suchen wir den Abschnitt „Items“ auf.



Auch das Itemfenster ist nahezu auf die gleiche Weise aufgebaut, wie der „Skills“-Abschnitt zuvor. Links sehen wir wieder die bekannte Liste mit den aufgelisteten Elementen. Diesmal handelt es sich um Items. Statt ein neues Item zu erstellen, werden wir uns diesmal ein bestehendes Item anschauen und editieren. Wir verwenden gleich das erste Item: Potion. Jeder FinalFantasy-Spieler wird wissen, dass es sich bei einer Potion um einen Heiltrank handelt. Dennoch werden wir im Feld „Name:“ dieses Item in „Heiltrank“ umbenennen. Bei „Icon:“ können wir wieder ein Bild für dieses Item auswählen. Doch belassen wir das bereits eingestellte Bild. Genau wie bei „Skills“ gibt es auch hier wieder ein „Description“-Eingabefeld. Hier wird die Beschreibung des Items eingegeben.

Wir schreiben „Heilt 500 Lebenspunkte“ in das Feld. Die nächsten sechs Felder verhalten sich genauso wie bei den Fertigkeiten.

Unter „Skope:“ (1) wird ausgewählt, wen das Item betrifft.

Bei „Occassion“ (2) wählen wir, wann das Item eingesetzt werden darf.

„User Animation“ (3) und „Target Animation“ (4) geben Aufschluss darüber, welche Animation beim Einsatz des Items eingeblendet wird.

Unter „Menu Use SE“ wird der Soundeffekt eingestellt, welcher beim Einsatz des Items im Menü abgespielt werden soll.

Zum Schluss gibt es noch das Feld „Common Events:“. Hier können wir ein Common Event auswählen, welches beim Einsatz des Items aktiviert werden soll. Dadurch können wir den Items zusätzliche Effekte verpassen. Zum Beispiel das beim Benutzen des Items „Landkarte“ das Bild einer Landkarte auf den Bildschirm angezeigt werden soll, etc.

Alle diese Felder belassen wir erstmal ihren Standardeinstellungen.

Als nächstes kommt wieder eine Reihe von Parametern. Wir werden kurz die einzelnen Parameter (5) durchgehen. Allerdings müssen wir an den bestehenden Werten nichts ändern. Auch sie belassen wir bei ihren Standardeinstellungen.

Bei „Price:“ (Preis) geben wir die Summe ein, die der Spieler in einem Itemladen zahlen muss, um das Item zu kaufen. „Consumable“ (Verbrauchbar) legt fest, ob das Item nach seiner Nutzung verschwinden soll. Bei einem Heiltrank muss hier natürlich „Yes“ (Ja) stehen, denn nach dem Benutzen eines Heiltranks soll dieser auch verschwinden. Ansonsten könnte man einen Heiltrank ja unendlich oft benutzen. Bei einer Landkarte stattdessen müsste „No“ (Nein) ausgewählt werden, denn eine Landkarte kann man schließlich so oft benutzen, wie man will. Der nächste Wert trägt die Bezeichnung „Parameter:“ (Attribut). Hier kann man sich ein Attribut des Helden auswählen. Sobald ein Attribut ausgewählt wurde, erscheint ein danebenliegendes Feld mit der Bezeichnung „Param Inc:“ (Attributssteigerung). Hier kann man eine Zahl wählen, um ein Attribut des Helden bei der Benutzung des Items steigen zu lassen. So könnten wir zum Beispiel das Item „Lebenselexier“ erstellen, welches die maximalen Lebenspunkte des Helden erhöhen soll. Dann müssten wir bei „Parameter:“ die Auswahl „Max HP“, und bei „Param Inc:“ zum Beispiel 20 eingeben. Natürlich kann man auch negative Zahlen eingeben, um einen Parameter zu senken.

Doch da unser Heiltrank nur heilen soll und die maximalen Lebenspunkte nicht verändert, belassen wir die „Parameter:“ Auswahl bei „None“ (keinen).

In der zweiten Zeile befinden sich vier Felder, welche alle die Bezeichnung „Rcvr“ (Wiederherstellung) und einen Parameter tragen. Hier kann man einstellen, ob das Item die Lebenspunkte (HP), oder die Fertigkeitenpunkte (SP) wiederherstellen soll. Bei den Feldern, die zusätzlich dieser Bezeichnung noch ein „%“ tragen, ist der eingegebene Wert prozentual gesehen. Wenn wir also ein Heilitem erstellen möchten, das den Helden 50% seiner Lebenspunkte zurückgibt, so geben wir in das Feld „Rcvr HP %“ den Wert 50 ein. Soll das Item jedoch dem Helden 100 Lebenspunkte zurückgeben, so müssen wir in das Feld „Rcvr HP“ den Wert 100 eingeben. Natürlich lässt sich auch beides kombinieren. Dann würde das Heilitem 50% der Lebenspunkte des Helden regenerieren, und zusätzlich noch 100 Lebenspunkte heilen.

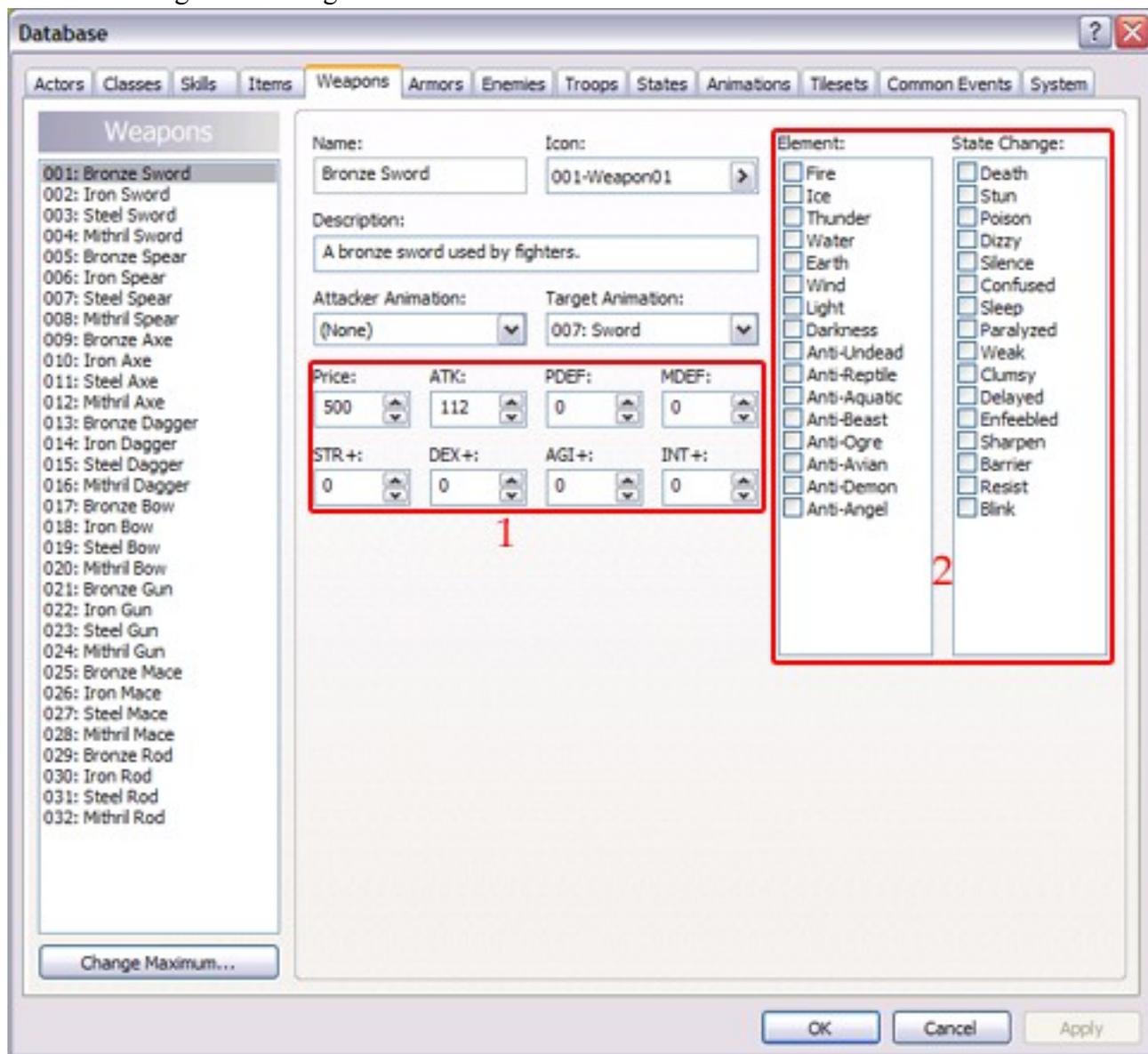
Außerdem kann man auch Kampfitems (z.B.. Eine Handgranate) erstellen, in dem man bei diesen Werten eine negative Zahl eingibt. Dann werden Lebenspunkte (oder Fertigkeitenpunkte) abgezogen, statt diese zu regenerieren.

Die letzte Zeile enthält wieder dieselben Felder wie bei den Fertigkeiten. „Hit-Rate“ gibt an, mit welcher Wahrscheinlichkeit der Einsatz des Items gelingt. „PDEF-F“ und „MDEF-F“ legen fest, wie stark die Wirkung des Items von der physischen oder magischen Verteidigung des Ziels abhängt.

Das letzte Feld (mit dem hübschen Rechtschreibfehler) trägt die Bezeichnung „Veriance“. Natürlich ist damit „Variance“ gemeint. Hier wird festgelegt, wie stark die Wirkung des Items durch den Zufall abweichen kann. Die zwei Spalten rechts (6) funktionieren wieder genauso wie bei den Skills. Bei „Elements“ wird festgelegt, welchem Element das Item angehört. Unter „States“ wird festgelegt, ob das Item einen Zustand heilt (Plus), oder das Ziel mit diesem Zustand belegt (Minus).

Weapons

Als nächstes werden wir eine Waffe für unseren Barbaren entwerfen. Öffnet den Abschnitt „Weapons“. Es erscheint nun folgender Dialog:



In der linken Spalte finden wir wieder eine Liste aller Waffen. Fügt über „Change Maximum“ eine neue Waffe der Liste hinzu, und wählt diese an. Die Felder „Name:“, „Icon:“, „Description“, „Attack Animation:“, „Target Animation:“ sind ja mittlerweile bekannt. Nennt diese Waffe „Streitaxt“, und gebt ihr ein passendes Icon, sowie eine beliebige Animation.

Nun gibt es wieder acht Parametereingaben (1), welche über die Qualität der Waffe entscheiden. Viele dieser Parameter dürften bereits aus den früheren Abschnitten bekannt sein.

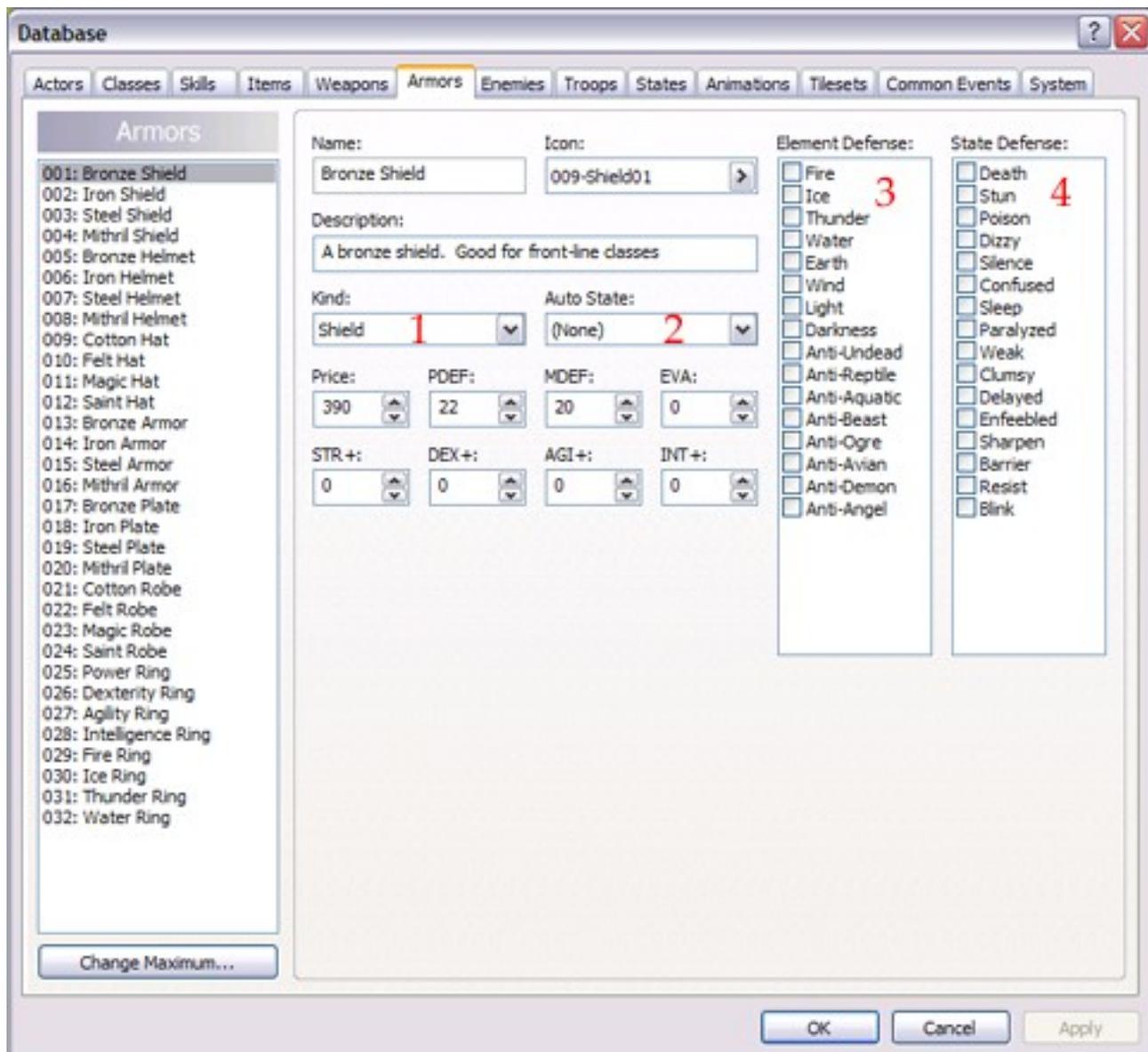
„Price:“ gibt die Kosten der Waffe beim Händler an. „ATK:“ gibt den Angriffsschaden der Waffe an. Dieser Wert ist für die Stärke der Waffe entscheidend. „PDEF:“ und „MDEF“ geben an, ob die Waffe die physische und magische Verteidigung des Spielers erhöht. Würden wir zum Beispiel eine Runenklinge als Waffe erstellen, so könnten wir den Wert bei „MDEF“ auf z.B. 25 setzen. Die Runenklinge würde nun die magische Verteidigung des Charakters verbessern. Doch im Allgemeinen sind Waffen zum Angriff, und weniger zur Verteidigung gedacht. Daher lasst beide Parameter auf 0.

In der zweiten Zeile befinden sich vier Parameter: „STR+“ gibt an, ob die Waffe die Stärke des Trägers erhöht. „DEX+“ steht für Geschicklichkeitserhöhung, „AGI+“ für Wendigkeitserhöhung und „INT+“ für Intelligenzserhöhung. Normalerweise bleiben diese Werte auf 0. Doch vielleicht fügt ihr ja mal ein besondere Artefaktwaffe hinzu, welche zum Beispiel die Intelligenz des Trägers erhöht. Wiederum lassen wir aber alle Werte auf 0.

Rechts befinden sich wieder die zwei Spalten mit „Elements“ und „State Change“ (2). Sie funktionieren auf die gleiche Weise wie bei den Fertigkeiten. Durch „Elements“ wird der Waffe ein Element zugewiesen, durch ein Plus in „States“ verursacht die Waffe womöglich eine Zustandsänderung beim Gegner.

Armors

Nun erstellen wir noch eine passende Rüstung für unseren Barbaren. Öffnet den Abschnitt „Armors“ (Rüstungen). Ihr solltet nun dieses Fenster sehen:



Der Dialog ähnelt wiederum dem „Weapons“-Abschnitt in vielerlei Hinsicht. Es gibt nur wenige kleine Änderungen. Erstellt über „Change Maximum“ eine neue Rüstung. Wir nennen sie „Lederwams“, geben ihr ein passendes Icon, sowie die Beschreibung „Eine feste Lederrüstung“. Neu ist das Feld „Kind:“ (1). Hier legt ihr fest, von welcher Art die Rüstung ist. Ein Held kann vier verschiedene Arten von Rüstungen auf einmal tragen. Diese Arten sind *Helmet* (Helm), *Shield* (Schild), *Body Armor* (Rüstung), *Accessory* (Zubehör).

Unser Lederwams gehört eindeutig zum Typ *Body Armor*. Daher stellen wir dies bei „Kind:“ ein. Das nächste Feld trägt die Bezeichnung „Auto State“ (2). Hier könnt ihr einen Zustand aus der Liste auswählen. Der ausgewählte Zustand ist daraufhin immer aktiv, wenn man die Rüstung trägt. Dies ist praktisch, wenn man zum Beispiel eine verfluchte Rüstung erstellt, welche den Helden mit einer negativen Zustandsänderung belegt.

Die übrigen Felder sollten euch bereits aus den früheren Abschnitten bekannt sein. „Price“ legt die Kosten der Rüstung fest, „PDEF“ legt die physische Verteidigung der Rüstung (wir wählen für unsere Rüstung den Wert 75), und „MDEF“ die magische Verteidigung der Rüstung fest (diesen, sowie alle weiteren Werte, belassen wir bei 0). „EVA“ bestimmt, ob die Rüstung die Ausweichrate des Trägers steigert (durch einen positiven Wert), oder senkt (durch einen negativen Wert). Die Ausweichrate legt fest, wie gut ein Held einer gegnerischen Attacke ausweichen kann.

Die Parameter in der zweiten Zeile verhalten sich genauso wie bei Weapons.

Bei der rechten Spalte seht ihr die Begriffe „Element Defense:“ (Elementverteidigung) (3) und „State Defense“ (Zustandsverteidigung) (4). Bei „Element Defense“ könnt ihr ein Häkchen bei den Elementen setzen, gegen die die Rüstung eine Resistenz aufbaut. Trägt ein Charakter eine Rüstung mit einer Elementverteidigung, so fügen Angriffe, welche diesem Element angehören, nur noch halben Schaden zu. Bei „State Defense“ verhält sich dies ähnlich. Hier können Zustände angehakt werden, gegen die der Träger der Rüstung immun wird. Solange der Charakter den Rüstungsteil trägt, ist er gegen alle Zustände immun, die bei „State Defense“ angehakt wurden. Unser Lederwams dagegen erhält keine Zustands- und Elementverteidigung.

Ihr könnt Testweise ja auch noch einen Helm (z.B. eine Fellmütze) und ein Schild (z.B. ein Rundschild) für den Barbaren erstellen. Wenn ihr euch bei den Werten nicht sicher seid, so orientiert euch bei den vorgegebenen Rüstungen.

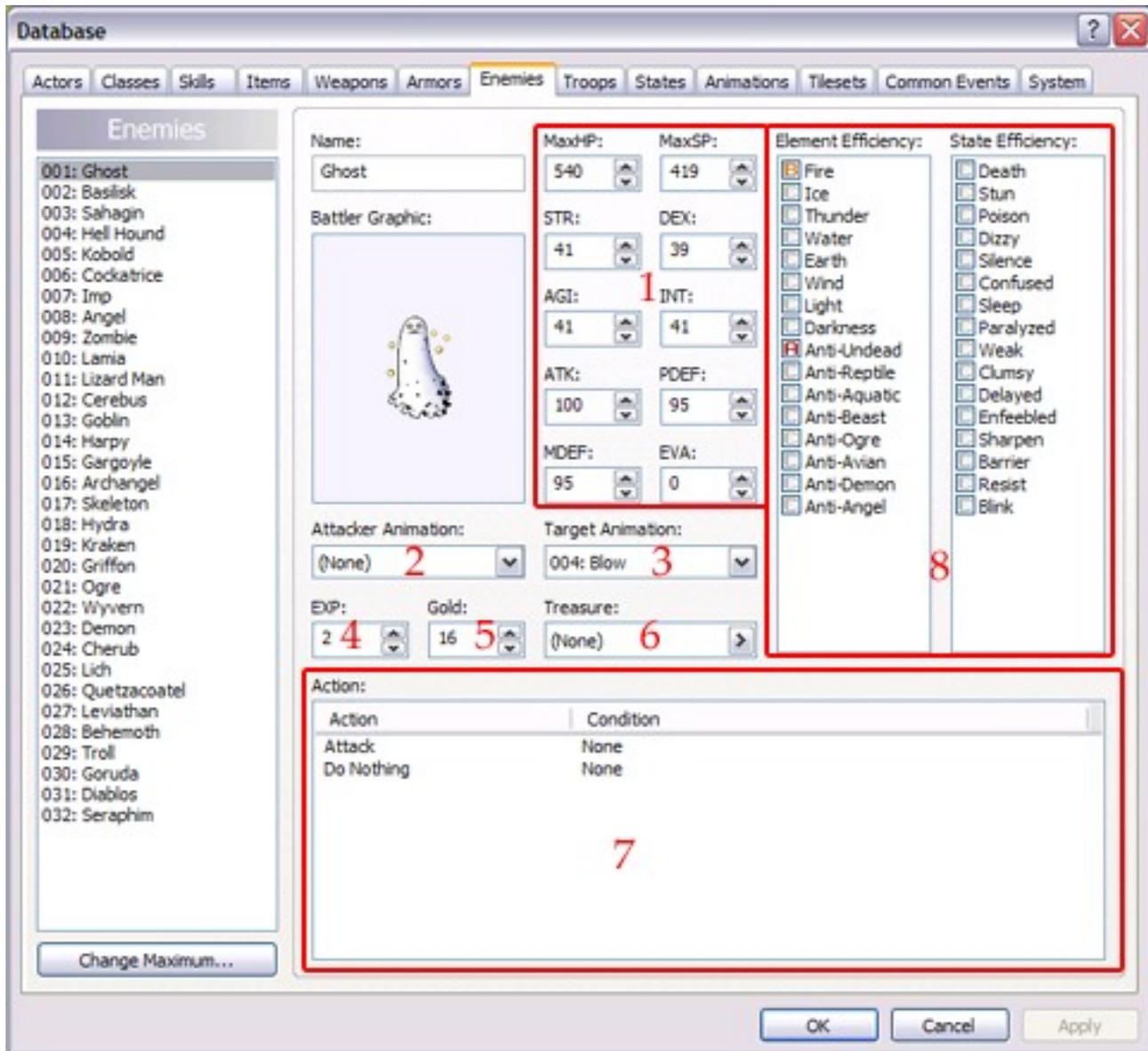
Nun haben wir sowohl eine Waffe, als auch Rüstungen für unseren Barbaren erstellt. Wir gehen nun in den Abschnitt „Classes“ und wählen die Klasse „Barbar“ aus, die wir zuvor erstellt haben. Nun setzen wir bei „Equipable Weapons“ einen Haken bei unserer neu erstellten Streitaxt, und bei „Equipable Armors“ haken wir unsere neuen Rüstungen an. Danach gehen wir in den Abschnitt „Actors“ und wählen unseren Helden „Alex“ an. Bei „Classes“ weisen wir ihm die Klasse „Barbar“ zu. Nun suchen wir das Feld „Starting Equipment“ und rüsten unseren Helden mit der neuen Waffe und den neuen Rüstungen aus.

Slot	Item	Fixed
Weapon:	033: Streitaxt	<input type="checkbox"/>
Shield:	035: Rundschild	<input type="checkbox"/>
Helmet:	034: Fellmütze	<input type="checkbox"/>
Body Armor:	033: Lederwams	<input type="checkbox"/>
Accessory:	(None)	<input type="checkbox"/>

Enemies

Gut ausgerüstet und mit einer starken Angriffstechnik ist unser neuer Held für den Kampf bereit. Nun fehlt nur noch eins: ein würdiger Gegner! Wir gehen also in den Abschnitt „Enemies“ (Feinde), um einen neuen Gegner zu erstellen.

Vor uns baut sich nun folgendes Dialogfenster auf:



Links sehen wir wieder die Liste aller Gegner. Wir erstellen per „Change Maximum“ einen neuen Gegner und geben ihm den Namen „Waldfee“. Unter der Namenseingabe befindet sich wieder ein Grafikenfenster, ähnlich wie wir es beim Abschnitt „Actors“ kennen. Per Doppelklick auf die leere Fläche öffnet sich eine Liste aller Kampfgrafiken. Wählt die Grafik „048-Fairy01“ aus.

Neben der Kampfgrafik sehen wir verschiedene Parameter (1). „MaxHP“ gibt die Lebenspunkte des Gegners an (wir geben unserer Fee 700 HP), während MaxSP die Fertigkeitenpunkte des Gegners festlegt (wir lassen diesen Wert bei 0). Des Weiteren hat der Gegner, ebenso wie die Helden, einen *STR*- (Stärke), *DEX*- (Geschicklichkeit), *AGI*- (Wendigkeit) und *INT*-wert (Intelligenz). Wir stellen alle Werte auf 50.

Da die Gegner keine Waffen und Rüstungen tragen, wird ihr Angriffs- und Verteidigungswert direkt festgelegt. Bei „ATK“ geben wir den Angriffsschaden ein. Unsere Fee bekommt einen Angriffsschaden von 100. „PDEF“ steht für die physische Verteidigung, während „MDEF“ für die magische Verteidigung steht. Unsere Fee erhält eine schwache physische Verteidigung von 50, dafür jedoch eine starke magische Verteidigung von 150.

Bei „EVA:“ kann man dem Gegner eine zusätzliche Ausweichrate geben. Normalerweise wird die Ausweichrate eines Kämpfers aus der Wendigkeit des Verteidigers und der Geschicklichkeit des Angreifers errechnet. Will man jedoch einen Gegner mit besonderer Ausweichrate schaffen, so kann man dies im EVA-Feld einstellen. Unsere Fee erhält jedoch keine besonders hohe Ausweichrate haben. Wir lassen dieses Feld bei 0.

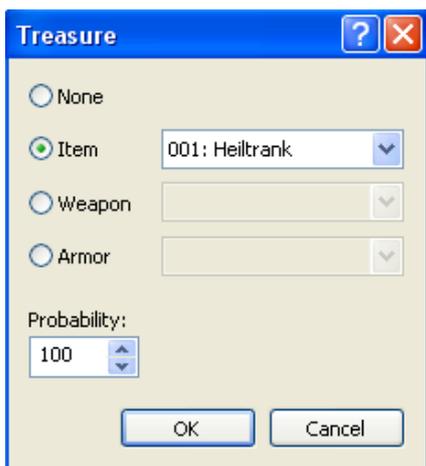
Die nächsten zwei Eingabefelder sollten uns bereits bekannt sein. Bei „Attackers Animation:“ (2) wird die Animation ausgewählt, welche auf den Gegner eingeblendet wird, wenn dieser angreift. Bei „Target Animation“ wird dagegen die Animation ausgewählt, welche auf das Opfer des Angriffes eingeblendet wird. Wir geben unserer Fee als „Target Animation:“ die Animation „004: Hit“. Das Feld „Attackers Animation:“ belassen wir auf „(none)“.

Das nächste Eingabefeld trägt die Bezeichnung „EXP:“ (4). Hier wird die Anzahl der Erfahrungspunkte eingetragen, welche die Heldentruppe beim Besiegen des Gegners erhält.

Gleich daneben steht das Feld „Gold:“ (5). Sie steht für den Geldgewinn der Truppe beim Bekämpfen des Feindes. Unsere Fee soll 3 Erfahrungspunkte und 12 Goldmünzen beim Kampf abwerfen.

Zusätzlich der Erfahrungspunkte und des Geldes kann die Truppe auch noch einen Gegenstand erhalten. Hierfür klickt doppelt auf das Feld mit der Bezeichnung „Treasure:“ (Schatz).

Es öffnet sich nun ein neuer Dialog.



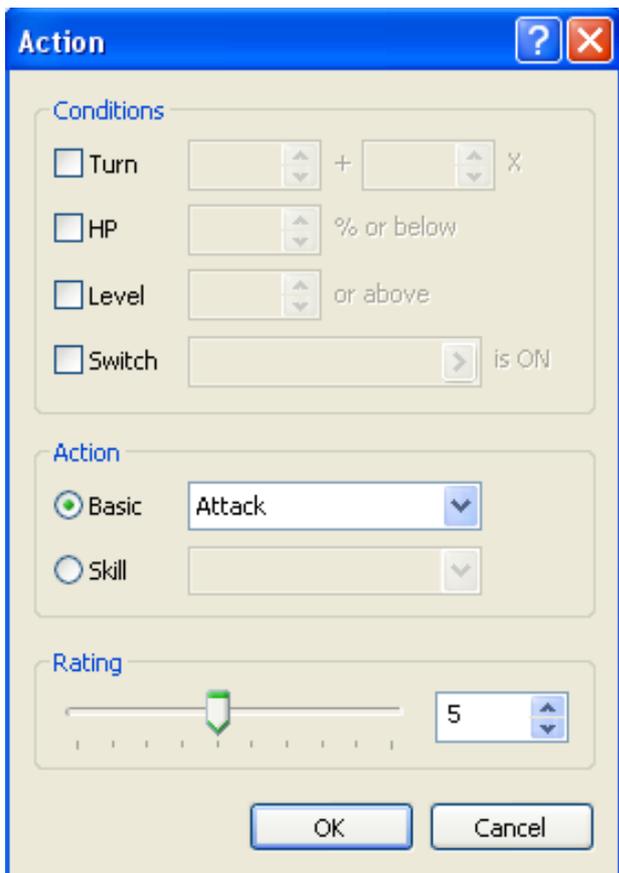
Hier könnt ihr nun einstellen, welchen Gegenstand der Gegner abwerfen soll. Bei „None“ lässt der Gegner keinen Gegenstand fallen. Bei „Item“ könnt ihr ein Item aus der Itemliste auswählen. Ebenso wie ihr bei „Weapon“ eine Waffe und bei „Armor“ eine Rüstung auswählen könnt. „Probability:“

(Wahrscheinlichkeit) gibt die prozentuale Wahrscheinlichkeit an, mit der ein Gegner ein Item besitzt. Wir geben unserer Fee den Heiltrank, den wir zuvor erstellt haben. Als Wahrscheinlichkeit geben wir 50 ein. Statistisch gesehen erhält die Heldentruppe für jede zweite besiegte Fee einen Heiltrank.

Nun haben wir fast alle Felder des „Enemies“-Abschnittes ausgefüllt. Doch zwei wichtige Bereiche fehlen noch. Unten sehen wir eine große Fläche mit der Bezeichnung „Action:“ (7).

Hier wird eingestellt, welche Aktionen der Gegner im Kampf unternehmen soll. Die Fläche ist eine Tabelle mit zwei Spalten: Action und Condition. Unter Action steht der Name der Aktion, unter Condition steht, unter welchen Umständen die Aktion eingesetzt wird.

Standardmäßig befindet sich bereits die Aktion „Attack“ (Angriff) mit der Condition „None“ (also mit keinem besonderem Umstand) in der Tabelle. Wir können diese Aktion aber auch löschen. Hierfür klickt auf die Aktion, und dann auf die Entfernen-Taste. Um eine neue Aktion der Tabelle hinzuzufügen, klickt doppelt auf eine leere Zeile in der Tabelle. Nun öffnet sich ein neues Fenster, in dem man die Aktion näher definieren kann.



Das Fenster ist in drei Abschnitte unterteilt:

Bei Conditions wird der Umstand eingegeben, unter dem die Aktion stattfindet. Bei „Action“ wird erklärt, welche Aktion eingesetzt wird, und bei „Rating“ wird die Wahrscheinlichkeit eingestellt, mit der diese Aktion realisiert wird.

Wenden wir uns zuerst dem Abschnitt „Action“ zu. Hier haben wir die Auswahl zwischen zwei Eingabefeldern: Basic und Skill. Unter Basic können wir grundlegende Aktionen einstellen. Zur Auswahl stehen:

- „Attack“, welches eine normale Angriffsaktion darstellt
- „Defend“, in dem der Gegner in eine Abwehrstellung geht, in der er eine Runde lang nur halben Schaden erhält
- „Escape“, bei welcher der Gegner zu fliehen versucht
- „Do Nothing“, bei der der Gegner eine Runde lang keine Aktion ausführt

Bei „Skill“ dagegen könnt ihr eine konkrete Fertigkeit auswählen, welche der Gegner einsetzen soll.

Wir geben der „Waldfee“ erst einmal eine normale Angriffsattacke. Wir wählen also bei Basic „Attack“. Danach klicken wir auf OK, und erstellen eine neue Aktion. Diesmal möchten wir, dass der Gegner ab und zu in die Abwehrstellung geht. Wir wählen wieder Basic und stellen diesmal „Defend“ ein.

Allerdings soll die Waldfee nicht so oft abwehren, wie sie

angreift. Nun kommt der Abschnitt „Rating“ ins Spiel. Hier wird festgelegt, mit welcher Priorität ein Gegner die jeweilige Aktion einsetzt. Die Aktion „Attack“ hatte eine Priorität von 5. Damit die Aktion „Defend“ nicht so oft eingesetzt wird, wie „Attack“, müssen wir die Priorität verringern. Wenn wir die Priorität um eins verringern, wird die Aktion „Defend“ nur 2/3 so häufig eingesetzt, wie „Attack“. Das heißt, die Fee greift durchschnittlich dreimal an, und verteidigt dann zweimal. Setzen wir die Priorität auf 3, also zwei niedriger, dann wird die Aktion sogar nur 1/3 so häufig eingesetzt wie die Aktion „Attack“. Die Fee würde dann einmal Verteidigen, und daraufhin dreimal angreifen. Würden wir die Priorität noch niedriger setzen, dann täte die Fee nur noch angreifen und gar nicht mehr verteidigen. Wir setzen bei der Aktion „Defend“ die Priorität auf 3. Danach klicken wir auf OK.

Nun wollen wir noch eine dritte Aktion erstellen. Klickt also noch mal doppelt auf eine leere Zeile im „Action“-Bereich. Diesmal wird es eine Flucht-Aktion. Wenn die Lebenspunkte der Fee unter 10% sinken, soll die Fee versuchen zu fliehen. Wir wählen also wieder Basic und „Escape“.

Da die Aktion nur bei einer bestimmten HP-Anzahl eingesetzt werden soll, handelt es sich hier um einen besonderen Umstand. Unser Augenmerk gilt nun also dem „Conditions“-Abschnitt.

Hier stehen uns vier Auswahlfelder zur Verfügung. Bei „Turn“ (Kampfrunde) können wir festlegen, dass eine Aktion nur nach einer bestimmten Kampfrundenfolge erfolgen soll. Damit der Gegner erst unter einer bestimmten Lebenspunktezahl die Aktion ausführen soll, wird das Feld „HP“ benutzt.

Über das Feld „Level“ kann man dem Gegner Aktionen geben, die er nur ausführen darf, wenn die Heldentruppe ein bestimmtes Durchschnittslevel erreicht hat. Bei „Switch“ kann man sogar einen Schalter als Umstand für eine Aktion einbringen.

Die Umstände sind also recht komplex und kompliziert, und wir werden uns daher nicht weiter tiefgründig damit befassen. Wählt einfach das Eingabefeld HP aus, und gebt 10 ein. Damit wird die Aktion Flucht erst eingesetzt, wenn der Gegner 10% oder weniger Lebenspunkte hat. Um diese Fluchtaktion dann wahrscheinlicher zu machen, als den normalen Angriff, können wir zudem noch die Priorität erhöhen. Geben wir der Flucht eine Priorität von 6, so verschiebt sich unser Prioritätsgefüge. Bei sechs Kampfrunden wird die Flucht nun durchschnittlich nur noch dreimal eingesetzt, der Angriff nur noch zweimal, und die Verteidigung nur noch einmal.

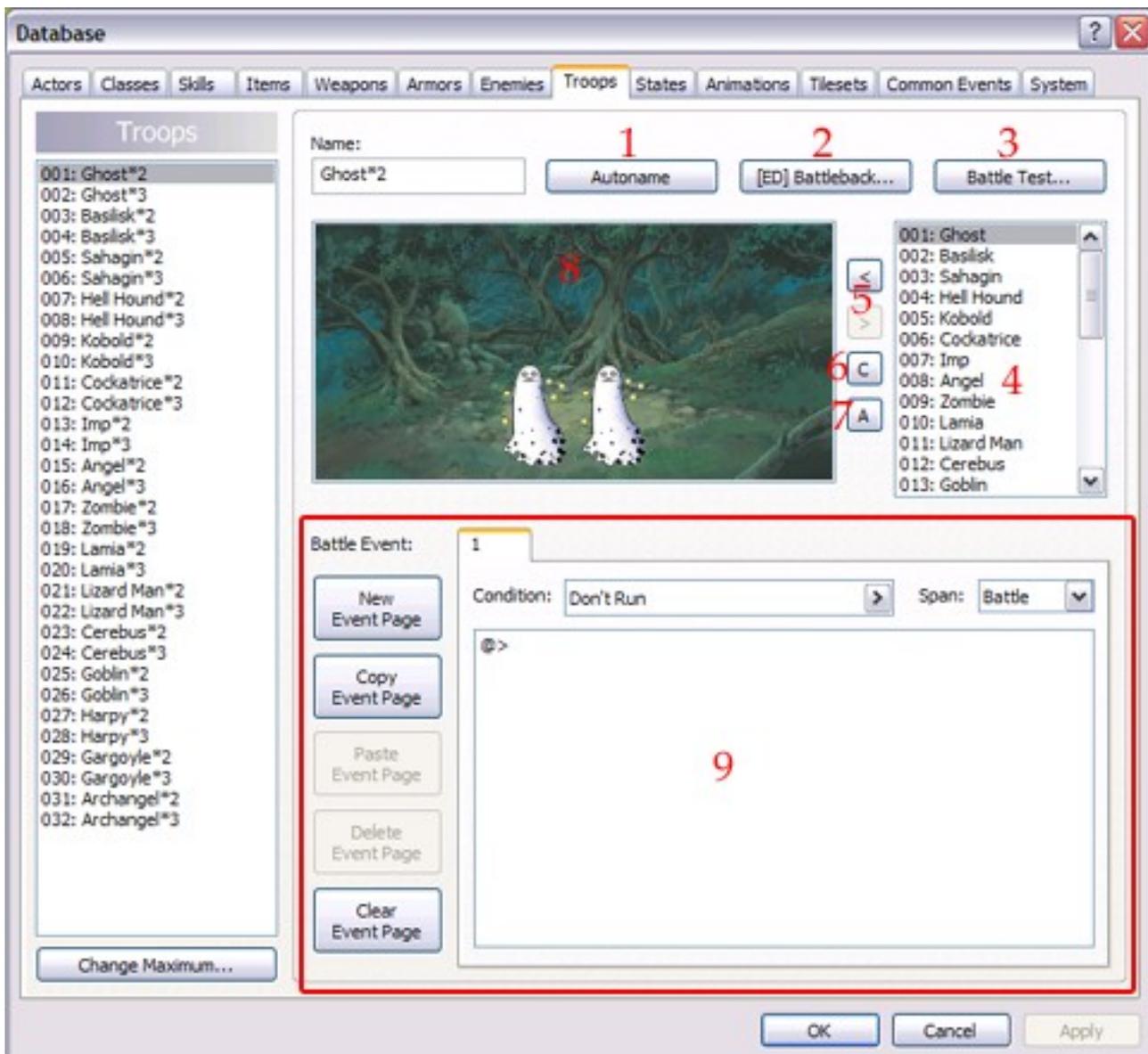
Klickt auf OK. Wir haben dem Gegner nun seine Kampfaktionen zugewiesen. Nun fehlt nur noch ein letzter Bereich, und unser Gegner ist fertig.

Rechts sollten euch sofort wieder die zwei Listen „Element Efficiency:“ und „State Efficiency:“ (8) auffallen. Sie verhalten sich genauso wie in dem Abschnitt „Classes“. Sie legen die Resistenzen und Schwächen des Gegners gegenüber den Elementen und Zuständen fest. Probiert ein bisschen bei den Einstellungen rum, und überlegt, welche Schwächen auf eine Waldfee zutreffen könnten. Ihr könnt diese beiden Listen aber auch einfach auf „C“ lassen, und der Fee so weder Schwächen noch Resistenzen geben.

Troops

Unser Gegner ist nun fertig. Doch damit unser Held gegen ihn antreten kann, muss der Gegner erst in eine Gegnertruppe eingefügt werden, genauso wie sich unser Held in einer Heldentruppe befindet. Wenden wir uns also dem Abschnitt „Troops“ zu.

Es sollte sich nun dieses Dialogfenster öffnen.



Links wieder die Liste aller Gegnertruppen. Fügt per „Change Maximum“ eine neue Gegnertruppe zu den bestehenden hinzu. Wählt diese nun an. Bei „Name:“ könnt ihr dieser Gegnertruppe wieder einen Namen geben. Doch im Gegensatz zu den vorherigen Abschnitten, wird dieser Name nicht im Spiel angezeigt. Er dient lediglich der besseren Übersicht. Aus diesem Grund gibt es auch einen „Autoname“-Button (1), welcher der Gruppe automatisch einen passenden Namen zuweist.

Die zwei Buttons daneben dienen dem „Battle Test“ (Testkampf). Bei „[ED] Battleback“ (2) könnt ihr den Kampfhintergrund für den Testkampf verändern. Dieser Hintergrund ist jedoch nur für den Test, nicht für das eigentliche Spiel. Bei „Battle Test“ (3) startet ihr den Testkampf. Wir werden diesen Testkampf jedoch erst später besprechen. Zuerst wollen wir die Gegnergruppe erstellen.

Hierfür müssen wir zuerst Gegner dieser Gegnertruppe hinzufügen. Rechts in der Liste (4) sind alle Gegner, die zuvor im Abschnitt „Enemies“ erstellt wurden, eingetragen. Sucht nun nach unserer „Waldfee“ und wählt diese an. Um sie der Gegnertruppe hinzuzufügen, klickt auf den Pfeil nach Links (5), gleich neben der Liste. Um sie wieder aus der Gegnertruppe zu entfernen, klickt auf den Pfeil nach rechts, der direkt unter dem ersten Pfeil steht. Um eine Gegnertruppe komplett zu leeren, klickt auf den „C“-Button (6).

In einer Gegnertruppe haben bis zu acht Gegner Platz. Also doppelt so viele, wie in einer Heldentruppe. In unserer Gegnertruppe werden wir jedoch nur drei Waldfeen einfügen. Ihr könnt natürlich auch verschiedene Gegner in eine Gruppe einfügen. Außerdem können diese Gegner auch noch auf dem Kampfbildschirm angeordnet werden. Der Kampfbildschirm ist das große Fenster, welches mit der Nummer 8 markiert wurde. Die eingefügten Gegner erscheinen auch dort auf dem Kampfbildschirm. Wenn ihr einen Gegner anklickt, könnt ihr ihn mit gedrückter Maustaste an eine andere Position verschieben. Des Weiteren könnt ihr einen Gegner mit der rechten Maustaste anklicken, um ein kleines Zusatzmenü zu öffnen.



Zur Auswahl stehen lediglich zwei Punkte. Setzt ihr einen Haken bei „Appear Halfway“, so wird der Gegner im Kampf versteckt und unsichtbar bleiben. Er wird selbst nicht angreifen, kann jedoch auch nicht angegriffen werden. Dieser Zustand kann nur durch den Eventbefehl „Enemy Appearance“ unterbrochen werden. Befinden sich alle Gegner in diesem Zustand, so hat die Heldentruppe den Kampf automatisch gewonnen. Diese Funktion eignet sich zum Beispiel in Kämpfen, in denen während des Kampfes

immer neue Gegner hinzustoßen sollen.

Die zweite Funktion „Immortal“ (unsterblich) macht den ausgewählten Gegner unsterblich. Zwar kann er noch Schaden erleiden (und Aktionen ausführen), er kann jedoch nicht sterben. Dies eignet sich vor allem für Bosskämpfe, in denen der Gegner nicht besiegt werden darf.

Doch wir lassen diese Zusatzfunktionen erst einmal, und wenden uns den anderen Funktionen des „Troops“-Fensters zu.

Ihr könnt die Gegner durch Gedrückthalten der Maus beliebig umversetzen. Wollt ihr die Gegner jedoch wieder in eine Reihe angeordnet haben, so drückt den „A“-Button (7).

Wenn die Gegner richtig aufgestellt sind, können wir einen Testkampf starten.

Zuerst jedoch: Was ist dieser Testkampf eigentlich? Für euer Rollenspiel ist es wichtig, das ihr eine gewisse Balance einhaltet. Eure Charaktere sollten nicht gegen zu starke Gegner kämpfen, doch die Gegner sollten auch nicht zu einfach sein. Die Bosskämpfe müssen stets fordernd aufgebaut werden, doch es darf auch nicht der Fall sein, dass eure Helden vor einem Bosskampf stundenlang trainieren müssen. Um herauszufinden, wie stark eine Heldentruppe sein muss, um gegen eine Gegnertruppe zu bestehen, gibt es den Testkampf. Er läuft ähnlich wie ein normaler Kampf an. Ihr könnt einstellen, aus welchen Helden die Heldentruppe besteht. Außerdem könnt ihr auch das Level sowie die Ausrüstung der Helden festlegen. Des Weiteren besitzt ihr während des Kampfes 99 von jedem Item, welches in der Datenbank erstellt wurde. Damit könnt ihr also nicht nur die Stärke eurer Helden und Gegner testen, sondern auch die Wirksamkeit eurer Items. Wir haben nun einen Helden mit einer Heldenklasse und mit Skills erstellt, wir haben Waffen, Rüstungen und Items, sowie ein Monster und die dazugehörige Monstergruppe entworfen. Nun wollen wir das Ergebnis dieser Arbeit in einem Testkampf sehen.

Klickt auf den Button „Battle Test“ (3). Es öffnet sich nun ein neues Fenster.



Das Fenster lässt sich in vier Abschnitte, für jeden Helden in der Heldentruppe einer, einteilen. Diese Abschnitte sind über die vier Buttons (1) mit den Zahlen 1 bis 4 erreichbar. In dem Feld „Actor:“ (2) wählt ihr aus, welche Helden in die Heldentruppe des Testkampfes kommen sollen. Ihr könnt in jedem Abschnitt einen Helden einstellen, so dass ihr am Ende eine Heldentruppe mit vier Helden habt. Wenn ihr mit weniger Helden kämpfen wollt, müsst ihr bei Actor „None“ eingeben. Wir wollen jetzt nur unseren Barbarencharakter testen. Deshalb stellt beim ersten Abschnitt als Helden den Held Alex ein, und bei allen anderen Abschnitten „none“. Bei dem Feld „Level:“ (3) wird das Level des jeweiligen Helden eingegeben. In den Feldern bei „Equipment“ (4) könnt ihr die Ausrüstung des Helden festlegen. Gebt unserem Barbarencharakter gleich unsere Ausrüstungsgegenstände: Die Streitaxt und den Lederwams.

In der rechten Spalte (5) seht ihr die Attribute des Helden im jeweiligen Level mit der jeweiligen Ausrüstung. Durch den Button „Initialize“ (Voreinstellung) könnt ihr wieder die im Abschnitt „Actors“ festgelegten Startbedingungen einstellen.

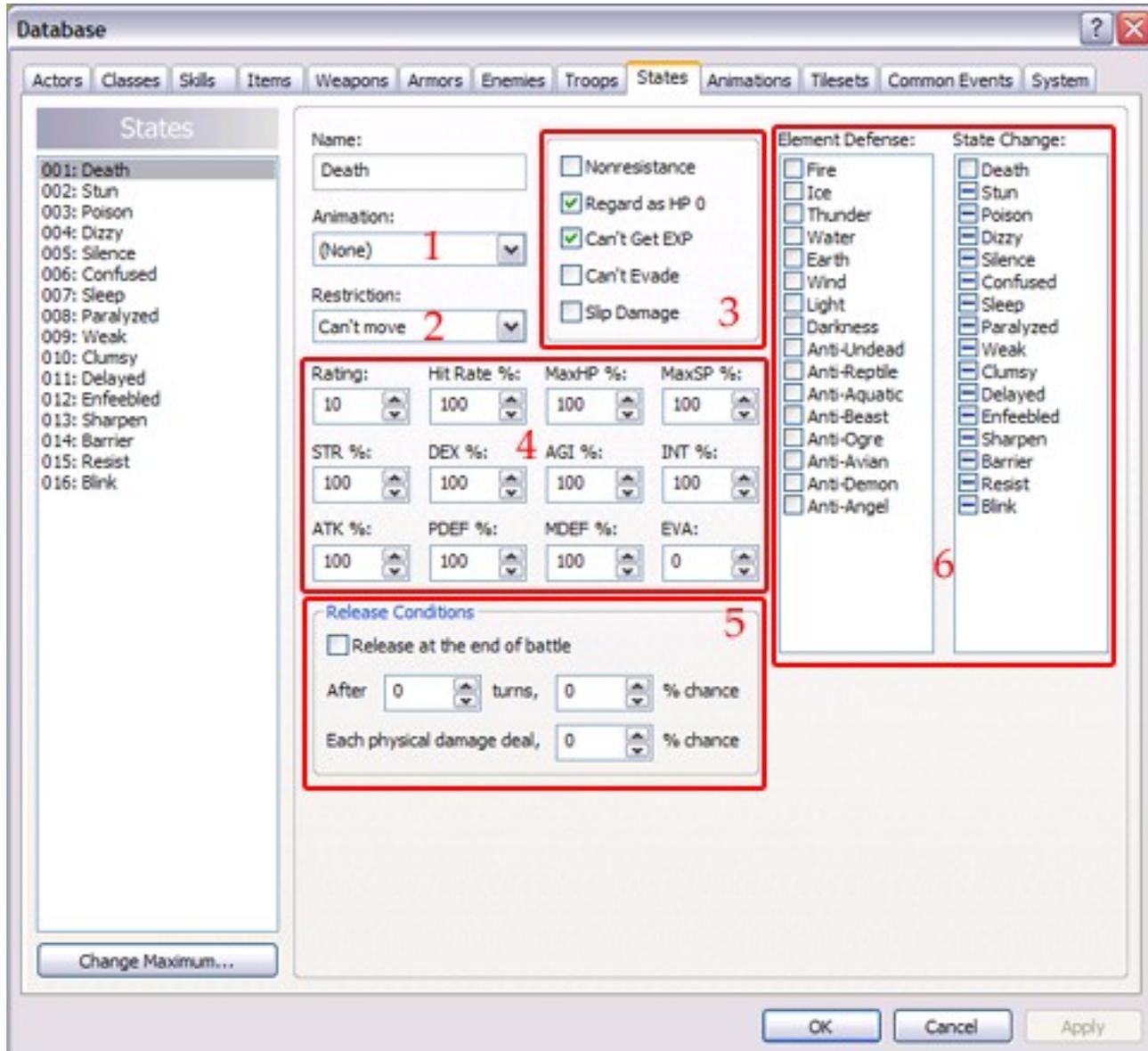
Sind alle Einstellungen vorgenommen, so klickt auf OK. Nun öffnet sich das Kampfsystem, und ihr könnt die Stärke des Helden, und seine Fertigkeiten austesten.

Den letzten Teil des „Troops“-Abschnitts bilden die „Battle-Events“ (9). Dies sind Ereignisse, welche im Kampf auftreten. Wir haben ja bereits häufiger über Events bzw. Ereignisse geredet. Es gibt Map-Events, also Ereignisse auf der Karte, es gibt Common-Events, also globale, das Spiel umfassende Ereignisse, und es gibt auch Battle-Events. Ereignisse, welche nur im Kampf auftauchen. Sie funktionieren genauso wie die Map-Events. Auch sie haben Event-Commands. Auch sie besitzen gewisse Conditions (Bedingungen), damit sie eintreten. Diese Battle-Events ermöglichen es zum Beispiel, Dialoge in den Kampf einzubauen, oder komplexe Ereignisse zu schreiben. Doch der Umgang mit Ereignissen ist sehr kompliziert. Daher werden wir in diesem Teil des Tutorials nicht näher darauf eingehen.

States

Als nächstes beschäftigen wir uns mit dem Abschnitt „States“ (Zustände). Wir haben bereits häufiger mit Zuständen gearbeitet, nun werdet ihr lernen, wie man neue Zustände erstellt.

Durch einen Klick auf „States“ gelangt ihr in folgendes Fenster:



Links befindet sich wieder die Liste mit allen Zuständen. Über „Change Maximum“ erstellt ihr nun einen neuen Zustand. Wir geben ihm den Namen „Schlaf“. Bei Animation (1) könnt ihr eine Animation einstellen, die auf das Opfer des Zustandes ständig angezeigt werden soll. Doch wisst, das solche Animationen auch sehr nervig sein können. Wir lassen daher die Animation weg und belassen das Feld auf „none“.

Bei „Restriction:“ (2) werden die Einschränkungen, welche der Zustand mit sich bringt, eingestellt. Zur Verfügung stehen die Einschränkungen:

- „none“, also keine Einschränkungen
- „can't use magic“ (kann keine Magie benutzen); in diesem Fall ist der Charakter, der von diesem Zustand betroffen ist, nicht mehr in der Lage Fertigkeiten zu benutzen, welche nicht vom Angriffsschaden seiner Waffe abhängig sind (also alle Skills, die bei „ATK-F“ eine 0 stehen haben). Dies eignet sich zum Beispiel für Fertigkeiten wie „Stummheit“.

- „Always attack enemies“ (greife immer Gegner an) bewirkt das der Betroffene unkontrolliert jede Runde einen zufälligen Gegner angreift. Er setzt keine Zauber mehr ein, sondern greift stets mit seiner normalen Attacke an (lustigerweise selbst dann, wenn ein normaler Angriff nicht zur Verfügung steht, wie es bei Gegnern vorkommen kann). Dies wäre zum Beispiel für einen „Berserker“-Zustand nützlich
- „Always attack friends“ (greife immer Freunde an) ist dasselbe wie „Always attack enemies“, nur das der Betroffene Mitglieder seiner eigenen Truppe, statt seine Gegner angreift. Anwendbar zum Beispiel bei einem „Verwirrt“-Zustand
- „can't move“ (kann sich nicht bewegen) bedeutet, dass der Betroffene keine Aktion mehr ausführen kann, solange er von dem Zustand befallen ist. Dies wäre bei Zuständen wie „Schlaf“, „Gelähmt“, „Versteinert“ und „Tod“ der Fall.

Für unseren „Schlaf“-Zustand darf der Held während des Kampfes keine Aktion ausführen. Deshalb stellen wir bei „Restriction“ die Auswahl „can't move“ ein.

Als nächstes wenden wir uns dem Feld daneben zu, welches mit der Nummer 3 gekennzeichnet ist. Hier stehen mehrere Optionen, welche angehakt werden können. Diese Optionen verleihen dem jeweiligen Zustand besondere Eigenschaften.

„Nonresistance“ (keine Resistenz) bedeutet, dass dieser Zustand nicht durch eine Resistenz abgewehrt werden kann. Jede Klasse und jeder Gegner verfügt über bestimmte Resistenzen oder Schwächen gegenüber den Zuständen. Hat ein Zustand diese Option angehakt, so gilt die Resistenz der Charaktere und Gegner gegenüber diesem Zustand nicht mehr. Diese Option wird normalerweise bei positiven Zuständen verwendet. Wenn man einen Charakter mit dem Zustand „Magiebarriere“ belegt, wäre es natürlich ungünstig, wenn dieser Charakter dem Zustand ausweichen würde. Durch anhängen dieser Option, wird dies verhindert.

„Regard as HP 0“ lässt einen Zustand automatisch dann eintreten, wenn der Charakter stirbt. Dies ist normalerweise nur beim Todeszustand der Fall. Doch man kann natürlich auch eigene Zustände wie „Zombie“ erstellen.

„Can't get EXP“ verhindert, das ein Charakter, welcher von diesem Zustand betroffen ist, Erfahrungspunkte am Ende des Kampfes erhält. Auch diese Option trifft normalerweise nur auf den Todeszustand zu.

Verfügt ein Zustand über die Option „Can't Evade“, so ist der Betroffene nicht mehr in der Lage Angriffen auszuweichen. Dies eignet sich natürlich für Zustände wie „Schlaf“ oder „Gelähmt“.

Die letzte Option trägt die Bezeichnung „Slip Damage“. Besitzt ein Zustand diese Option, so erleidet der betroffene Kämpfer jede Kampfrunde Schaden in Höhe von 10% seiner maximalen Lebenspunkte. Wirkt dieser Zustand auch auf dem Felde, so erleidet der Held alle zwei Schritte Schaden in Höhe von 1% seiner maximalen Lebenspunkte. Dies ist natürlich für Zustände wie „Vergiftet“ ideal.

Für unseren „Schlaf“-Zustand ist jedoch jediglich die Option „Can't Evade“ interessant. Setzt hier einen Haken. Alle anderen Optionen lasst ausgeschaltet.

Als nächstes wenden wir uns den vielen Parametern zu, welche durch die Nummer 4 gekennzeichnet sind. „Rating“(Rangfolge) ist dafür entscheidend, welcher Zustand im Kampffenster angezeigt werden soll. Bei jedem Statusfenster ist nur Platz für einen Zustand. Zwar kann ein Charakter mit beliebig vielen Zuständen befallen sein, doch es kann nur ein Zustand gleichzeitig angezeigt werden. Der Zustand mit dem höchsten Rating, wird angezeigt. Man sollte also Zustände nach Wichtigkeit ordnen. Die wichtigsten Zustände sollten eine hohe Zahl bei „Rating“ erhalten. Weniger wichtige dagegen eine niedrige Zahl. Unser Schlaf-Zustand erhält den Wert 5.

Die weiteren Parameter sind Attribute des Kämpfers. Die Parameter geben an, wie stark diese Attribute gesenkt oder gesteigert werden sollen. Steht ein Parameter bei 100, so bleibt das Attribut wie es ist. Ist ein Parameter über 100, so wird dieser Zustand um diese Prozentzahl gesteigert. Ist der Parameter unter 100, so wird das Attribut gesenkt.

Da man, während man schläft, keine gute Verteidigung besitzt, senken wir das Attribut „PDEF“ auf 50. Damit hat der vom Schlaf-Zustand Betroffene nur noch die Hälfte seiner physischen Verteidigung. Die anderen Attribute lassen wir so, wie sie sind.

Der nächste Bereich trägt die Bezeichnung „Release Conditions“ (5).

Hier wird festgelegt, auf welche Weise der Zustand wieder verschwindet. Dafür gibt es eigentlich nur drei Möglichkeiten:

Wird ein Haken auf „Release at the end of Battle“ (Auflösung am Ende des Kampfes) gesetzt, so verschwindet der Zustand nach Ende des Kampfes.

Des Weiteren kann eingestellt werden, dass der Zustand nach einer bestimmten Rundenzahl vergeht. Diese Rundenzahl wird im Eingabefenster links neben „Turns“ (Kampfrunden) eingegeben. Die prozentuale Wahrscheinlichkeit, mit der dieser Zustand nach der eingegebenen Zahl an Kampfrunden vergeht, wird rechts neben „Turns“ eingegeben.

Damit unser Schlafzustand jede Runde zu einer 20%igen Wahrscheinlichkeit verschwindet, geben wir ein: „After 1 Turns 20 % Chance“.

Darunter befindet sich noch der Schriftzug „Each physical damage deal“, und ein weiteres Eingabefenster. Hiermit kann man einen Zustand auch vergehen lassen, wenn der Held physischen Schaden erhält. Da ein Schlafender sicherlich aufgeweckt wird, wenn er von einem Monster angegriffen wird, trifft diese Option sicherlich für unseren Schlaf-Zustand zu.

In dem Eingabefeld wird die prozentuale Wahrscheinlichkeit festgelegt, mit welcher der Zustand nach einem physischen Angriff verschwindet. Wir geben hier die Zahl 50 ein. Damit besteht nach jedem Angriff auf einen schlafenden Kämpfer eine Wahrscheinlichkeit von 50%, dass dieser aufwacht.

Lässt man all diese Eingabefelder auf 0, so kann dieser Zustand nur noch von bestimmten Items oder Zaubern geheilt werden.

Rechts befinden sich wieder die zwei Spalten (6) mit Elementen und Zuständen. „Element Defense:“ verhält sich genauso wie bei den Rüstungen. Hakt man ein Element an, so erhält der vom Zustand Betroffene nur noch die Hälfte an Schaden, vom angehakten Element.

Mit „State Change:“ kann ein Zustand den Betroffenen von anderen Zuständen heilen, oder ihm andere Zustände einflößen. So könnte zum Beispiel der Zustand Schlaf den Betroffenen vom Zustand „Weaken“ (Schwach) heilen. Denn wer schläft, ist danach ausgeruht und nicht mehr schwach. Setzt also ein Minus bei Weak.

Nun haben wir einen neuen Zustand geschaffen. Ihr könnt noch ein wenig mit den bisher besprochenen Abschnitten der Datenbank herumprobieren. Erstellt doch neue Fertigkeiten, welche einen Gegner einschläfern, und testet diese Fertigkeiten durch den Testkampf.

Die nächsten drei Abschnitte sind zu komplex, um in diesem Tutorial Platz zu finden. Sie werden im nächsten Teil des Kurses erläutert.

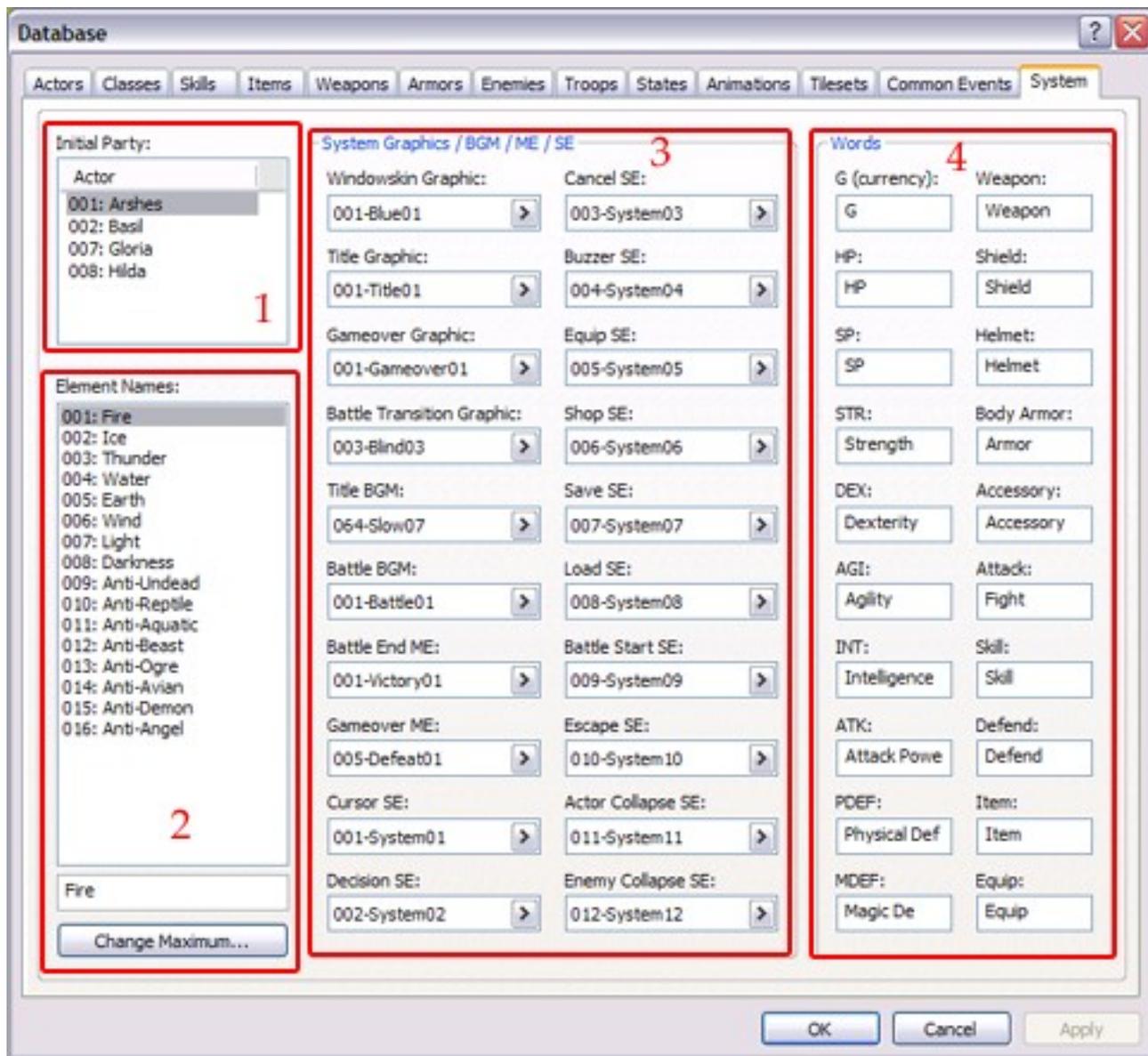
Denn das Erstellen von Animationen ist sehr mühsam und verlangt viel Geschick. Auch beim Erstellen von Tilesets bedarf es vieler Einstellungen. Glücklicherweise sind bereits 100 Animationen und 50 Tilesets im Projekt erhalten.

Wie man neue Ressourcen in das Projekt einfügt, wird unter anderem Thema des nächsten Kurses sein.

Ebenso verhält es sich bei den CommonEvents. Zwar funktionieren sie nahezu identisch wie die Events auf den Maps, dennoch ist ihre Anwendung zu komplex und kompliziert, als dass sie in diesem Tutorial Platz fänden.

System

Doch auf den letzten Abschnitt der Datenbank wollen wir noch eingehen. Es ist der Abschnitt „System“. Hier werden grundlegende Einstellungen zum Projekt getätigt. Betrachtet erst einmal dieses Dialogfenster mit seinen unzähligen Einstellungen.



Oben links befindet sich ein Fenster mit der Bezeichnung „Initial Party“ (1). Dort wird eingestellt, mit welcher Heldentruppe der Spieler das Spiel beginnt. Es können nur vier Helden in die Heldentruppe. Um einen Helden zu entfernen, wählt ihn an, und klickt die Entfernen-Taste. Ihr könnt auch ohne Helden das Spiel beginnen. Dies macht sogar manchmal Sinn, wenn ihr zuerst eine Introsequenz einspielen wollt.

Um einen neuen Helden hinzuzufügen, klickt doppelt auf eine leere Zeile im Fenster. Nun erscheint eine Auswahl aller Helden/Akteure. Ihr könnt nun vier andere Helden für eure Starttruppe festlegen. Natürlich kann die Starttruppe auch weniger Helden enthalten.

Das Fenster darunter legt die Elemente fest, welche im Spiel vorkommen. Über „Chance Maximum“ kann man neue Elemente hinzufügen. Über die Entfernen-Taste auf der Tastatur lassen sich bestehende Elemente löschen. Die Namenseingabeleiste für die Elemente steht über dem „Change Maximum“-Button.

Die vielen Optionen in der Mitte (3) sind einige wichtige Einstellungen für euer Projekt.

Bei „Windowskin Graphic:“ wird die Grafik der Fenster für das Kampfsystem, das Menü und die Textboxen ausgewählt. Durch Doppelklick auf „Title Graphic:“ könnt das Titelbild, welches beim Starten des Spiels angezeigt wird, ändern. Dasselbe gilt für „GameOver Graphic:“, welches jedoch für das Bild gilt, welches beim Verlieren des Spiels angezeigt wird.

„Battle Transition Graphic:“ ist für den Einblenden-Effekt verantwortlich, welcher bei einem Kampfbeginn erscheint.

Danach wird die Musik für das Menü, bei „Title BGM:“, und für den Kampf, „Battle BGM:“, festgelegt. Bei „Battle End ME:“ wird der Musikeffekt ausgewählt, welcher am Ende eines Kampfes abgespielt wird. „GameOver ME:“ ist der Musikeffekt, welcher während der Gameover-Sequenz gespielt wird. Unter „Cursor SE“ könnt ihr den Soundeffekt ändern, welcher beim bewegen des Cursors zum Beispiel im Kampfsystem oder im Menü erklingt. Dasselbe gilt für „Decision SE“, welcher beim auswählen einer Option mit dem Cursor ertönt, „Cance SE“, welcher beim Abbrechen eines Menüs gespielt wird und „Buzzer SE“, welcher dagegen beim auswählen einer nicht auswählbaren/deaktivierten Option mit dem Cursor abgespielt wird.

Auch die weiteren Auswahlfelder sind für Soundeffekte verschiedener Aktionen, die wir jetzt allerdings nicht näher erläutern brauchen.

Rechts befindet sich ein letzter Bereich mit der Bezeichnung „Words“ (4). Hier befinden sich die verschiedenen englischen Begriffe, welche im Spiel vorkommen. Damit in eurem Projekt aber deutsche Begriffe verwendet werden, könnt ihr hier diese Begriffe übersetzen. Des Weiteren wird hier auch der Name der Währung, sowie die Bezeichnung der Attribute festgelegt.

Leider kann über diese Vokabular nicht jeder Begriff in eurem Projekt übersetzt werden. Manche Übersetzungen muss man über die RGSS erledigen. Wie das funktioniert, soll allerdings nicht Bestandteil dieses Tutorials sein.

Schlusswort

Nach diesem kurzen Einstieg solltet ihr bereits in der Lage sein, ein kleines Rollenspiel zu erschaffen. Ihr wisst nun wie Helden erstellt werden und wie diese auf der Karte herumlaufen und gegen Feinde kämpfen. Außerdem könnt ihr bereits NPCs erstellen, die mit den Helden reden.

Im nächsten Tutorial werden wir auf diese NPCs und auf andere Ereignisse näher eingehen, so dass ihr komplexere Ereignisse erstellen könnt.

Damit verabschieden wir uns bis zur nächsten Ausgabe der Screenfun!

Sollten noch Fragen bezüglich des RPG Maker Kurses bestehen, könnt ihr euch an das Technikforum auf www.rpga.info wenden. Wenn ihr Vorschläge und Feedback zum Kurs geben wollt freuen wir uns auf eure Mail die ihr bitte an rmxpkurs@rpga.info schickt!