

Sudden Strike Resource War
&
Sudden Strike II

Map and Mission Editor
User Guide V1.0

von Gareth

Vorbemerkung

Als begeisterter Spieler von Sudden Strike¹ war ich sehr angetan, als das Add-On Sudden Strike Forever mit einem Editor für eigene Missionen geliefert wurde. Leider bin ich nie so recht dazu gekommen, den Editor zu benutzen – entweder fehlte die Zeit oder die Lust, manchmal auch beides. Dann kam Sudden Strike II ebenfalls mit Editor und wieder wurde nichts aus der geplanten Arbeit an eigenen Missionen. Nun ist Sudden Strike Resource War² schon längst nicht mehr aktuell (Sudden Strike III steht vor der Tür), aber ich habe es endlich geschafft, mich mit dem Editor zu befassen. Zwar spät, aber besser spät als nie...

Die erste Quelle für Informationen war natürlich das mit Sudden Strike II mitgelieferte Handbuch zum Map- und Mission-Editor³. Der Umgang mit dem Karteneditor ist dort recht ordentlich beschrieben, so dass das Erstellen von einigermaßen ansehnlichen Karten kein großes Problem darstellt. Da eine Karte ohne darauf ablaufende Mission aber ungefähr so sinnvoll ist wie eine Multiplayer-Schlacht ohne Mitspieler, stand anschließend die Beschäftigung mit den Skriptbefehlen auf dem Programm. Zu der Beschreibung der Skriptgrundlagen in dem erwähnten Editorhandbuch ist nicht viel zu sagen. Der oder die Verfasser behaupten zwar, dass die Befehle und Optionen selbsterklärend sind, aber das ist wohl nur dann der Fall, wenn man so tief in der Materie drinsteckt, dass man eigentlich gar keine Erklärungen mehr braucht. Für den Anfänger bleiben jedoch auch nach intensivem Lesen so viele Fragen offen, dass man um eigene zeitraubende und oft frustrierende Versuche nicht herumkommt.

Da ich nicht immer in der Lage bin, permanent an den eigenen Missionen zu arbeiten, sondern immer wieder viel Zeit zwischen zwei Editorsitzungen vergeht, habe ich daher angefangen, mir Notizen zu machen, damit ich nicht immer wieder die gleichen Probleme lösen muss. Irgendwann kam mir bei der Suche nach Informationen im WWW der Gedanke, meine noch recht spärlichen Erfahrungen zusammenzufassen und der SuSt-Community zur Verfügung zu stellen, da offenbar viele die gleichen Probleme hatten und haben wie ich. Et voilà – hier ist also der ultimative SSRW-Map- und Mission Editor User Guide. Bevor ich jetzt angesichts dieser fürchterlich eingebildeten Bemerkung gesteinigt werde – wenn auch nur virtuell – muss ich einige Einschränkungen machen, denn so ultimativ ist dieser Text dann auch wieder nicht.

In der Originaldokumentation wird z.B. auch die Erzeugung von Mehrspielermissionen beschrieben. Da mein Interesse sich aber auf die Entwicklung von Einzelspielermissionen konzentriert, habe ich Mehrspielermissionen in diesem User Guide nicht behandelt. Bei Interesse sei auf das genannte Originalhandbuch oder – viel besser – auf die sehr hilfreiche Editor-Fibel von Irrläufer⁴ verwiesen. In diesem Zusammenhang möchte ich auch das Kampagnentutorial von Slyder und Lindberghs Handbuch zum Briefing-Editor⁵ empfehlen, die die Erstellung von Kampagnen bzw. Briefing-Animationen sehr anschaulich darstellen.

Einiges von dem, was in diesem User Guide zu finden ist, habe ich mir mangels entsprechender Dokumentation irgendwie zusammengereimt. Dabei mögen mir Fehler oder Ungenauigkeiten unterlaufen sein. Außerdem gibt es immer noch einige Lücken und offene Fragen. Fehlerfreiheit (übrigens auch im Hinblick auf Rechtschreibung und Zeichensetzung) und Vollständigkeit hatten bei der Arbeit an diesem User Guide jedoch nicht die oberste Priorität. Mein Ziel war es ursprünglich nur, die Original-

¹ kurz SuSt

² kurz SSRW

³ Das Handbuch findet man unter C:\Programme\Sudden Strike II\editor\Editorhandbuch.htm. Von diesem Handbuch existiert auch eine englische, an einigen Stellen etwas ausführlichere Version (Download von <http://www.suddenstrike.com>, unter Downloads > SSRW). In dem zip-Archiv findet man zusätzlich eine kurze Beschreibung des Briefing-Editors von SuStII und einen Text, der die Änderungen behandelt, die sich in SSRW gegenüber SuStII ergeben haben.

⁴ Download als jpg-Bilder von <http://www.strategicgames.de/maps/include.php?path=start.php> (unter Forum > Editor > Editor-Fibel). Es existiert auch eine pdf-Version, die man z.B. auf <http://www.hs-noobs.org> (unter Downloads) findet.

⁵ Beide Tutorials kann man von <http://www.sudden-strike-maps.de/> herunterladen.

dokumentation in für mich brauchbarer Form aufzubereiten. Ich werde aber natürlich bei entsprechender Rückmeldung in Erwägung ziehen, den User Guide zu überarbeiten

Die hier beschriebenen Vorgehensweisen sind naturgemäß subjektiv geprägt. Es liegt mir fern, diese als die einzige oder beste Möglichkeit darzustellen. Es handelt sich dabei ausdrücklich um *meine* Arbeitsweise. Wer anders vorgeht, soll das natürlich auch weiterhin tun. Besonders die Profis, die mir an Erfahrung weit voraus sind, werden dazu sicher ihre eigene Meinung haben. In erster Linie richtet sich dieser User Guide allerdings auch an Neulinge, die ihre ersten Missionen erstellen wollen und an der Originaldokumentation verzweifeln.

Da sich von SuStII zu SSRW am Editor nicht so furchtbar viel verändert hat, sollte dieser User Guide auch für die Arbeit mit SuStII ganz brauchbar sein. Ich habe versucht, die Unterschiede so weit wie möglich anzugeben. Wie weit die Darstellung mit irgendwelchen Mods kollidiert, kann ich nicht sagen, da ich bisher nur für die nicht modifizierte Version entwickle.

Der vorliegende User Guide wird so zur Verfügung gestellt, wie er ist. Für Fehler usw. wird keine Gewährleistung oder Haftung übernommen. Die Verwendung des User Guides und der darin enthaltenen Informationen erfolgt daher auf eigenes Risiko und eigene Verantwortung.

Mein Dank geht an alle, die im Forum von www.sudden-strike-maps.de Fragen beantwortet haben und immer noch beantworten, an meine Betatester und Testleser. Einige möchte ich (in keiner besonderen Reihenfolge) besonders erwähnen:

- Irrläufer für seine Editor-Fibel, die den Anstoß für diesen User Guide gegeben und mich gleichzeitig vor der Versuchung bewahrt hat, auch noch Mehrspielermissionen zu behandeln,
- Lindbergh für die Anleitung zum Briefing-Editor,
- Maddin dafür, dass er seine Alpenfestung Ost als Editordatei zur Verfügung gestellt hat, die ich als Anschauungsmaterial vor allem für die Realisierung von Teilzielen (hübsche Storyline übrigens) und von humoristischen Einsprengseln (Werner-Fan, wie?) ausdrücklich empfehle,
- Slyder für sein Kampagnentutorial und für die Hilfe bei der Textgestaltung für das Briefing,
- www.lexikon-der-wehrmacht.de für die Informationen über Waffen und Gliederung von Einheiten,
- Astro, Ciceri, OHK und Pet Hauptmann für ihre Rückmeldungen zu der Vorabversion des User Guides
- und natürlich den Betreibern von www.sudden-strike-maps.de.

Nach diesen Vorbemerkungen wünsche ich allen Lesern viel Erfolg bei der Entwicklung von Missionen für SSRW und SuStII. Die Profi-Mapper und -Skriptor mögen mir verzeihen, wenn ich allzu Bekanntes ausbreite.



Dortmund, im Dezember 2006

Inhaltsverzeichnis

Vorbemerkung	2
Inhaltsverzeichnis	4
1 Überblick	8
2 Die Benutzeroberfläche des Editors	10
3 Karten	13
3.1 Dateioperationen mit Karten.....	13
3.1.1 Anlegen einer neuen Karte.....	13
3.1.2 Speichern und Laden von Karten.....	13
3.1.3 Erzeugen einer Bitmap der Karte.....	14
3.2 Terrain.....	14
3.2.1 Terrain editieren.....	14
3.2.2 Helligkeit des Terrains.....	15
3.2.3 Weitere Ausgestaltung des Terrains.....	16
3.3 Statische Objekte.....	16
3.3.1 Plazieren von Objekten.....	16
3.3.2 Löschen von Objekten.....	18
3.4 Automatische Generierung von Klippen, Straßen und Bahnlinien.....	18
3.4.1 Vorgehensweise.....	18
3.4.2 Erzeugung von Klippen.....	19
3.4.3 Erzeugung von Bahnlinien.....	19
3.5 Flugplätze.....	19
3.6 Sichtbarkeit und andere Flags.....	21
3.6.1 Sichtbarkeit.....	22
3.6.2 Blockierte Felder.....	22
3.6.3 Wellendarstellung unterdrücken.....	23
3.6.4 Seichte Wasserflächen.....	23
3.6.5 Bau von Pontonbrücken verhindern.....	23
3.6.6 Undurchdringliche Felder.....	24
3.7 Wiederverwendung von Kartenausschnitten.....	24
3.7.1 Speichern von Objektblöcken.....	24
3.7.2 Laden von Objektblöcken.....	25
3.7.3 Beispiele für die Verwendung von Objektblöcken.....	25
3.8 Kartengestaltung.....	25
3.8.1 Texturübergänge.....	26
3.8.2 Blockieren von Objekten.....	26
3.8.3 Darstellung von Feldern.....	26
4 Missionen	28
4.1 Dateioperationen mit Missionen.....	28
4.1.1 Anlegen einer neuen Mission.....	28

4.1.2 Speichern und Laden von Missionen.....	29
4.1.3 Zurücksetzen einer Mission.....	29
4.1.4 Erzeugen einer Bitmap der Mission.....	29
4.2 Grundlegende Missionseinstellungen.....	29
4.2.1 Spielereinstellungen.....	29
4.2.2 Missionsziele.....	30
4.2.3 Taktische Karte.....	31
4.2.4 Missionseinstellungen.....	31
4.3 Der Einheiteneditor.....	32
4.4 Der Skripteditor.....	34
4.4.1 Skriptverwaltung.....	34
4.4.2 Skriptprogrammierung.....	34
4.5 Testen und Fertigstellen einer Einzelmission.....	36
4.6 Erzeugung von Kampagnen.....	36
4.6.1 Skripte.....	36
4.6.2 Erzeugen einer Kampagnenmission.....	36
4.6.3 Erzeugen der Kampagnendatei.....	36
5 Skriptgrundlagen.....	39
5.1 Einheiten.....	39
5.1.1 Hinzufügen neuer Einheiten.....	39
5.1.2 Bearbeiten vorhandener Einheiten.....	40
5.1.3 Verhalten von Einheiten.....	40
5.1.4 Einstellung der Wiederbelebungsoptionen.....	41
5.1.5 Löschen von Einheiten.....	42
5.1.6 Arbeit mit Gruppen.....	42
5.2 Verstärkungen.....	42
5.2.1 Verstärkungen bearbeiten.....	42
5.2.2 Tore.....	44
5.3 Orte.....	44
5.3.1 Locations.....	44
5.3.2 Marker.....	45
5.3.3 Minenfelder.....	45
5.3.4 Ortsabhängige Sounds.....	46
5.4 Texte.....	46
6 Verhalten von Einheiten.....	48
6.1 Nicht an Einheitentypen gebundene Verhaltensmodelle.....	48
6.1.1 Das Verhaltensmodell None.....	48
6.1.2 Das Verhaltensmodell Reconnaissance.....	49
6.2 Verhaltensmodelle für Infanterie-Einheiten.....	49
6.2.1 Das Verhaltensmodell Infantry – Defend location.....	49
6.2.2 Das Verhaltensmodell Infantry – Pursue.....	51

6.2.3 Das Verhaltensmodell Infantry – Gun crew.....	52
6.3 Verhaltensmodelle für Panzer-Einheiten.....	52
6.3.1 Das Verhaltensmodell Tanks – Defend location.....	52
6.3.2 Das Verhaltensmodell Tanks – Pursue.....	52
6.4 Verhaltensmodelle für Transport-Einheiten.....	53
6.4.1 Das Verhaltensmodell Trucks – Help.....	53
6.4.2 Das Verhaltensmodell Trucks – Move to location.....	54
6.4.3 Das Verhaltensmodell Trucks – Transport from location to location.....	54
6.5 Verhaltensmodelle für andere Einheitentypen.....	55
6.5.1 Das Verhaltensmodell Maintenance – Repair.....	55
6.5.2 Das Verhaltensmodell Katyusha – Move to location.....	56
6.5.3 Das Verhaltensmodell Planes – Move to location.....	57
6.5.4 Das Verhaltensmodell Trains – Defend location.....	57
7 Skriptprogrammierung.....	59
7.1 Bedingungen und Boolesche Logik.....	59
7.1.1 Der Operator not.....	59
7.1.2 Die Operatoren and und or.....	60
7.1.3 Klammern in Bedingungen.....	61
7.2 Einmalige und wiederholte Ausführung von Skripten.....	61
7.3 Informationen für den Spieler.....	62
7.3.1 Textausgaben.....	62
7.3.2 Dialogboxen.....	63
7.3.3 Anzeige bestimmter Kartenbereiche.....	63
7.4 Zeitgesteuerte Skripte.....	64
7.4.1 Abhängigkeit vom Zeitpunkt des Missionsbeginns.....	64
7.4.2 Timer.....	64
7.4.3 Countdown.....	65
7.5 Skriptsteuerung von Einheiten.....	65
7.5.1 Aufenthaltsort von Einheiten.....	66
7.5.2 Anzahl von Einheiten.....	66
7.5.3 Zustand von Einheiten.....	67
7.5.4 KI-Verhalten.....	67
7.5.5 Patrouillen.....	68
7.5.6 Langstreckenfeuer.....	68
7.5.7 Transfer von Gruppen.....	68
7.5.8 Spezielle Bedingungen und Aktionen.....	68
7.6 Verstärkungen.....	69
7.6.1 Verstärkungen in Einzelspielermissionen.....	69
7.6.2 Verstärkungen in Mehrspielermissionen.....	69
7.7 Skriptsteuerung von Flugzeugen.....	70
7.7.1 Flugzeuge zuteilen.....	70

7.7.2 Flugzeuge auf die Karte schicken.....	70
7.7.3 Flugzeuge zwischen Flugplätzen wechseln lassen.....	71
7.7.4 Verwendung von Flugrouten.....	71
7.8 Missionsende und Kampagnen.....	71
7.8.1 Sieg und Niederlage.....	71
7.8.2 Folgemission in Kampagnen.....	71
7.8.3 Übertragung von Einheiten zwischen Missionen.....	72
7.9 Zellen (Variablen).....	72
7.9.1 Bedingungen und Aktionen.....	73
7.9.2 Einsatz von Zellen.....	73
7.10 Weitere Bedingungen und Aktionen.....	74
7.10.1 Zustand von Objekten.....	74
7.10.2 Zünden von Minen.....	74
7.11 Verwendete Abkürzungen.....	74
8 Der Briefing-Editor.....	76
8.1 Die Benutzeroberfläche des Briefing-Editors.....	76
8.2 Vorüberlegungen.....	78
8.3 Erstellen und Laden des Hintergrundbildes.....	78
8.4 Arbeiten mit Formen.....	79
8.4.1 Formen einfügen, löschen und kopieren.....	79
8.4.2 Formen bearbeiten.....	80
8.4.3 Formenspezifische Einstellungen.....	81
8.5 Animation erstellen.....	83
8.5.1 Prinzip der Animation.....	83
8.5.2 Keyframes bearbeiten.....	84
8.5.3 Animation erzeugen und speichern.....	85
8.5.4 Einbinden der Briefing-Animation in eine Mission.....	86
9 Anhang.....	87
9.1 Gegenüberstellung SSRW und SuStII.....	87
9.2 Fehlermeldungen.....	87
9.2.1 yy== -1 Internal Error.....	87
9.2.2 inpCur[curW]>=inpNum[curW].....	88
9.2.3 Fehler beim Speichern.....	88
9.3 Bugs.....	89
9.3.1 Mehrfaches Zählen von Einheiten.....	89
9.3.2 Einfrieren beim Löschen der letzten Einheit.....	89
9.3.3 Einfrieren bei Anlegen und Löschen großer Zonen.....	89
9.3.4 Optionen zum Verhaltensmodell Maintenance – Repair u.a.....	89
Index.....	90

1 Überblick

Ich bin der Meinung, dass ein einigermaßen planvolles und strukturiertes Vorgehen bei jeder Entwicklung, also auch bei der Entwicklung von Missionen für Sudden Strike, die Grundlage für ordentliche Ergebnisse darstellt. Da bei der Arbeit an einer Mission immer wieder neue Ideen auftauchen, kann es ohnehin relativ schnell unübersichtlich werden. Ohne vorangegangene Planung entsteht daraus dann schnell das pure Chaos. Bevor also der Editor zum Einsatz kommt, sollte man sich daher einige Gedanken zum Aufbau der Mission machen. Im Rahmen dieser Überlegungen könnte man sich z.B. zu folgenden Punkten einige Skizzen und/oder Notizen machen:

- In welcher Umgebung und zu welcher Jahreszeit spielt die Mission? Es bietet sich an, eine Skizze von den wesentlichen Landschaftselementen (Anhöhen, Flüsse, Wälder, Straßen und Siedlungen) anzufertigen. Die Skizze sollte groß genug sein, um weitere Notizen aufnehmen zu können. Es kann auch hilfreich sein, sich von der Skizze mehrere Kopien zu machen, in die dann Einheiten, wichtige Orte, Truppenbewegungen usw. eingetragen werden können. Die Jahreszeit beeinflusst die Wahl des Kartenschemas (vgl. 3.1.1) und der Wetterbedingungen (vgl. 4.2.4).
- Wie ist die Ausgangslage und was ist das Ziel der Mission? Auf dieser Grundlage können weitere Überlegungen angestellt und ggf. als Nebenziele formuliert werden. Damit hat man die ersten Anhaltspunkte dafür, wie die Mission ablaufen soll.
- Welche Einheiten sind auf der Seite des Spielers (auch verbündete Truppen) und seines Gegners beteiligt? Welche Aufgaben sollen die einzelnen Einheiten während der Mission haben?⁶
- In welchem Ausmaß erhalten die Teams Verstärkungen? Stehen Flugzeuge zur Verfügung? In diesem Zusammenhang kann man auch überlegen, an welche Ereignisse die Verstärkungen geknüpft sein sollen.
- Welche Orte sind für den Ablauf der Mission von besonderer Bedeutung? Wo stehen starke Truppenverbände? Welche Orte sind zu erreichen, zu zerstören oder zu befreien? Bei SSRW sind natürlich insbesondere die Nachschubstationen, Fabriken und Krankenhäuser von Bedeutung.

Nach diesen Überlegungen kann die Arbeit an der Mission beginnen. Eine spielbare Mission in Sudden Strike besteht aus einer *Karte* und der *Mission* selbst. Beides wird in dem gleichen Programm bearbeitet, im *Editor* edit3.exe, den man über Start > Programme > Sudden Strike – Resource War > Editor startet. Kapitel 2 enthält einen Überblick über die Benutzeroberfläche des Editors.

Die *Karte (Map)* enthält die statischen Missionsanteile. Dazu gehören das Terrain, die Vegetation, die Gebäude sowie eine Reihe weiterer statischer Objekte. Die Erstellung von Karten wird in Kapitel 3 beschrieben.

Die eigentliche *Mission* besteht aus den dynamischen Anteilen. Dazu gehören die Einheiten des Spielers und des Gegners sowie deren Verhalten. Das Verhalten der nicht vom Spieler gesteuerten Einheiten wird in allen außer den grundlegenden Situationen durch *Skripte* festgelegt. Auch die Skripte gehören zu der Mission. Da die Erstellung von Missionen ein komplexes Thema darstellt, habe ich die Erläuterungen auf mehrere Kapitel verteilt.

Kapitel 4 beschreibt die Editorfunktionen, die bei der Erstellung von Missionen zum Einsatz kommen, sowie einige grundlegende Missionseinstellungen. Auch die Erzeugung spielbarer Missionen wird beschrieben.

Die Vorbereitungen für die Skriptprogrammierung umfassen die Platzierung von Einheiten und die Einstellung ihres Verhaltens, das Definieren von Orten und anderes mehr. Dies ist der Inhalt von Kapitel 5.

Kapitel 6 umfasst eine ausführliche Beschreibung der möglichen Verhaltensweisen von Einheiten und der entsprechenden Optionen, die das Verhalten genauer festlegen.

⁶ Wer es etwas genauer nimmt, informiert sich für Missionen nach historischen Vorbildern ggf. über die zum Einsatz gekommenen Einheiten.

Die eigentliche *Skriptprogrammierung* wird in Kapitel 7 beschrieben. Hier werden die möglichen Bedingungen und Aktionen erläutert, aus denen ein Skript besteht.

Zu jeder Mission kann man ein *Briefing* erzeugen, in dem der Spieler Informationen über die Aufgaben während der Mission informiert wird. Sollen die Missionsziele nicht nur durch Text, sondern auch durch eine Animation erläutert werden, dann kommt der bei SuStII mitgelieferte Briefing-Editor zum Einsatz⁷. Seine Bedienung wird in Kapitel 8 erklärt.

In Kapitel 9 werden einige weitere Informationen zum Editor gegeben. So ist es ja z.B. durchaus nichts Neues, dass der Editor nicht gerade zu den stabilen Softwareprodukten gehört. Aus diesem Grund habe ich in diesem Kapitel einige Bugs und Fehlermeldungen zusammengestellt, die mir bei meiner bisherigen Beschäftigung mit dem Editor begegnet sind. Für die Benutzer von SuStII findet man auch eine Gegenüberstellung der Benutzerschnittstellen von SuStII und SSRW.

Man kann diesen User Guide natürlich von vorn bis hinten lesen. Der User Guide lässt sich aber auch als Nachschlagewerk nutzen. Dazu habe ich ein Stichwortverzeichnis angefügt, das die gezielte Suche nach Informationen erleichtern soll.

⁷ Bei SSRW wird der Briefing-Editor leider nicht mitgeliefert. Der Editor von SuSt II lässt sich aber auch für SSRW verwenden.

2 Die Benutzeroberfläche des Editors

Die Oberfläche des Editors ist für die Bearbeitung von Karten und Missionen die gleiche, es werden lediglich unterschiedliche Menübefehle benutzt. Die folgende Abbildung zeigt das *Editorfenster* mit seinen Bestandteilen.

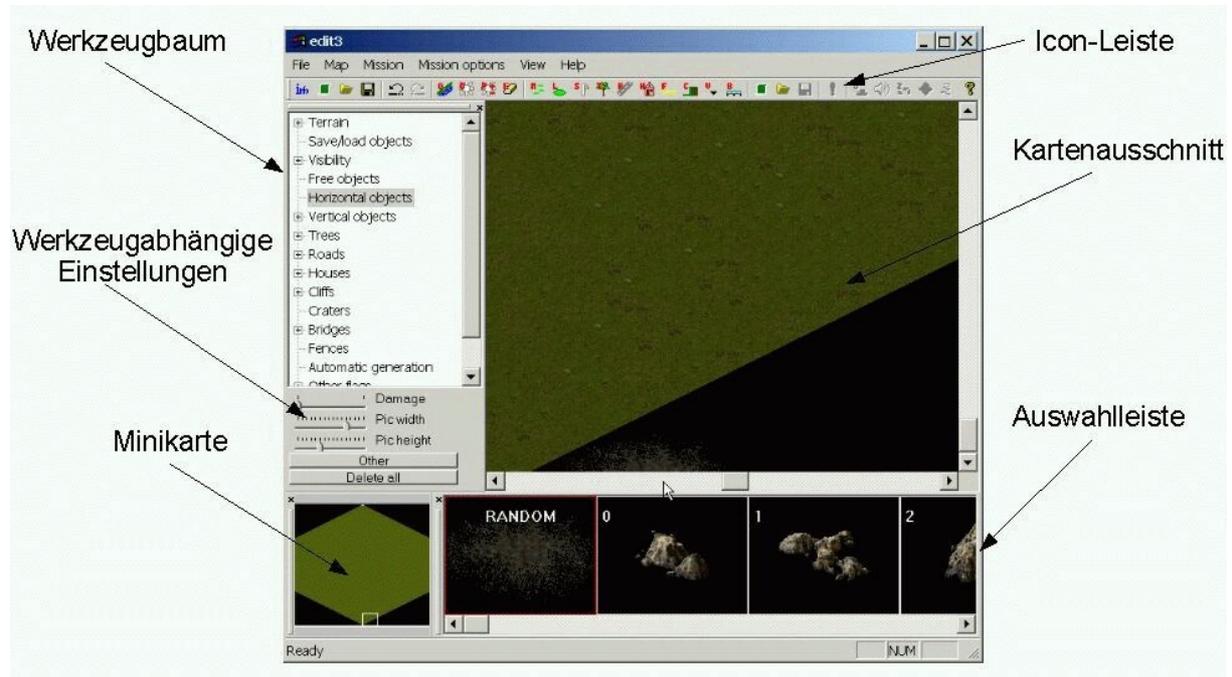


Abb. 1: Das Editorfenster

Der *Werkzeugbaum* enthält die verschiedenen Werkzeuge für die Gestaltung von Karten. Dazu gehören unter anderem Kategorien von statischen Objekten (z.B. Häuser und Bäume), die auf einer Karte platziert werden können. Einige Kategorien bestehen aus mehreren Unterkategorien. Sie sind mit einem Pluszeichen versehen. Durch Klick auf das Pluszeichen kann man die Anzeige der Unterkategorien aus- und einklappen. Unterhalb des Werkzeugbaums werden je nach aktuell gewähltem Werkzeug unterschiedliche Einstellungsmöglichkeiten angeboten.

Bei der Bearbeitung einer Mission kommen im Werkzeugbaum weitere Werkzeuge hinzu, die es ermöglichen, Einheiten zu setzen, Zonen festzulegen usw. Damit werden die Grundlagen für die Skripte geschaffen, die später den Ablauf der Mission bestimmen.

Neben dem Werkzeugbaum ist ein *Ausschnitt* der bearbeiteten Karte zu sehen. In diesem Teilfenster werden in einer gegenüber dem Spiel etwas verkleinerten Ansicht die Bestandteile der Karte sowie Einheiten, Zonen usw. platziert. Der angezeigte Ausschnitt kann über die Bildlaufleisten oder mit Hilfe der Cursortasten verschoben werden.

Die *Minikarte* (*Minimap*) gibt einen Überblick über die gesamte Karte. Der im eigentlichen Editorfenster angezeigte Kartenausschnitt ist in der Minikarte als weißes Rechteck markiert. Durch Mausklick an eine Stelle der Minikarte wird der Kartenausschnitt an diese Stelle verschoben, was zur schnellen Navigation auf der Karte nützlich ist.

Neben der Minikarte ist die *Auswahlleiste* zu sehen. Ihr Inhalt besteht aus den Objekten der im Werkzeugbaum ausgewählten Kategorie bzw. Unterkategorie. Hier können die zu platzierenden Objekte gewählt werden. Bei der Verwendung von Werkzeugen, die für die Bearbeitung der dynamischen Missionsanteile gedacht sind (Einheiten, Zonen usw.), wird die Auswahlleiste teilweise durch eine andere Darstellung ersetzt, die vom gerade gewählten Werkzeug abhängt (mehr dazu in Kapitel 5).

Viele der Funktionen des Editors lassen sich nicht nur über eines der Menüs, sondern auch über ein Icon aus der *Icon-Leiste* aufrufen. Die einzelnen Icons zeigt die folgende Abbildung.

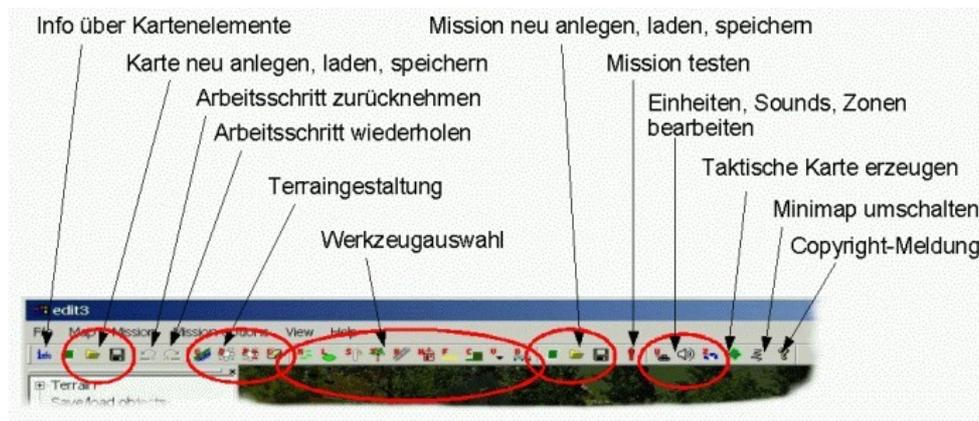


Abb. 2: Die Icon-Leiste

Die Icons im einzelnen (von links nach rechts):



Info über Kartenelemente (File > Show map info, Taste **F1**): Zeigt ein Fenster mit Informationen über die Anzahl der bereits verwendeten Objekte und Einheiten an.



Neue Karte erzeugen (Map > New map): Erzeugt eine neue Karte.



Karte laden (Map > Load map): Lädt eine bereits angelegte und gespeicherte Karte.



Karte speichern (Map > Save map): Speichert die aktuelle Karte.



Undo⁸ (Taste Strg-**Z**): Macht den letzten Arbeitsschritt rückgängig. Es lassen sich mehrere Arbeitsschritte rückgängig machen.



Redo (Taste Strg-**Y**): Stellt den zuletzt mit Undo gelöschten Arbeitsschritt wieder her.



Terrain (Taste **G**): Öffnet die Werkzeugkategorie Terrain im Werkzeugbaum.



Set brightness (Taste **B**): Auswahl des Werkzeugs zum Einstellen der Terrainhelligkeit.



Adjust brightness (Taste **L**): Auswahl des Werkzeugs Lighten/darken zum Aufhellen bzw. Abdunkeln des Terrains.



Eraser (Taste **E**): Auswahl des Werkzeugs Eraser zum Löschen von Kartenteilen.



Free objects (Taste **M**): Auswahl der Werkzeugkategorie Free objects (Steine usw.).



Horizontal objects (Taste **H**): Auswahl der Werkzeugkategorie Horizontal objects (flache Objekte und Bodentexturen).



Vertical objects (Taste **V**): Auswahl der Werkzeugkategorie Vertical objects (senkrecht stehende Objekte wie Telegrafmasten, Denkmäler usw.).



Trees (Taste **T**): Auswahl der Werkzeugkategorie Trees (Bäume und Büsche).

⁸ Im Editor von SuStII ist die Undo-Funktion (ebenso wie Redo) noch nicht vorhanden.



Roads (Taste **R**): Auswahl der Werkzeugkategorie Roads (Wege, Straßen und Bahnlinien sowie einige Bodentexturen).



Houses (Taste **H**): Auswahl der Werkzeugkategorie Houses (Gebäude).



Fences (Taste **F**): Auswahl der Werkzeugkategorie Fences (Zäune).



Cliffs (Taste **C**): Auswahl der Werkzeugkategorie Cliffs (Klippen und Flussufer).



Craters (Taste **V**): Auswahl der Werkzeugkategorie Craters (Krater von Geschosseinschlägen).



Bridges (Taste **B**): Auswahl der Werkzeugkategorie Bridges (Brücken).



Neue Mission erzeugen (Mission > New mission): Erzeugt eine neue Mission auf der aktuellen Karte.



Mission laden (Mission > Load mission): Lädt eine auf der aktuellen Karte bereits angelegte und gespeicherte Mission.



Mission speichern (Mission > Save mission): Speichert die aktuelle Mission.



Mission testen⁹ (Mission > Run game): Erzeugt aus Karten- und Missionsdaten eine spielbare Mission und startet diese aus dem Editor heraus in SSRW. Erleichtert das Testen, da man nicht für jeden Testlauf die Missionsdatei erzeugen, ins SSRW-Verzeichnis kopieren und SSRW starten muss.



Units (Taste **U**): Auswahl der Werkzeugkategorie Units (Einheiten) zum Plazieren von Einheiten auf der Karte.



Sounds: Auswahl der Werkzeugkategorie Sounds (Geräusche) zum Anlegen von Gebieten, in denen im Spiel Umgebungsgeräusche (Vogelgezwitscher, Wasserrauschen usw.) zu hören sein sollen.



Locations (Taste **Z**): Auswahl der Werkzeugkategorie Locations (Zonen). Dient zum Anlegen von Gebieten, die bei der Ausführung von Skripten wichtig sind.



Tactical map (Mission options > Generate tactical map): Setzt im Menü Mission options die Option Generate tactical map, so dass beim Erzeugen der Mission die taktische Karte erzeugt und eingebunden wird, die im Spiel über die Feststelltaste (bei SuStII: Taste M) eingeblendet werden kann.



Schaltet bei der Bearbeitung einer Mission die Minimap zwischen Anzeige der Karte und der Missionszone hin und her¹⁰.



About (Help > About edit3...): Zeigt die Copyright-Meldung des Editors an.

⁹ Diese Funktion gab es im Editor von SuStII noch nicht.

¹⁰ Danke an den Zermetzmeister!

3 Karten

3.1 Dateioperationen mit Karten

Die Dateioperationen lassen sich jeweils durch Auswahl des entsprechenden Menüpunktes oder durch Klick auf das entsprechende Icon auslösen. Zu beachten ist dabei, dass die Icons jeweils zweimal vorhanden sind, einmal für den Umgang mit Kartendateien (links in der Icon-Leiste) und einmal für den Umgang mit Missionsdateien (weiter rechts in der Icon-Leiste).

3.1.1 Anlegen einer neuen Karte

Eine neue *Karte* wird über Map > New map angelegt. Es erscheint ein Dialog, in dem man die gewünschte Größe der Karte und das Kartenschema auswählen kann (vgl. Abb. 3). Nach Klick auf OK wird die Karte entsprechend den gewählten Angaben erzeugt. Eine Änderung von Größe und Kartenschema ist dann im Editor nicht mehr möglich.

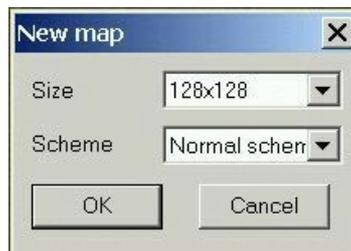


Abb. 3: Dialog zum Anlegen einer neuen Karte

Als *Kartengrößen* sind 128x128, 256x256, 384x384 und 512x512 Felder wählbar. Die geeignete Größe richtet sich unter anderem danach, ob die Karte als Grundlage einer einzelnen oder mehrerer Missionen dienen soll. Im ersten Fall sind meiner Meinung nach 128x128 und 256x256 besonders geeignet. Größere Karten werden leicht unübersichtlich. Zu bedenken ist auch, dass sich bei einer Verdoppelung der Seitenlänge der Karte die zu bearbeitende Fläche (und damit die Arbeit) vervierfacht. Außerdem läuft man bei großen Karten eher Gefahr, die Limits für Objekte und Einheiten zu überschreiten, was die Stabilität des Spiels negativ beeinflussen kann (Absturzgefahr). Andererseits haben große Karten natürlich einen gewissen Reiz, weil weiträumigere Operationen möglich sind und bei SSRW auch die Treibstoffversorgung an Bedeutung zunimmt.

Das *Kartenschema* bestimmt das generelle Aussehen der Karte und die zur Verfügung stehenden Objekte zur Kartengestaltung. Als Schema kann man Normal, Winter (relativ geringe Auswahl an Objekten), Island (geeignet für Pazifikmissionen) und Desert (für Afrikamissionen) wählen¹¹.

3.1.2 Speichern und Laden von Karten

Karten werden in einem von 255 Speicherplätzen (sog. *Slots*) gespeichert, die von 0 bis 254 durchnummeriert sind. Zum Speichern wählt man Map > Save map oder klickt auf das Disketten-Icon. In dem Dialog, der sich daraufhin öffnet, sind bereits belegte Slots durch ein vorangestelltes Sternchen gekennzeichnet, während freie Slots an dem Vermerk —empty— zu erkennen sind (vgl. Abb. 4). Man wählt einen Slot aus und klickt auf OK. In dem folgenden Dialog sollte man eine kurze Beschreibung der Karte angeben, um die Karte auch später leicht identifizieren zu können.

Wenn es beim Speichern einer Karte eine Fehlermeldung gibt, dann kann das daran liegen, dass im Editor-Verzeichnis der Ordner maps.src fehlt. Dieser Ordner wird zwar bei der Installation von SuStII standardmäßig angelegt, bei der von SSRW jedoch nicht¹². Man kann den Ordner dann ggf. von Hand anlegen. Dazu erzeugt man im Windows Explorer im Verzeichnis C:\Programme\Sudden Strike - Resource War\Run\Editor einen Unterordner mit dem Namen maps.src. Dann sollte alles klappen.

¹¹ Bei SuStII kann man nur zwischen Sommer (entspricht Normal) und Strand (entspricht Island) wählen.

¹² Bei der Installation des German Patch zu SSRW wird der Ordner ebenfalls angelegt.

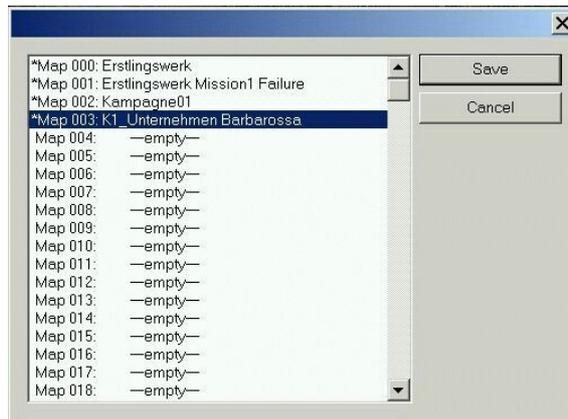


Abb. 4: Speichern einer Mission

Das *Laden* einer bereits angelegten und gespeicherten Karte erfolgt über Map > Load map oder durch Klick auf das Ordnericon. Im folgenden Dialog wählt man den Slot mit der gewünschten Karte aus und klickt auf OK.

3.1.3 Erzeugen einer Bitmap der Karte

Durch Map > Generate bitmap kann man eine Bilddatei im *.jpg- oder *.bmp-Format erzeugen, die eine verkleinerte Darstellung der Karte enthält. Unter Resolution kann man die Auflösung der Bilddatei einstellen. Der Regler Quality dient zur Einstellung der Bildqualität für das *.jpg-Format, während er für das *.bmp-Format wirkungslos ist¹³. Die Standardeinstellungen sind in der Regel gut geeignet.

3.2 Terrain

3.2.1 Terrain editieren

Als *Terrain* wird hier die *Textur* bezeichnet, die zur Darstellung des Geländeuntergrundes der Karte verwendet wird. Im Werkzeugbaum findet man bei Verwendung des Schemas Normal unter der Kategorie Terrain die Texturen Grass (Wiese), Plough (Ackerland), Asphalt (Straßenpflaster), Ground (Erde), Dry ground (Erde, hellere Variante), Water (Wasserflächen) und Sand (heller Sand). Bei einem anderen Schema findet man hier auch andere Texturen.



Abb. 5: Terraintexturen des Schemas Normal

Die ausgewählte Textur kann mit gedrückter linker Maustaste auf die Karte gezeichnet werden. Mit gedrückter rechter Maustaste wird die Standardtextur gezeichnet, was zum Löschen der anderen Texturen hilfreich ist. Mit dem Regler Brush size unter dem Werkzeugbaum kann die für das Zeichnen der Texturen verwendete *Pinselgröße* eingestellt werden (1x1, 3x3 oder 5x5 Felder). Die Wirkung unterschiedlicher Pinselgrößen sieht man in der folgenden Abbildung.

¹³ Das *.jpg-Format ist ein sog. verlustbehaftetes Bildformat, d.h. beim Speichern eines Bildes in diesem Format gehen Bildinformationen verloren, weil das zugrundeliegende Kompressionsverfahren einige Informationen zusammenfasst, um die Dateigröße zu minimieren. Das Ausmaß, in dem diese Verluste auftreten, kann man durch die Einstellung für die Bildqualität festlegen. Das *.bmp-Format ist dagegen verlustfrei, so dass diese Einstellung keine Auswirkungen hat. Dafür sind die erzeugten Dateien relativ groß.

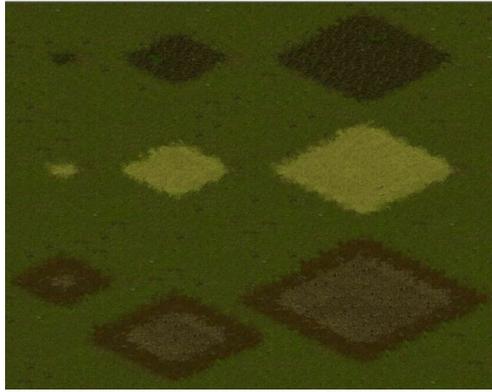


Abb. 6: Terraintexturen Plough, Sand und Dry ground in den Pinselgrößen 1x1, 3x3 und 5x5. Bei Dry ground (unten) wird automatisch ein Rand in der Textur Ground angelegt.

3.2.2 Helligkeit des Terrains

Um eine weniger eintönige Darstellung des Terrains zu erhalten, gibt es verschiedene Möglichkeiten, die *Helligkeit* des Terrains zu variieren.

Das Werkzeug Set brightness dient dazu, die Helligkeit des Terrains auf den über den Regler Brightness eingestellten Wert zu setzen. Dazu wird die Helligkeitseinstellung wie beim Einzeichnen von Texturen mit dem Pinsel auf die Karte „gemalt“.



Abb. 7: Wirkung von Set brightness auf die Texturen Grass, Plough und Sand (jeweils von links nach rechts: Brightness-Regler ganz links, mittig, ganz rechts)

Zur Realisierung gradueller Helligkeitsunterschiede dient das Werkzeug Lighten/darken. Mit gedrückter linker Maustaste wird das Terrain aufgehellt, mit gedrückter rechter Maustaste wird es abgedunkelt. Mehrfaches Übermalen des Terrains bewirkt eine Verstärkung des jeweiligen Effekts. Ansonsten ist die Wirkung wie bei Set brightness.



Abb. 8: Andeutung unterschiedlicher Wassertiefen durch Helligkeitsunterschiede

Die Werkzeuge zur Änderung der Helligkeit können z.B. bei größeren Wasserflächen zum Einsatz kommen, um unterschiedliche Wassertiefen anzudeuten. Ein Beispiel ist in Abb. 8 dargestellt.

Eine einfache Möglichkeit, Helligkeitsvariationen für größere Kartenbereiche zu realisieren, bietet der Menüpunkt File > Wave map¹⁴. Nach Auswahl erscheint ein Dialog, in dem einige Optionen wählbar sind. Im oberen Teil des Dialogs kann man die Terraintexturen auswählen, auf die sich die Helligkeitsvariation auswirken soll. Ist Generate only on selection markiert, wird nur der ausgewählte Kartenbereich (vgl. 3.7.1) bearbeitet, andernfalls die gesamte Karte. Die Stärke des Effekts wird über den Regler Density eingestellt.

Auf den angegebenen Flächen wird dann nach Klick auf OK ein zufälliges Muster von Hell- und Dunkelstufen generiert. Ein mögliches Ergebnis sieht man in Abb. 9. Da das Ergebnis vom Zufall bestimmt wird, ist in den meisten Fällen die manuelle Bearbeitung der Helligkeit vorzuziehen.



Abb. 9: Wirkung der Funktion Wave map auf eine Wasserfläche

3.2.3 Weitere Ausgestaltung des Terrains

Auch mit Helligkeitsunterschieden wirkt das Terrain oft noch recht eintönig. Abhilfe schafft die Platzierung von weiteren Texturen, die man im Werkzeugbaum unter Horizontal objects findet (vgl. 3.3). Dort und unter Free objects sind auch Felsen und ähnliche Objekte zu finden, die ebenfalls zur optischen Auflockerung des Terrains beitragen können.

3.3 Statische Objekte

3.3.1 Plazieren von Objekten

Statische Objekte (Kategorien Free objects, Horizontal objects, Vertical objects, Trees, Roads, Houses, Cliffs, Craters, Bridges, Fences) werden in der Auswahlleiste angewählt und dann durch Linksklick im Kartenausschnitt an die gewünschte Stelle gesetzt. Das gewählte Objekt wird dabei als Mauscursor verwendet. In jeder Kategorie gibt es ein Pseudoobjekt, das in der Auswahlleiste ganz links steht und mit RANDOM gekennzeichnet ist. Ist dieses Objekt ausgewählt, dann wird nach jedem plazierten Objekt der Kategorie ein weiteres Objekt der Kategorie zufällig gewählt. Soll das zufällig bestimmte Objekt nicht gesetzt werden, kann durch Klick auf den Button Other unter dem Werkzeugbaum oder durch Drücken der Tab-Taste ein anderes Objekt zufällig bestimmt werden. Die Verwendung dieses Objekts ist z.B. bei der Erstellung von Wäldern und der Ausgestaltung größerer Flächen mit unterschiedlichen Texturen oder kleinen Objekten wie Felsen sinnvoll, immer dann also, wenn Eintönigkeit vermieden werden soll.



Abb. 10: Auswahlleiste der Kategorie Houses, Unterkategorie [1] (Schema Normal)

¹⁴ Diesen Menüpunkt gibt es nur in SSRW.

Beim Setzen von Objekten ist zu berücksichtigen, dass ein Rand von vier Feldern um die Missionszone (erkennbar am Helligkeitsunterschied) nicht in die spielbare Mission übernommen wird. Objekte sollten daher nicht in dieser Randzone platziert werden (vgl. Abb. 11).



Abb. 11: Platzierung von statischen Objekten am Rand der Missionszone im Editor (links) und im Spiel (rechts): Das weiße Gebäude rechts unten ist im Spiel nicht zu sehen, da es außerhalb der Missionszone liegt.

Bei dicht beieinander platzierten Objekten (z.B. bei Wäldern) ist es manchmal schwierig, zu erkennen, wo noch Objekte platziert werden können. In solchen Fällen kann es hilfreich sein, durch Druck auf **F4** die bereits platzierten Objekte auszublenden (vgl. Abb. 12). Es sind dann nur noch die Schatten der Objekte zu sehen, was den Überblick erleichtert.



Abb. 12: Ausblenden statischer Objekte durch Drücken von F4

Unter dem Werkzeugbaum findet man je nach Kategorie verschiedene Regler. Für zerstörbare Objekte wie z.B. Gebäude kann man mit dem Regler Damage den *Schadenzustand* des Objekts einstellen. Die verschiedenen Schadensstufen kann man auch durch Druck auf die Leertaste durchlaufen. Die Regler Pic width und Pic height beeinflussen die Darstellungsgröße der Objekte in der Auswahlliste.

Objektart	Objekte insges.	Häuser	Brücken	Einheiten
Max. Anzahl	50000	800	32	1000

Tabelle 1: Maximal erlaubte Anzahl von Objekten

Beim Platieren von Objekten sollte man beachten, dass Sudden Strike nur eine gewisse Anzahl von Objekten verwalten kann. Die aktuelle Anzahl der Objekte kann über File > Show map info, durch Klick auf das Info-Icon oder durch Drücken von **F4** angezeigt werden. Die in der Anzeige angege-

benen Obergrenzen findet man in der folgenden Tabelle. Besonders problematisch ist die Begrenzung bei den Brücken, denn die angegebene Anzahl bezieht sich nicht auf das Gesamtbauwerk, sondern auf die Teile, aus denen die fertige Brücke besteht.

3.3.2 Löschen von Objekten

Ein bereits gesetztes Objekt kann gelöscht werden, indem man die entsprechende Kategorie auswählt und mit der rechten Maustaste auf die Position des zu löschenden Objekts klickt. Sollen Objekte unterschiedlicher Kategorien gelöscht werden, bietet sich die Verwendung des Radiergummis an. Dieses Werkzeug findet man in der Kategorie Terrain unter Eraser.

3.4 Automatische Generierung von Klippen, Straßen und Bahnlinien

Das Anlegen von *Klippen*, *Straßen* und *Bahnlinien* kann mühsam sein. Daher gibt es ein Hilfsmittel zur automatischen Erzeugung solcher Kartenbestandteile. Besonders bei der Erzeugung von Klippen und Bahnlinien führt die automatische Generierung zu guten Ergebnissen. Die Vorgehensweise ist in allen Fällen identisch. Die Auswahlmöglichkeiten sind allerdings in den verschiedenen Kartenschemata recht unterschiedlich. Das Schema Winter z.B. bietet sogar überhaupt keine Möglichkeit zur automatischen Erzeugung von Klippen und Straßen.

3.4.1 Vorgehensweise

Nach Auswählen der Kategorie Automatic generation kann man unterhalb des Werkzeugbaums im Auswahlfeld Cliff type die Art des zu erzeugenden Kartenelements auswählen. Dabei ist Cliff 0 für Berge und Cliff 1 für Flussufer geeignet. Da die Einstellung bei jeder Auswahl von Automatic generation wieder auf Cliff 0 zurückspringt, sollte man vor der Erzeugung von Flussufern noch einmal prüfen, ob tatsächlich Cliff 1 eingestellt wurde, da sonst die Spiegelung der Klippe im Wasser fehlt (vgl. Abb. 13). Für die Erzeugung von Bahnlinien ist Road 2 auszuwählen.

Die ungefähre Form des Kartenelements wird durch einen Polygonzug festgelegt, den man durch Setzen von Punkten auf der Karte bestimmt. Die Punkte setzt man durch Linksklick. Hat man einen Punkt falsch gesetzt, kann man durch Rechtsklick den Polygonzug neu anlegen. Will man nicht gleich den ganzen Polygonzug neu aufbauen, kann man alternativ auf den Button Delete last node klicken und den zuletzt gesetzten Punkt korrigieren. Durch mehrmaligen Klick auf den Button kann man auch mehrere Punkte wieder entfernen.

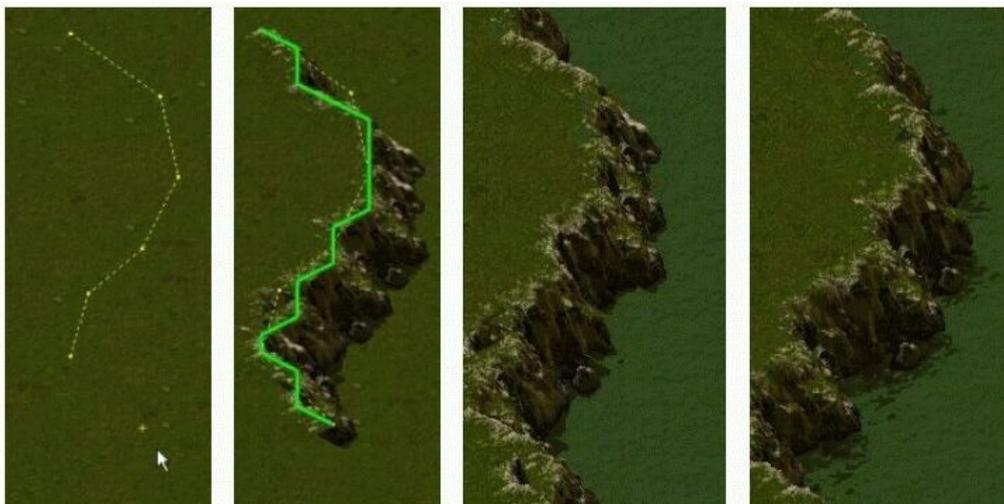


Abb. 13: Automatische Erzeugung von Klippen (von links nach rechts: Polygonzug, Vorschlag für die automatisch erzeugte Klippe, Cliff 0 ohne Spiegelung, Cliff 1 mit Spiegelung)

Ist der Polygonzug fertig, kann durch Klick auf den Button Build oder durch einen Linksklick auf den Kartenausschnitt bei gehaltener Umschalttaste die automatische Generierung des Kartenelements gestartet werden. Es wird ein Vorschlag für das Kartenelement angezeigt und durch einen grünen Polygonzug gekennzeichnet. Dieser Vorschlag kann durch Rechtsklick auf den Kartenausschnitt angenommen werden. Der grüne Polygonzug verschwindet und das angelegte Kartenelement wird in die Karte

übernommen. Ist der Vorschlag nicht zufriedenstellend, kann man durch erneuten Linksklick auf den Kartenausschnitt bei gehaltener Umschalttaste bzw. erneuten Klick auf den Build-Button einen neuen Vorschlag anfordern oder durch Rechtsklick in den Kartenausschnitt bei gehaltener Umschalttaste zum Editieren des Polygonzuges zurückkehren. Dabei bleiben leider manchmal Reste des Vorschlags zurück, die manuell z.B. mit dem Radiergummi entfernt werden müssen.

Die Form des Ergebnisses lässt sich durch die beiden Regler *Curv priority* und *Exact end* in gewissem Umfang beeinflussen. Durch *Curv priority* wird festgelegt, wie weit das Ergebnis sich an den vorgegebenen Polygonzug hält bzw. wie weit es sich davon entfernt, um eine geradere Darstellung zu erreichen. Die Einstellung bei *Exact end* gibt an, wie weit sich das Ergebnis an den Endpunkten des Polygonzugs von der Vorgabe entfernen darf.

3.4.2 Erzeugung von Klippen

Bei der Erzeugung von Klippen kann man zwischen *erhöhten* und *abgesenkten* Klippen wählen. Dazu setzt man die Punkte des formgebenden Polygonzugs im Uhrzeigersinn für eine erhöhte, entgegen dem Uhrzeigersinn für eine abgesenkte Klippe. Durch Klick auf *Reverse* lässt sich die Reihenfolge umkehren.

Bei der Erzeugung langer Klippen ist es in der Regel einfacher, die Klippe in mehreren Teilschritten aufzubauen. Dieses Vorgehen ermöglicht eine bessere Kontrolle über das Ergebnis. Außerdem ist es weniger ärgerlich, wenn man nur das letzte Teilstück korrigieren muss.

Nach der Fertigstellung der Klippe sollte man die Blockierungsflags entlang der Klippe kontrollieren. Es kommt häufig vor, dass bei der automatischen Erzeugung von Klippen Lücken entstehen (vgl. Abb. 14), die der Infanterie das Betreten von Klippen ermöglicht, wo das eigentlich nicht gewünscht ist. Solche Lücken kann man aber durch manuell gesetzte Blockierungen schließen (vgl. 3.6.2).



Abb. 14: Automatisch erzeugte Klippe mit Lücken

3.4.3 Erzeugung von Bahnlinien

Beim Bau längerer *Bahnlinien* ist es hilfreich, diese stückweise zu erstellen. Die Regler *Curv priority* und *Exact end* schiebt man beide ganz nach rechts. Das neue Teilstück baut man dann mit einem Polygonzug aus nur zwei Punkten auf, wobei man den ersten Punkt ungefähr auf die Verlängerung der Bahnlinie und den zweiten auf das exakte Ende der bisher aufgebauten Strecke setzt. Auf diese Weise wird das Gleis exakt in gerader Linie verlängert. Andere Vorgehensweisen führen oft zu Unterbrechungen oder Knicken in der Gleisführung.

3.5 Flugplätze

Da die Möglichkeit besteht, dem Spieler Flugzeuge nicht nur über das Spielmenü, sondern auch direkt auf der Karte zur Verfügung zu stellen, muss man für diese Situation mindestens einen *Flugplatz* anlegen. Zum Anlegen von Flugplätzen dient das Werkzeug *Airfields* mit seinen Unterkategorien. Zunächst markiert man das Gelände, das für Flugzeuge benutzbar sein soll. Dazu wählt man die Unterkategorie *Territory* aus. Wie bei der Gestaltung des Terrains kann man dann einen Pinsel in unterschiedlichen Breiten verwenden, um das Gelände zu markieren. Die markierten Flächen werden in

gelb dargestellt (vgl. Abb. 15). Es ist übrigens nicht notwendig, das *Flugplatzgelände* in anderer Weise (z.B. mit Terraintexturen oder Straßen) darzustellen, auch wenn das natürlich besser aussieht.

Wird mehr als ein Flugplatz benötigt, dann kann man in dem Auswahlfeld *Airfields* unterhalb des Werkzeugbaums für jeden weiteren Flugplatz eine neue Nummer vergeben. Insgesamt kann man auf einer Karte bis zu 16 Flugplätze erstellen.

Beim Markieren des Flugplatzgeländes ist zu beachten, dass ein Flugzeug einen Platz von 3x3 Feldern einnimmt, sodass die Abstellplätze und Rollbahnen wenigstens 5 Felder breit sein sollten, damit das Starten und Landen von Flugzeugen keine Probleme bereitet.

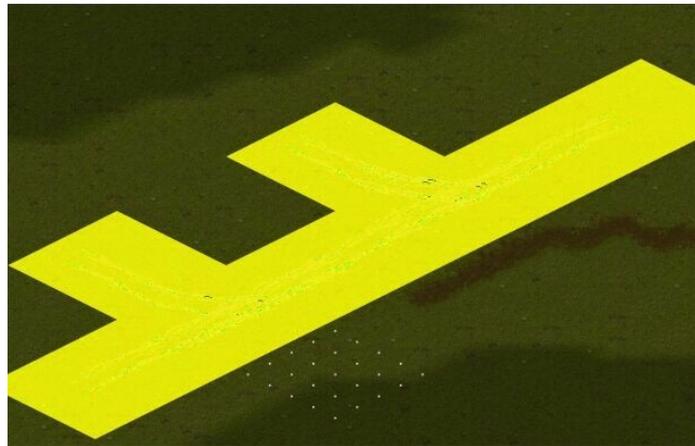


Abb. 15: Darstellung des Flugplatzgeländes im Editor

Neben dem Gelände gehören zu einem Flugplatz noch mindestens ein *Hangar*, ein *Start-* und ein *Landepunkt*. Ein Hangar in diesem Zusammenhang ist kein Gebäude, sondern wie das Flugplatzgelände ein Flag auf der Karte. Es gibt an, wo ein Flugzeug abgestellt werden kann. Einen Hangar legt man an, indem man die Unterkategorie *Hangars* im Werkzeugbaum auswählt und dann durch Linksklick an der gewünschten Stelle eine Hangarmarkierung (rot dargestellt) setzt. Es sind bis zu 16 solcher Markierungen pro Flugplatz erlaubt. Gelöscht wird eine Hangarmarkierung mit Rechtsklick auf die Markierung.

Die Hangarmarkierung enthält die Nummer des Flugplatzes und ein kleines weißes Quadrat, das die Ausrichtung des Flugzeugs festlegt, das in dem Hangar abgestellt wird. Durch Linksklick auf die Hangarmarkierung kann die Position des weißen Quadrats und damit die Ausrichtung des Flugzeugs wie gewünscht angepasst werden. Ein Beispiel ist in Abb. 16 zu sehen.

Flugzeuge, die schon zu Beginn der Mission auf der Karte vorhanden sein sollen, müssen auf eine Hangarmarkierung gesetzt werden. Da beim Plazieren von Einheiten diese Markierungen normalerweise nicht zu sehen sind, kann man sie durch Druck auf  einblenden.

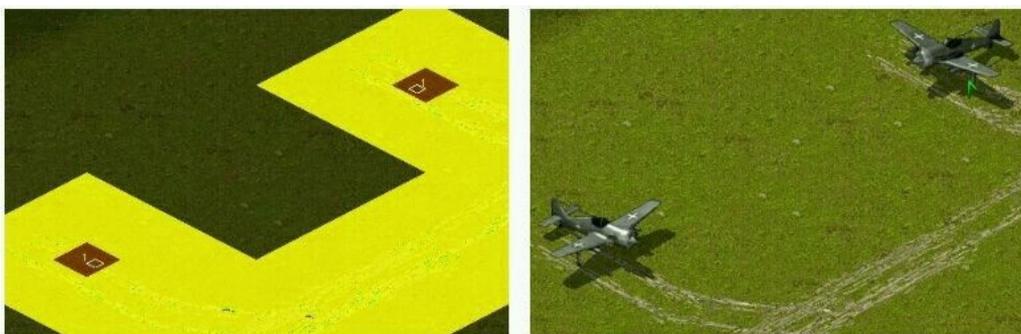


Abb. 16: Hangarmarkierungen und die Ausrichtung von Flugzeugen

Auf die gleiche Art und Weise wie die Hangarmarkierungen wird auf jedem Flugplatz noch eine *Start-* und eine *Landemarkierung* gesetzt, die festlegen, wo die Flugzeuge abheben bzw. landen sollen. *Startmarkierungen* (Unterkategorie *Liftoff point*) werden grün, *Landemarkierungen* (Unterkategorie *Landing point*) hellblau dargestellt. Ein Beispiel zeigt Abb. 17.



Abb. 17: Start- und Landemarkierung (links), Start (Mitte) und Landung (rechts) im Spiel

Hangar-, Start- und Landemarkierungen sollten so gesetzt werden, dass um sie herum noch mindestens zwei Felder des Flugplatzgeländes liegen (vgl. Abb. 18). Auch bei der Positionierung von Flugplatzgebäuden sollte man einen entsprechenden Abstand einhalten. Ist das nicht der Fall, dann kann es passieren, dass die Flugzeuge nicht abheben oder landen.

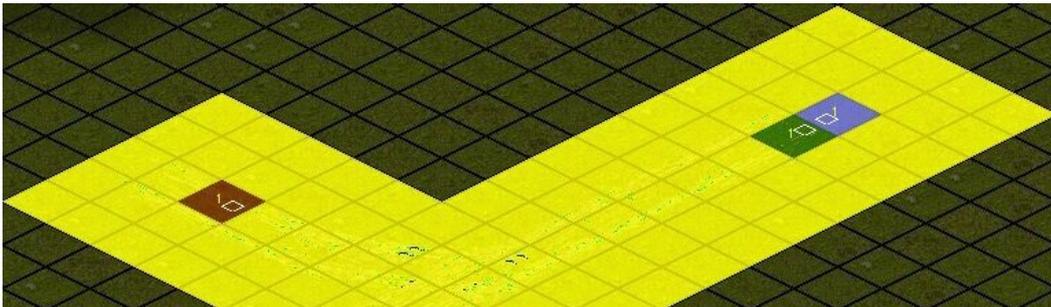


Abb. 18: Position von Hangar-, Start- und Landemarkierung

3.6 Sichtbarkeit und andere Flags

Sudden Strike verwaltet eine Reihe von *Flags* für die Felder der Karte, die die Sichtweite und Wegfindung von Einheiten beeinflussen können. Durch Drücken von  kann man das Raster der Karte und die gesetzten Flags einblenden (s. Abb. 19). Jedes Feld ist dabei in vier *Zellen* oder *Teilfelder* aufgeteilt. Einige Flags wirken auf ganze Felder, andere dagegen auf der Ebene der Teilfelder.

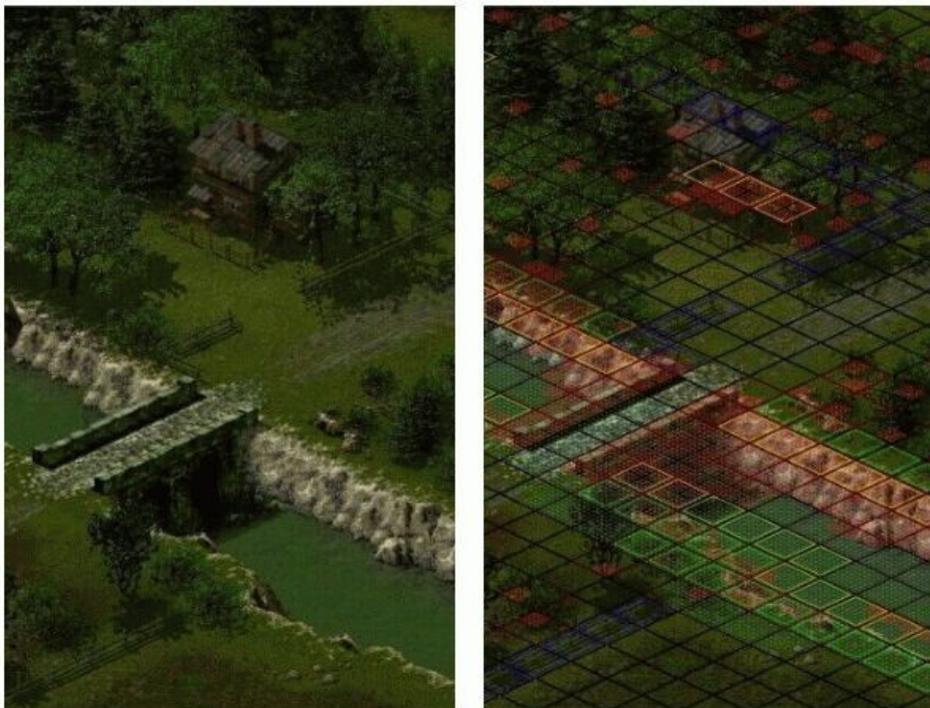


Abb. 19: Kartenausschnitt ohne (links) und mit eingeblendeten Informationen (rechts)

3.6.1 Sichtbarkeit

Um die Sichtbarkeitsverhältnisse auf der Karte festzulegen, markiert man einzelne Felder der Karte als *erhöht* (*raised*) oder *abgesenkt* (*lowered*). In der Abb. 19 erkennt man erhöhte Felder an den grünen, abgesenkte an den gelben Rahmen. Im Regelfall grenzen die beiden Arten von Sichtbarkeitsflags direkt aneinander an (vgl. Abb. 20 rechts). Diese Markierungen (*Flags*) beeinflussen die *Sichtbarkeit* von Einheiten auf der Karte wie folgt:

- Trifft die von einer Einheit ausgehende Sichtlinie zuerst auf eine Absenkung und dann auf eine Erhöhung, dann ist für die Einheit alles hinter der Absenkung nicht sichtbar.
- Trifft die von einer Einheit ausgehende Sichtlinie zuerst auf eine Erhöhung und dann auf eine Absenkung, dann ist für die Einheit alles hinter den Markierungen sichtbar.

In der Kategorie Visibility wählt man das entsprechende Werkzeug aus und kann dann durch Linksklick Markierungen setzen und durch Rechtsklick wieder entfernen. Absenkungen werden als gelbe Felder, Erhöhungen als grüne Rahmen dargestellt, wenn die Kategorie Visibility gewählt ist.

An Klippen werden Sichtbarkeitsflags automatisch gesetzt, wobei manchmal *Sichtlücken* entstehen, die nicht erwünscht sind (vgl. Abb. 20 links und Mitte). In solchen Fällen kann es nötig sein, diese Flags zu überarbeiten. Man entfernt das automatisch gesetzte Flag mit einem Linksklick und macht die Löschung mit einem Rechtsklick rückgängig. Die Maustastenbelegung ist also gegenüber dem manuellen Setzen der Sichtbarkeitsflags vertauscht.



Abb. 20: Sichtbarkeitsflags an einer Klippe (links: automatisch gesetzte Flags, Mitte: dadurch entstandene Sichtlücke im Spiel, rechts: korrigierte Flags)

3.6.2 Blockierte Felder

Durch das Setzen statischer Objekte werden deren Standorte *blockiert*, d.h. sie sind für Einheiten nicht mehr passierbar. Die blockierten Felder kann man sich im Editor anzeigen lassen, indem man in der Kategorie Other flags die Unterkategorie Blocked auswählt oder die Taste **F3** drückt. Felder bzw. Teilfelder, die für alle Einheiten blockiert sind, werden in rot angezeigt. Blaue Rahmen kennzeichnen Felder, die für schwere Einheiten (z.B. Panzer), aber nicht für leichte Einheiten (z.B. Infanterie) passierbar sind. Diese Markierungen sind in Abb. 21 zu sehen.



Abb. 21: Beispiele für blockierte Felder: Blockiert für alle Einheiten (rot), für leichte Einheiten (blau), manuell blockierter Krater (gelb)

Mit Linksklick lassen sich zusätzlich zu den automatisch erzeugten Blockierungsflags weitere setzen und mit Rechtsklick wieder löschen. Die manuell gesetzten Blockierungsflags wirken auf der Ebene der Teilfelder und werden orangegelb gekennzeichnet. Dies kann z.B. genutzt werden, um tiefe Krater oder ähnliche Hindernisse unpassierbar zu machen¹⁵.

Die Breite der manuell gesetzten Flags ist abhängig von der zuletzt benutzten Pinselgröße. Leider fehlt an dieser Stelle die Möglichkeit, die Pinselgröße zu verändern, so dass man ggf. in eine andere Kategorie (z.B. Terrain) wechseln und die Pinselgröße dort anpassen muss.

3.6.3 Wellendarstellung unterdrücken

Auf Wasserflächen werden Wellen dargestellt, um einen realeren Eindruck zu erwecken. In bestimmten Bereichen (z.B. am Rand von Klippen) wird diese Darstellung automatisch unterdrückt, damit die Wellen auf die reine Wasserfläche beschränkt bleiben. Wählt man in der Kategorie Other flags die Unterkategorie No waves, so werden die entsprechenden Flags als blau umrandete Felder gekennzeichnet.

Man kann durch Linksklick zusätzliche Flags setzen und durch Rechtsklick wieder löschen, auch wenn dies im Regelfall nicht notwendig werden sollte. Manuell gesetzte Flags werden als blau ausgefüllte Felder dargestellt.

3.6.4 Seichte Wasserflächen

Normalerweise sind Wasserflächen für Landeinheiten unpassierbar. Die Ausdehnung der Wasserfläche ist bei Drücken von **F3** am hellblauen Punktraster erkennbar. Will man einzelne Stellen passierbar machen, kann man entsprechende Flags durch Linksklick setzen und durch Rechtsklick entfernen, wenn man in der Kategorie Other flags die Unterkategorie Shallows auswählt. Die entsprechend mit Flags versehenen Felder werden hellblau umrahmt dargestellt (vgl. Abb. 22). Sie bleiben für Boote weiterhin passierbar.

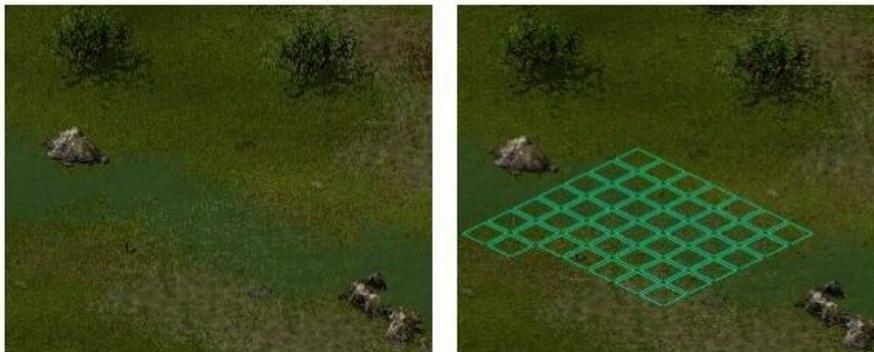


Abb. 22: Passierbarer Wasserlauf (durch Bodentextur für den Spieler kenntlich gemacht)

Eine interessante Anwendung ist das Setzen von Shallows-Flags auf Wasserpfützen aus der Kategorie Horizontal Objects. Wenn Infanterie eine solche Pfütze überquert, hört man das Platschen der Schritte im Wasser – sehr niedlich!

3.6.5 Bau von Pontonbrücken verhindern

Beim Anlegen von Wasserflächen auf der Kartenebene entstehen Stellen an denen *Pontonbrücken* gebaut werden können. Das scheint für alle Stellen zu gelten, an denen die beiden Ufer auf einer Breite von wenigstens drei Feldern gerade und parallel zueinander verlaufen. Wenn man **F3** drückt, werden diese Stellen durch eine weiße Linie von Ufer zu Ufer angedeutet.

Soll der Bau von Pontonbrücken an solchen Stellen nicht möglich sein, wählt man aus der Kategorie Other flags die Unterkategorie No pontoons. Durch Linksklick auf den Kartenausschnitt kann man dann entsprechende Flags setzen und durch Rechtsklick wieder löschen. Die gesetzten Flags werden

¹⁵ In einigen Fällen ergab sich bei meinen Karten das Problem, dass die manuell gesetzten Blockierungen keine Wirkung hatten. Leider konnte ich trotz vieler Versuche keine konkrete Ursache und – schlimmer – keine Abhilfe finden.

als pinkfarbene Felder dargestellt. Es reicht aus, ein Ufer mit entsprechenden Markierungen zu versehen, um den Brückenbau zu unterbinden.



Abb. 23: Bau von Pontonbrücken verhindern (von links nach rechts: Gerade Ufer, eingblendete Möglichkeit des Brückenbaus, Flag No pontoons, Brückenbau wird durch gesetztes Flag verhindert)

3.6.6 Undurchdringliche Felder

Die letzte Unterkategorie der Kategorie Other flags ist Fullstop. Mit einem Fullstop-Flag markierte Felder sind für Einheiten undurchsichtig. Gibt man einer Einheit, die auf einer Seite einer Fullstop-Markierung steht, den Befehl, auf die andere Seite zu schießen, dann wird sich die Einheit zunächst bis zu der Markierung bewegen und erst dann schießen. Bei Auswahl dieser Unterkategorie kann man solche Flags durch Linksklick manuell setzen und durch Rechtsklick wieder löschen. Im Regelfall ist dies jedoch nicht notwendig, da entsprechende Flags automatisch passend gesetzt werden.

3.7 Wiederverwendung von Kartenausschnitten

Die Erstellung ansprechender Karten kann sehr viel Arbeit machen. Aus diesem Grund kann man Teile der Karte als Block von Objekten speichern und bei der Erstellung weiterer Karten wiederverwenden. Mit der Zeit kann man sich so eine ganze Bibliothek von Kartenteilen anlegen, die den Aufwand für die Kartenerstellung erheblich reduzieren kann. Die dazu benötigten Funktionen des Editors verbergen sich hinter der Kategorie Save/load objects.

3.7.1 Speichern von Objektblöcken

Bevor man einen *Objektblock* speichern kann, muss man den entsprechenden Kartenbereich markieren. Durch Klick auf den Button Sel all (Select all) wird die gesamte Karte markiert, durch Klick auf Sel mis (Select mission) die gesamte Missionszone (falls eine Mission im Editor geöffnet ist). Soll ein kleinerer Bereich der Karte markiert werden, kann man wie bei der Terrainerstellung mit einem Pinsel den gewünschten Bereich markieren. Bei gehaltener linker Maustaste werden Markierungen gesetzt, bei gehaltener rechter Maustaste werden sie wieder entfernt. Die gesamte Markierung kann durch Klick auf Deselect rückgängig gemacht werden.

Ist der gewünschte Bereich der Karte markiert, klickt man auf den Button Save, gibt einen Dateinamen für den Objektblock an und klickt auf Speichern. Die Datei erhält die Dateierweiterung *.ldf.



Abb. 24: Speichern und Laden von Objektblöcken (von links nach rechts: Objektblock, markierter Block, geladener Block – leider ohne Terraintextur und Beleuchtung)

3.7.2 Laden von Objektblöcken

Klickt man auf den Button Load, öffnet sich der Dateidialog zum Laden eines Objektblocks. Nach Auswahl des gewünschten Blocks klickt man auf Öffnen und der Block wird geladen, aber noch nicht plaziert. Zur Platzierung des geladenen Blocks drückt man die Strg-Taste. Der Cursor wird dann durch den geladenen Block ersetzt. Durch Linksklick an die gewünschte Position kann der Block dort plaziert werden. Durch Rechtsklick an der gleichen Stelle kann das Platzieren des Blocks rückgängig gemacht werden.

Soll der Block als Zentrum einer neuen Karte dienen, kann man alternativ auf den Button Load at center klicken. Die Vorgehensweise ist dann ähnlich wie oben beschrieben. Der Unterschied besteht darin, dass der geladene Block nicht manuell plaziert werden muss, sondern automatisch in der Mitte der Karte eingefügt wird. Dieser Modus ist besonders zum Kopieren ganzer Missionen geeignet (vgl. dazu die Originaldokumentation zum Editor).

Oft ist nach dem Laden eines Objektblocks noch etwas Nacharbeit erforderlich, da Terraintexturen, Helligkeitsinformationen und Zäune nicht Bestandteil des Blocks sind und daher beim Speichern verloren gehen. Objekte wie Bodentexturen und Wege bleiben aber erhalten, so dass die Nacharbeit sich in Grenzen hält.

3.7.3 Beispiele für die Verwendung von Objektblöcken

Objektblöcke vereinfachen die Erstellung von großen Wäldern. Man baut sich auf einer Karte z.B. einige Waldstücke unterschiedlicher Art (Nadelwald, Laubwald, Herbstwald usw.) mit Bodentexturen, Steinen und Baumstämmen, speichert diese Waldstücke als Blöcke und kann sich dann auf anderen Karten große Wälder schnell „zusammenstempeln“. Damit das Muster nicht zu schnell erkennbar wird, kann man mehrere unterschiedliche Stücke einer Art speichern und nach der „Stempelaktion“ hier und da einzelne Bäume löschen oder hinzufügen.

Gut geeignet sind Objektblöcke auch für oft wiederkehrende Kartenelemente wie z.B. Geschützstellungen mit Sandsäcken usw. Diese Elemente werden so oft benötigt und sind dabei so ähnlich, dass man gut mit vorgefertigten Objektblöcken arbeiten kann, die man dann ggf. leicht verändert. In der folgenden Abbildung sieht man, wie man mit kleinen Änderungen ganz unterschiedliche Stellungen erzeugen kann. Als Grundlage diente in allen Beispielen der Objektblock aus Abb. 24.



Abb. 25: Demonstration zur Verwendung von Objektblöcken

3.8 Kartengestaltung

Was die Gestaltung der Karte angeht, hat wohl jeder seine eigenen Vorstellungen und seinen Stil. Daher beschränke ich mich auf einige kurze Hinweise.

3.8.1 Texturübergänge

Es gibt Stellen, an denen deutlich abgegrenzte *Texturen* aneinander stoßen. Das ist z.B. an den Stellen der Fall, an denen eine Brücke an das Ufer grenzt. Leider ist es nicht möglich, die gewünschte Textur bis an das Brückenende fortzusetzen. Um den abrupten Übergang zu kaschieren, kann man aber auf die Werkzeuge Free objects und Horizontal objects zurückgreifen. Hier findet man einige Bodentexturen, die zum Abdecken der Übergänge geeignet sind. Anders als die Terraintexturen und ähnliche Bodentexturen aus der Kategorie Roads überlagern sie die Brückenenden, so dass die Übergänge nicht mehr so stark auffallen (vgl. Abb. 26).



Abb. 26: Kaschieren von Texturübergängen an Brücken

Ähnlich kann man vorgehen, wenn man verschiedene Straßenarten ineinander übergehen lassen will. Wenn man die „auslaufenden“ Straßenobjekte benutzt, kann man den Übergang durch Überlagerung verschiedener Texturobjekte einigermaßen harmonisch gestalten.

3.8.2 Blockieren von Objekten

In den Kategorien Free objects und Craters gibt es einige große Felsen und Krater, bei denen es etwas merkwürdig aussieht, wenn ein Fahrzeug sie ohne weiteres überquert. Insbesondere bei dem Krater in Abb. 27 sieht es aus, als ob die Einheiten darüber hinweg schweben. Um diesen Eindruck zu vermeiden, kann man die Objekte mit Hilfe manuell gesetzter Blockierungsflags (vgl. 3.6.2) unbefahrbar machen.



Abb. 27: „Schwebender“ Panzer

3.8.3 Darstellung von Feldern

Wenn die Mission nicht gerade schwerpunktmäßig in einer Stadt oder in einem Waldgebiet spielt, kommt man nicht umhin, Felder zu gestalten. Verwendet man dazu nur die Terraintexturen Ground und Plough (vgl. 3.2), wird die Darstellung allerdings sehr eintönig. Glücklicherweise gibt es eine ganze Reihe von Möglichkeiten, Ackerflächen abwechslungsreich zu gestalten. Einige Beispiele sind in Abb. 28 zu sehen.

Mit dem Werkzeug Set brightness lassen sich Pflugfurchen nachbilden, indem man das Feld mit helleren und dunkleren Streifen versieht. Ebenfalls für die Andeutung von Pflugfurchen eignen sich die gerade verlaufenden Reifenspuren aus der Werkzeugkategorie Roads. Eine weitere Möglichkeit ist das streifenweise Aufbringen der Terraintextur Dry ground, was wegen der automatischen Erzeugung eines dunkleren Randes ebenfalls zu einem Muster von hellen und dunklen Streifen führt.

Mit den grünen Bodentexturen aus der Kategorie Horizontal objects lässt sich beginnender Bewuchs des Feldes andeuten. Stellt man in der Kategorie Trees den Damage-Regler in die rechte Position, sieht man die Darstellung der zerstörten Bäume, die auch für die Darstellung von Pflanzenwuchs verwendet werden können.

Ähnlich kann man die Kistenstapel aus der Kategorie Vertical objects im zerstörten Zustand benutzen. Man erhält so bewachsene Furchen. Um eine allzu regelmäßige Struktur zu vermeiden, sollte man darauf achten, in unregelmäßigen Abständen unterschiedlich große Lücken zu lassen. Abwechselnde Streifen der Terraintexturen Grass und Ground, Sand oder Plough führen zu einem ähnlichen Ergebnis.

Stark bewachsene Felder kann man mit den Büschen aus der Kategorie Trees [5] oder mit der Fanpflanze aus der Kategorie Vertical objects darstellen.

Die hier aufgeführten Verfahren lassen sich auch kombinieren, so dass noch reichlich Spielraum für Experimente vorhanden ist.

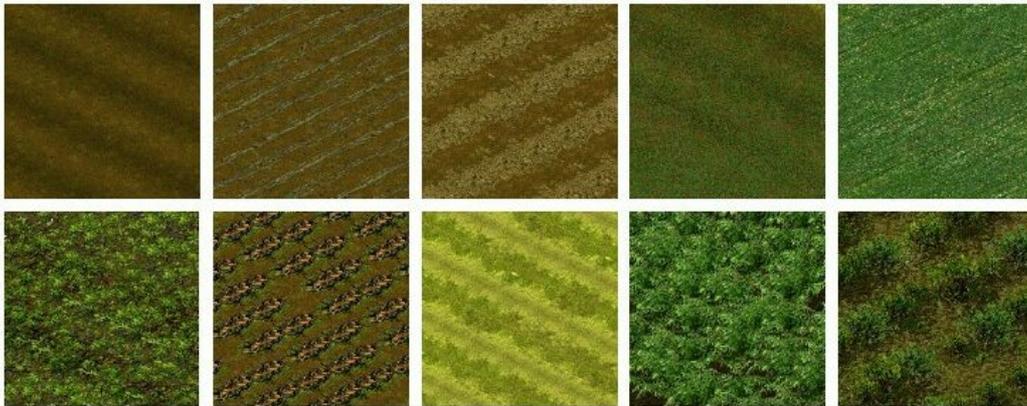


Abb. 28: Beispiele für die Gestaltung von Feldern

4 Missionen

Bei der Bearbeitung einer *Mission* ergeben sich Änderungen bei den Bestandteilen des Editorfensters. Es kommen einige neue Kategorien im Werkzeugbaum hinzu. Außerdem wird die Auswahlleiste je nach der ausgewählten Kategorie durch ein Hilfsfenster ersetzt, in dem zusätzliche Informationen zu Orten, Einheiten usw. angezeigt werden.

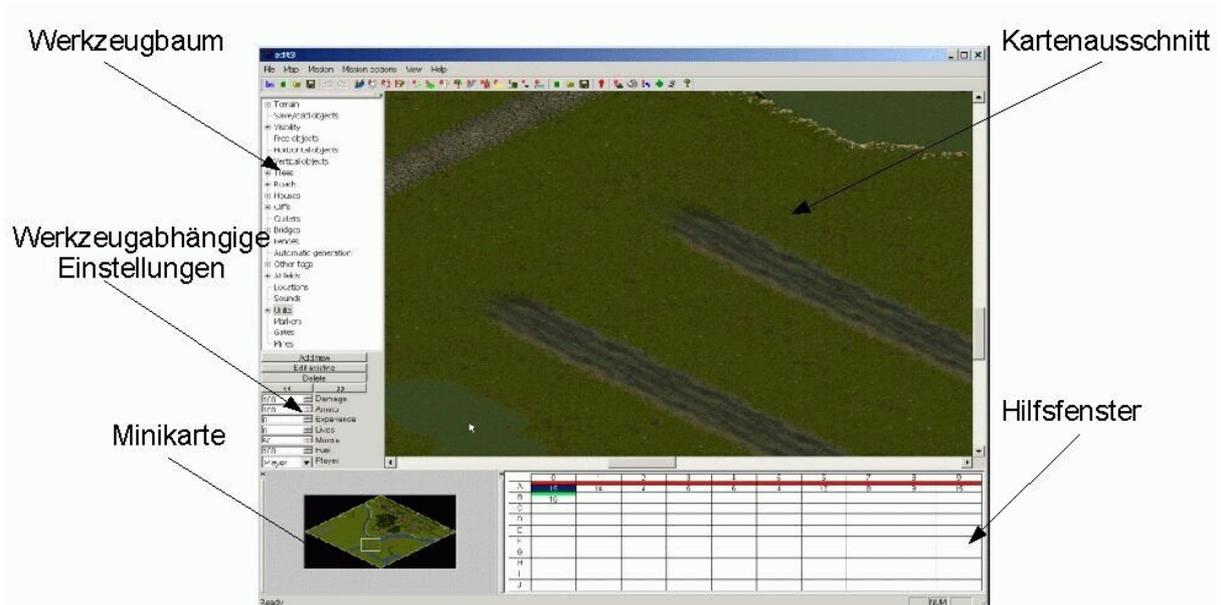


Abb. 29: Das Editorfenster bei der Arbeit an einer Mission (hier: Bearbeitung von Einheiten)

4.1 Dateioperationen mit Missionen

Missionen existieren grundsätzlich nur in Verbindung mit einer Karte, d.h. die nachfolgend beschriebenen Operationen sind nur durchführbar, wenn bereits eine Karte erstellt und/oder geladen wurde.

4.1.1 Anlegen einer neuen Mission

Eine neue Mission wird über *Mission > New mission* oder durch Klick auf das entsprechende Icon angelegt. In dem daraufhin erscheinenden Dialog legt man die Größe der Mission fest. Wie bei der Kartengröße sind prinzipiell die Größen 128x128, 256x256, 384x384 und 512x512 möglich. Die *Missionsgröße* darf allerdings höchstens die Kartengröße erreichen. Ist die *Missionsgröße* geringer als die Kartengröße, kann man die *Missionszone*, die von einem heller dargestellten Rahmen umgeben ist, bei gehaltener Umschalttaste mit den Cursortasten auf der Karte passend ausrichten.

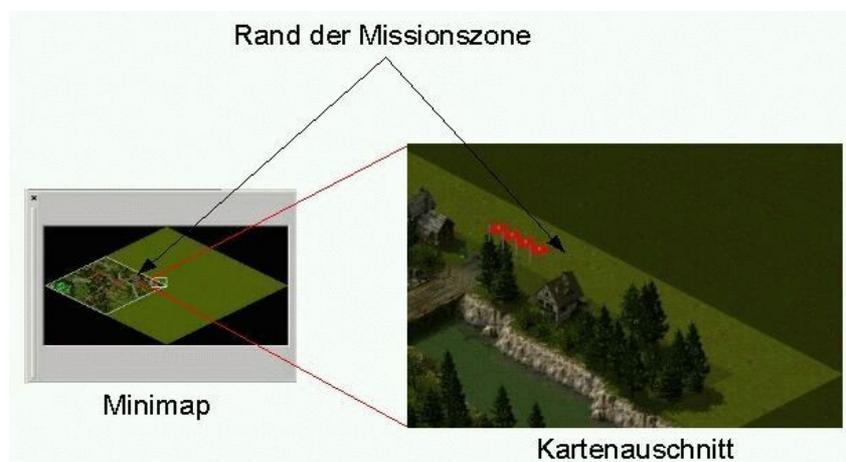


Abb. 30: Darstellung des Randes der Missionszone in Minimap und Kartenausschnitt

Soll eine Mission auf einer Karte spielen, die größer ist als die Missionszone, ist es sinnvoll, Karte und Mission parallel zu bearbeiten, damit man sich beim Bau der Karte an der eingeblendeten Grenze der Missionszone orientieren kann. Sonst passiert es leicht, dass man Objekte am Rand der Missionszone so plaziert, dass sie in der fertigen Mission nicht angezeigt werden (vgl. Abb. 11).

4.1.2 Speichern und Laden von Missionen

Die Funktionen zum Speichern und Laden von Missionen findet man unter Mission > Save mission bzw. Mission > Load mission. Es existieren auch entsprechende Icons in der Iconleiste. Das Vorgehen entspricht dem beim Speichern und Laden von Karten. Jede Karte hat 16 Slots zum Speichern von Missionen, d.h. auf einer Karte können bis zu 16 Missionen angelegt werden. Man kann eine Mission erst speichern, nachdem die Karte, auf der sie basiert, bereits gespeichert ist. Entsprechend kann man zu einer Karte auch nur die darauf basierenden Missionen laden.

4.1.3 Zurücksetzen einer Mission

Soll die bearbeitete Mission von Grund auf neu begonnen werden, kann man Mission > Clear mission auswählen. Dadurch werden sämtliche Missionseinstellungen gelöscht. Dies ist nur selten notwendig und auch mit Vorsicht zu genießen, denn der Editor scheint dabei relativ oft abzustürzen.

4.1.4 Erzeugen einer Bitmap der Mission

Über Mission > Generate bitmap kann man eine Bilddatei der Missionszone erzeugen. Das Vorgehen entspricht dem unter 3.1.3 beschriebenen, es wird jedoch nur die Missionszone der aktuellen Mission berücksichtigt. Ist die Karte also größer als die Missionszone, dann wird nur der für die Mission definierte Kartenausschnitt in die Bilddatei übernommen.

4.2 Grundlegende Missionseinstellungen

4.2.1 Spielereinstellungen

Die Einstellungen im Dialog Players, den man über Mission options > Players öffnet, sind vor allem für Einzelspielermissionen von Bedeutung. Für Informationen über Mehrspielermissionen sei auf die Originaldokumentation oder auf Irrläufers Editor-Fibel verwiesen.



Abb. 31: Spielereinstellungen

Die Einstellung Name braucht nicht verändert zu werden, da Änderungen sich nicht auswirken. Auch die Einstellung Color sollte man unverändert lassen, damit Spieler nicht durch vom Standard abweichende Farben auf der taktischen Karte verwirrt werden. Die Nation des Spielers sollte man der Mission entsprechend einstellen, da sie Auswirkungen auf die Anzeige des Missionsergebnisses hat. Die *Teameinstellungen* legen fest, welche Teams zusammenarbeiten. Normalerweise spielen Team 1 (Spieler) und Team 3 (computergesteuerte Verbündete des Spielers) bei Einzelspielermissionen gegen Team 2 (Gegner), während Team 4 neutral ist (z.B. Zivilisten). Wenn der Spieler alles sehen soll, was auch seine Verbündeten sehen, muss man für Spieler und *Verbündete* das gleiche Team einstellen.

Sollen dem bearbeiteten Team bereits bei Beginn der Mission *Flugzeuge* zur Verfügung stehen, kann man die entsprechenden Einstellungen im Bereich Planes vornehmen. In dem Auswahlmenü wird festgelegt, für welche Art von Flugzeugen die weiteren Einstellungen gelten sollen. Mit Hilfe der acht rautenförmig angeordneten Buttons legt man die Richtung fest, aus der die Flugzeuge des Teams die Karte erreichen sollen. In Abb. 31 wurde z.B. als Richtung Südost ausgewählt.

Nach Klick auf Options kann man das passende Flugzeugmodell (Plane type) auswählen und die Verzögerung (Delay) einstellen, mit der die Flugzeuge die Karte erreichen. Die maximale Anzahl gleichzeitig zur Verfügung stehender Flugzeuge dieser Art stellt man über Number of planes, die maximale Anzahl von Flügen dieser Flugzeugart über Number of flights ein.

Für Transportflugzeuge kann man nach Klick auf Paradrop noch einstellen, welche Ladung sie transportieren und abwerfen sollen. Da es zwei Arten von Transportflugzeugen (Transports, Cargo planes) gibt, kann man zwei verschieden zusammengesetzte Ladungen erstellen. Diese Einstellungen werden in einer Variante des *Einheiteneditors* (vgl. 4.3) vorgenommen, die gegenüber der normalen Version weniger Einstellmöglichkeiten bietet (vgl. Abb. 32). Eine Ladung besteht aus bis zu vier verschiedenen Einheitentypen. Um eine bestimmte Einheit in die Ladung aufzunehmen, wählt man diese im linken Teil des Editorfensters aus. Die Auswahl ist dabei auf die Einheiten eingeschränkt, die sich in einem Flugzeug transportieren lassen – alles, was sich im linken Teil auswählen lässt, kann auch in die Ladung übernommen werden. Dies geschieht, indem man auf einen der vier großen Buttons im rechten Teil des Editors klickt. Der Button zeigt dann das Bild der ausgewählten Einheit an. Die Anzahl der Einheiten dieser Art trägt man in dem Eingabefeld rechts neben dem Button ein. Abschließend kann man noch die Gruppe und die Erfahrung der Einheiten festlegen.

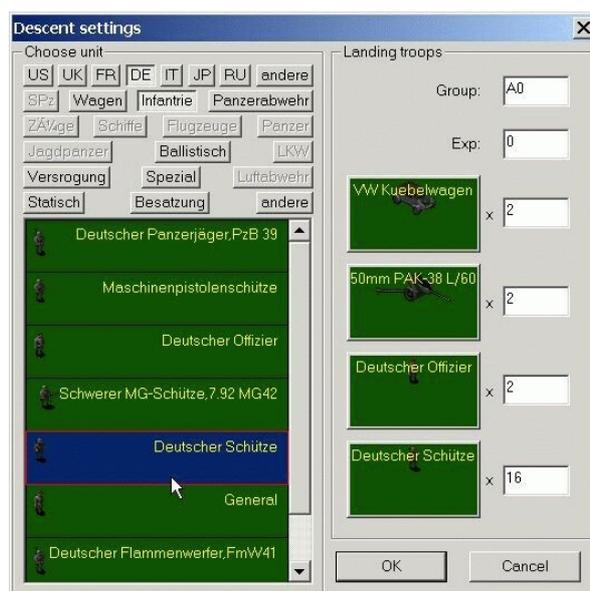


Abb. 32: Festlegung der Ladung eines Transportflugzeugs (in diesem Fall 2 Kübelwagen, 2 PAKs, 2 Offiziere und 16 Schützen)

4.2.2 Missionsziele

Über Mission options > Objectives öffnet man ein Fenster, in dem man eine Beschreibung der *Missionsziele* eingeben kann. Diese Beschreibung wird sowohl zu Beginn der Mission im *Briefing* angezeigt als auch während des Spiels, wenn man im Spielmenü Ziele auswählt.

Wenn man seine Mission mit einem animierten Briefing versehen hat, kann man im *Briefing-Editor*¹⁶ die Animation mit bis zu drei *Teilungspunkten* versehen (vgl. 8.5.3). Dadurch entstehen bis zu vier Teilanimationen, die jeweils einen Missionsabschnitt darstellen. Nach einem Klick auf einen der Buttons unten links im Briefing-Bildschirm (vgl. Abb. 33) werden diese Teilanimationen angezeigt.

¹⁶ Der Briefing-Editor ist in der SSRW-Installation nicht enthalten. Man kann aber den Briefing-Editor von SuS-II ohne weiteres auch für SSRW verwenden.

Damit zu jeder Teilanimation auch ein passender Text eingeblendet wird, fügt man am Ende des Textes mit den Missionszielen die Befehle für die einzelnen Missionsabschnitte an. Soll z.B. für das erste Missionsziel der Text aus Abb. 33 angezeigt werden, dann gibt man im Editor bei den Missionszielen den Text ~1. Durchbrechen Sie die Grenzbefestigungen. ein. Es reicht also aus, vor dem eigentlichen Text für das Missionsziel eine Tilde (~) einzugeben¹⁷.

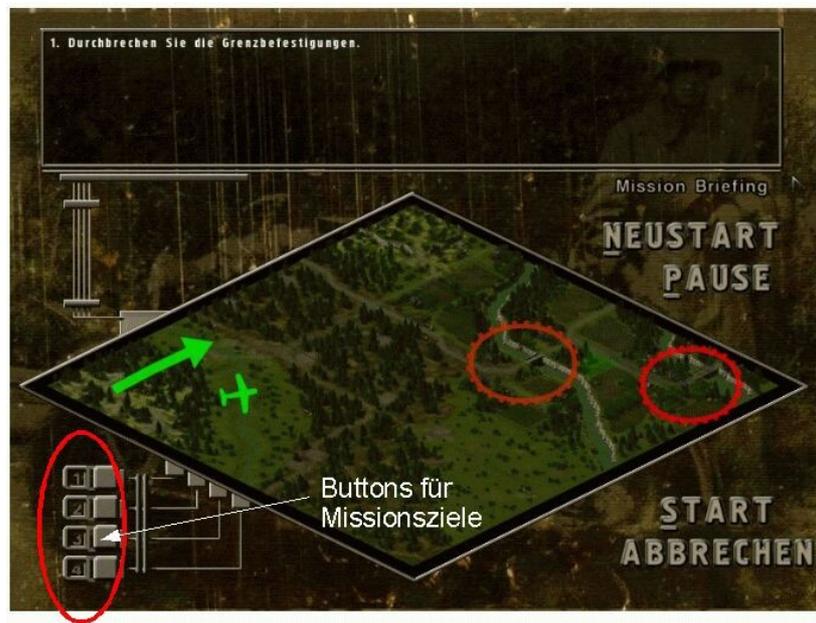


Abb. 33: Briefinganimation mit vier Teilanimationen

4.2.3 Taktische Karte

Wenn der Menüpunkt Mission options > Generate tactical map angeklickt und mit einem Häkchen versehen wird, dann wird bei der Erzeugung der spielbaren Mission auch eine *taktische Karte* eingebunden, die im Spiel durch Drücken der Feststelltaste¹⁸ eingeblendet werden kann.

4.2.4 Missionseinstellungen

Über Mission options > Settings öffnet man das Fenster mit den *Missionseinstellungen*, das in der folgenden Abbildung zu sehen ist.



Abb. 34: Missionseinstellungen

¹⁷ Danke an Slyder für diese Information!

¹⁸ Bei SuStII mit der Taste **M**

In den beiden ersten Eingabefeldern können Namen für die Mission eingegeben werden. Der erste Name (Short mission name) wird nur benötigt, wenn die Mission mehrsprachig sein soll¹⁹. Der zweite Name (Long mission name) wird in der Liste der Einzel- bzw. Mehrspielermissionen angezeigt. Für Kampagnenmissionen ist dieser Name nicht notwendig, da für Kampagnenmissionen keine Auswahlliste existiert.

Unter Weather kann man den *Wettereffekt* einstellen, der während der Mission zu sehen ist und die Sichtweite der Einheiten beeinflusst. Die Optionen für Regen (Rain) und Nebel (Fog) sind vertauscht. Will man Regen als Wettereffekt einsetzen, muss man Nebel auswählen und umgekehrt. Die übrigen Einstellungen in diesem Dialog werden nur für mehrsprachige Missionen (Language file name, Use lng file) oder Mehrspielermissionen (Multiplayer options) benötigt.

4.3 Der Einheiteneditor

Der *Einheiteneditor* wird sowohl beim Einfügen neuer Einheiten als auch beim Editieren von Verstärkungen verwendet. Er bietet die Möglichkeit, die neuen Einheiten an die Erfordernisse der Mission anzupassen, z.B. durch Begrenzung der Munitionsmenge oder die Zusammenstellung der Besatzung oder Ladung.

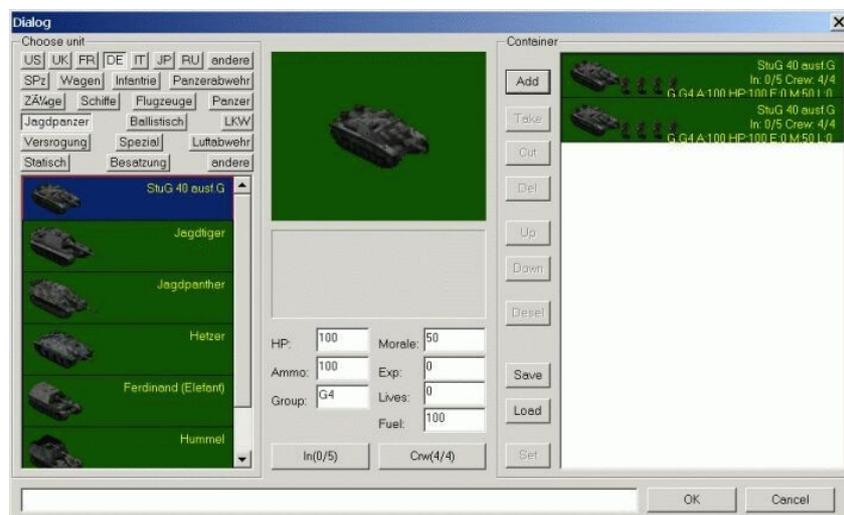


Abb. 35: Der Einheiteneditor

Im linken Teil des Einheiteneditors wird die neue Einheit ausgewählt. Die erste Reihe mit Buttons dient zur Auswahl der Nation, zu der die Einheit gehört. Die übrigen Buttons schränken die Auswahl auf bestimmte Klassen ein. Es ist möglich, mehrere Buttons zu aktivieren, wenn man bei der Auswahl die Umschalttaste gedrückt hält. Die so eingeschränkte Auswahl wird in der Liste unterhalb der Buttons angezeigt. Durch Anklicken eines Eintrags wird die Einheit im mittleren Teil des Editorfensters angezeigt. Sie wird als aktuelle Einheit bezeichnet.

Unterhalb des großen Bildes der aktuellen Einheit kann man Werte (in Prozent) für Schaden (HP), Munition (Ammo), Moral (Morale), Erfahrung (Exp) und Treibstoff (Fuel²⁰) für die Einheit einstellen, falls die Standardwerte nicht passend erscheinen. Dabei steht 0 für 0%, 100 für 100% des Maximalwertes (bei Moral und Erfahrung 1000, bei den anderen Werten ist das Maximum abhängig von der Einheit). Unter Schaden wird hier übrigens nicht der schon erlittene Schaden verstanden, sondern der Schaden, den die Einheit noch nehmen kann. Das bedeutet, ein Wert von 100 steht für voll einsatzfähig, ein Wert von 0 für tot bzw. zerstört. Neben den erwähnten Werten kann auch die Gruppe (Group), zu der die aktuelle Einheit gehören soll, noch geändert werden.

¹⁹ Da ich nicht vorhabe, mehrsprachige Missionen zu erstellen, wird dieses Thema in diesem User Guide nicht weiter ausgeführt.

²⁰ nur bei SSRW

Die Einstellung Leben (Lives) ermöglicht es, der aktuellen Einheit zusätzliche Leben zuzuweisen (Maximalwert ist 255). Wird eine solche Einheit zerstört, kann sie aus einem speziellen Zwischenspeicher (Hell, also Hölle) wiederhergestellt werden. Die Vorgehensweise wird in 5.1.4 und 7.5.8 genauer beschrieben.

Die Buttons In und Crw ermöglichen die Bearbeitung der *Passagiere* bzw. der *Besatzung* der aktuellen Einheit. Wird einer dieser Buttons angeklickt, dann wird er aktiviert dargestellt und im rechten Teil des Editorfensters erscheinen die Slots, in die Besatzungsmitglieder bzw. Passagiere oder Ladung eingefügt werden können. Die Auswahlmöglichkeiten im linken Teil des Einheiteneditors werden entsprechend eingeschränkt. Die jeweils in der Mitte angezeigte Einheit kann mit Add in einen freien Slot der aktuellen Einheit übertragen werden.

Sind bereits Slots der aktuellen Einheit belegt, können diese im rechten Teil des Einheiteneditors markiert werden. Durch Halten der Umschalt- oder der Strg-Taste ist auch Mehrfachauswahl möglich. Sind Einheiten markiert, werden weitere Buttons aktiv, die zur Bearbeitung der Slots dienen. Take überträgt die markierte Einheit in den mittleren Teil des Einheiteneditors, wo ihre Einstellungen weiter bearbeitet werden können. Del entfernt die markierte Einheit aus dem Slot. Cut bewirkt das gleiche wie die Nacheinanderanwendung von Take und Del. Desel (Abkürzung für deselect) hebt die Markierung auf. Die Reihenfolge der Einheiten in den Slots kann mit den Buttons Up und Down verändert werden. Die markierte Einheit wird damit um einen Slot nach oben bzw. unten verschoben.

Wird eine Einheit in einem Slot markiert, dann lassen sich die Einstellungen dieser Einheit in der Mitte des Editorfensters verändern. Die Änderungen werden übernommen, wenn man den Button Set anklickt.

Ist die Bearbeitung von Besatzung oder Inhalt abgeschlossen, kann man die entsprechenden Einheiten mit ihren Einstellungen mit Save abspeichern und später mit Load wieder laden. Dadurch wird die Bearbeitung von Einheiten und ihren Besatzungen bzw. Passagieren auf die Dauer erheblich vereinfacht. Es bietet sich z.B. an, typische Ladungen wie Treibstoff, Munition, Granatwerfer und MGs sowie typische Infanteriezusammenstellungen vorzubereiten und für spätere Wiederverwendung zu speichern.

Manchmal kommt es vor, dass durch einen Bug im Editor eine Einheit mehrfach gezählt wird, was daran zu erkennen ist, dass die Einheit auf der rechten Seite des Einheiteneditors in einem Slot mehrfach abgebildet wird. Das sieht dann in etwa so aus, wie in der (nachgestellten) Abb. 36. In solchen Fällen hilft es, die Mission und die Karte zu speichern und den Editor neu zu starten.



Abb. 36: Bug im Einheiteneditor

4.4 Der Skripteditor

Der *Skripteditor* besteht aus zwei Fenstern, von denen das eine die Verwaltung der Skripte (auch als *Trigger* bezeichnet), das andere die eigentliche Programmierung ermöglicht. An dieser Stelle wird nur die Benutzeroberfläche beschrieben. Die eigentliche Skriptprogrammierung wird in Kapitel 7 genauer erklärt.

4.4.1 Skriptverwaltung

Über Mission options > Scripts oder durch gleichzeitiges Drücken von **S** und Umschalttaste wird der Dialog zur Verwaltung der zur Mission gehörenden Skripte geöffnet, der nachfolgend abgebildet ist.

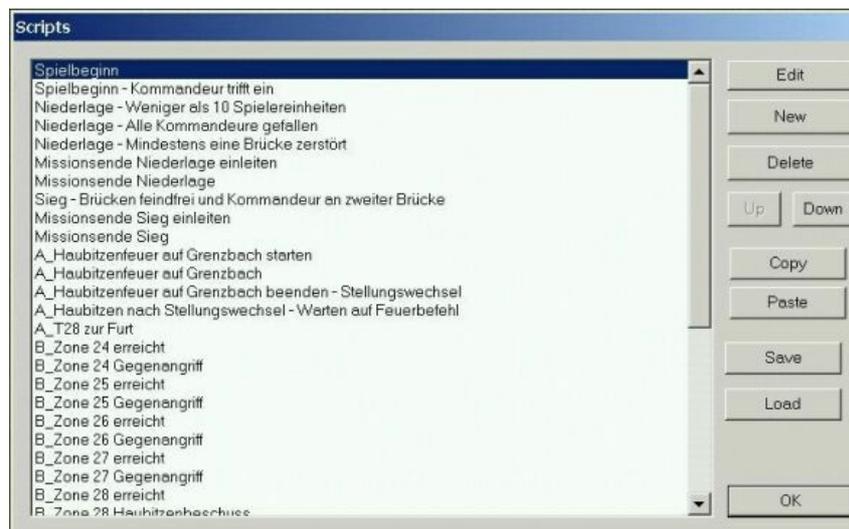


Abb. 37: Dialog zur Skriptverwaltung

In der Liste, die den überwiegenden Teil des Dialogfensters einnimmt, kann man eines der bereits vorhandenen Skripte auswählen, das dann nach Klick auf **Edit** bearbeitet werden kann. Durch Klick auf **New** wird ein neues Skript angelegt und zur Bearbeitung geöffnet. Die Bearbeitung findet in einem weiteren Fenster statt, das im nachfolgenden Abschnitt beschrieben wird.

Müssen mehrere ähnliche Skripte erstellt werden, kann man eines dieser Skripte erstellen und als Grundlage für die übrigen nehmen. Dazu markiert man das zuerst als Grundlage für die übrigen erstellte Skript und klickt auf den Button **Copy**. Dadurch wird eine Kopie des Skripts erzeugt, die mit Klick auf **Paste** nach dem markierten Skript in die Liste eingefügt werden kann.

Um Skripte wiederverwenden zu können, ist es möglich Skripte zu speichern und wieder zu laden. Dazu dienen die Buttons **Save** und **Load**. Beim Speichern wird nur das Skript gespeichert, nicht jedoch die darin angesprochenen Texte (Phrases) und Zonen. Gruppenbezeichnungen bleiben erhalten, allerdings ohne die in der ursprünglichen Mission in den Gruppen zusammengefassten Einheiten. Kurz gesagt wird nur der missionsunabhängige Teil des Skripts gespeichert.

Das in der Liste markierte Skript kann auch gelöscht werden. Dazu ist nur ein Mausklick auf den Button **Delete** erforderlich.

Die Reihenfolge der Skripte in der Liste entspricht der Reihenfolge, in der die auslösenden Bedingungen während des Spiels überprüft werden. Dadurch kann in seltenen Fällen die Situation vorkommen, dass die gewünschten Effekte nicht eintreten, da die Reihenfolge der Skriptausführung dies nicht erlaubt. In solchen Fällen kann man mit den Buttons **Up** und **Down** die Reihenfolge der Skripte verändern, indem man das markierte Skript in der Liste auf oder ab bewegt.

4.4.2 Skriptprogrammierung

Wird ein *Skript* zur Bearbeitung geöffnet, so wird das nachfolgend abgebildete Dialogfenster angezeigt, das aus fünf Bereichen besteht.

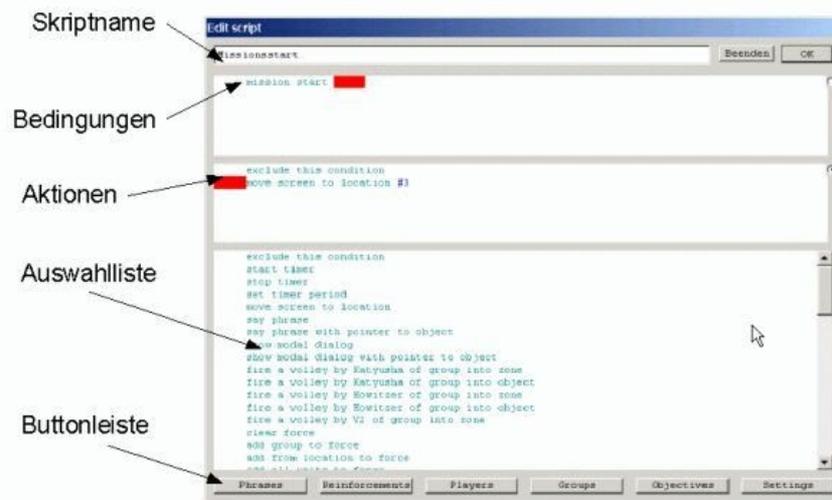


Abb. 38: Dialog zur Skriptbearbeitung

In dem Eingabefeld am oberen Rand des Fensters wird standardmäßig New trigger (d.h. neues Skript) eingeblendet. Diesen Text sollte man durch einen aussagekräftigen Namen ersetzen, der aus bis zu 63 Zeichen bestehen kann.

Im ersten großen Eingabefeld wird die *Bedingung* angegeben, durch die die Ausführung des Skripts ausgelöst wird. Die Bedingungen der Skripte werden während des Spiels mehrmals in der Sekunde überprüft. Ist die Bedingung eines Skripts erfüllt, dann werden die im Skript festgelegten Aktionen ausgeführt. Die möglichen Bedingungen stehen in dem Bereich, der in Abb. 38 als *Auswahlliste* bezeichnet ist.

Durch Mausklick auf die gewünschte Bedingung in der *Auswahlliste* wird sie in das Bedingungs Fenster übertragen. Bei einigen Bedingungen sind eine oder mehrere Stellen mit <<*>> markiert. An diesen Stellen sind weitere Informationen anzugeben, die die Bedingung genauer beschreiben, z.B. der Ort oder die Gruppe, auf die sich die Bedingung bezieht. Durch Mausklick auf diese Stellen öffnet sich ein Dialog, in dem die erforderlichen Informationen angegeben werden können.

Durch einen Rechtsklick auf eine Bedingung wird diese gelöscht. Durch einen Klick links oder rechts neben eine Bedingung wird diese Stelle aktiv (angedeutet durch ein rotes Rechteck). An dieser Stelle wird die nächste in der *Auswahlliste* angeklickte Bedingung eingefügt.

Soll die Ausführung eines Skripts nur dann erfolgen, wenn mehrere Bedingungen erfüllt sind, dann kann man die einzelnen Bedingungen mit den *Operatoren* and und or (vgl. 7.1) verknüpfen. Soll ein Skript dagegen ausgeführt werden, wenn eine Bedingung nicht erfüllt ist, dann setzt man den Operator not vor die Bedingung. Ggf. kann man auch Klammern setzen. Die Operatoren und Klammern sind in der *Auswahlliste* mit den möglichen Bedingungen zu finden. Der Operator not und die Klammer (stehen bei den normalen Bedingungen, während and, or und (falls eine Klammer geöffnet wurde) die Klammer) angezeigt werden, wenn man rechts neben die Bedingung klickt, auf die der Operator bzw. die Klammer folgen soll.

Die *Aktionen*, die bei erfüllter Bedingung ausgeführt werden sollen, werden in dem Bereich unterhalb der Bedingungen angegeben. Wechselt man mit einem Mausklick in diesen Bereich, so werden die möglichen Aktionen in der *Auswahlliste* angezeigt. Durch einen Klick auf den Anfang einer Zeile wird diese Zeile aktiv (angedeutet durch ein rotes Rechteck). An dieser Stelle wird die nächste in der *Auswahlliste* angeklickte Aktion eingefügt. Durch einen Rechtsklick auf eine Zeile wird die Aktion in dieser Zeile gelöscht.

In der *Buttonleiste* findet man eine Reihe von Buttons, die den Zugriff auf Elemente ermöglichen, die für die Skriptprogrammierung wichtig sind, z.B. Missionstexte, Verstärkungen usw. Dadurch kann man diese Elemente bearbeiten, ohne den Skripteditor verlassen zu müssen.

Die Bearbeitung des Skripts beendet man mit einem Mausklick auf OK. Bei einem Mausklick auf Beenden werden die Änderungen an dem Skript nicht übernommen.

Das Fenster für die Skriptverwaltung bleibt übrigens während der Skriptbearbeitung aktiv, so dass man ständig Zugriff auf die Liste aller Skripte hat. Durch Mausklick auf ein anderes Skript kann man direkt zu diesem Skript wechseln²¹.

4.5 Testen und Fertigstellen einer Einzelmission

Zum *Testen* der bearbeiteten Mission wählt man Mission > Run game. Die Mission wird dann als Einzelspielermission unter dem Namen `__test__.smm` erzeugt und in SSRW gestartet. Der Editor bleibt währenddessen geöffnet, so dass das System ausreichend Arbeitsspeicher besitzen sollte. Auf meinem System mit 256 MB gibt es bei diesem Vorgehen aber keine Probleme²², auch wenn noch mehrere andere Programme zusätzlich geöffnet sind.

Ist die Mission fertiggestellt, kann man sie über Mission > Build as single mission als *Einzelspielermission* kompilieren. In dem Dialog Build single mission kann man angeben, unter welchem Namen und in welchem Ordner das Ergebnis gespeichert werden soll. Die entstehende Datei vom Typ `*.ssm` muss – falls man sie nicht sowieso dort erzeugt hat – anschließend noch in das Verzeichnis `C:\Programme\Sudden Strike - Resource War\Run\maps\SINGLE` kopiert werden, damit sie im Spiel in der Liste der Einzelspielermissionen erscheint und ausgewählt werden kann.

4.6 Erzeugung von Kampagnen

Im Vergleich zu Einzelmissionen ist die Erzeugung von *Kampagnen* etwas komplizierter. Die Missionen der Kampagne können zwar wie gewohnt erstellt werden, allerdings sind einige zusätzliche Skriptbefehle erforderlich. Die als *Kampagnenmissionen* erstellten Missionen müssen dann noch mit dem Programm `Sue.exe` zu der Kampagnendatei zusammengefasst werden. Die erforderlichen Arbeitsschritte werden in den folgenden Abschnitten beschrieben²³.

4.6.1 Skripte

Wie in jeder Mission werden auch bei den Missionen einer *Kampagne* Skripte benötigt, die Sieg und Niederlage regeln. In diesen Skripten muss jeweils angegeben werden, welche Mission auf die gerade abgeschlossene Mission folgen soll. Dazu verwendet man den Skriptbefehl `set next mission`, der in 7.8.2 ausführlich beschrieben wird.

4.6.2 Erzeugen einer Kampagnenmission

Die Mission wird nach Fertigstellung mittels Mission > Build as campaign mission als *Kampagnenmission* erzeugt. Anders als bei der Erzeugung von Einzelmissionen kann man weder Pfad noch Dateinamen angeben. Der Dateiname ergibt sich aus der Nummer der Karte und der Nummer der Mission. Die erzeugte Datei vom Typ `*.ssc` wird im Verzeichnis `C:\Programme\Sudden Strike - Resource War\Run\maps\CAMPAIGN` abgelegt. Zusätzlich wird eine weitere Datei vom Typ `*.ssc` erzeugt, deren Name nur aus der Nummer der Karte besteht. Diese Datei wird aktualisiert, wenn weitere Kampagnenmissionen auf der gleichen Karte spielen.

Ob dabei Dateien in diesem Verzeichnis überschrieben werden, wird leider nicht geprüft. Daher sollte man den Inhalt des Verzeichnisses unbedingt vor dem Erzeugen der Missionen sichern.

4.6.3 Erzeugen der Kampagnendatei

Um die spielfertige Kampagne zu erzeugen, sind zwei Schritte notwendig. Zunächst legt man mit einem Texteditor, z.B. dem mit Windows mitgelieferten, eine Textdatei mit dem Namen `info.txt` an. Da im Verzeichnis `C:\Programme\Sudden Strike - Resource War\Run\maps\CAMPAIGN` bereits eine gleichnamige Datei existiert, ist es sinnvoll, wenn man die neue Datei in einem anderen Verzeichnis speichert. Man könnte z.B. im Verzeichnis `C:\Programme\Sudden Strike - Resource`

²¹ Bei SuStII gibt es diese Möglichkeit leider nicht.

²² Weniger als 256 MB Arbeitsspeicher hat heutzutage wohl kaum noch jemand. Ich bin in meinem Umfeld jedenfalls meist derjenige, der seinen PC als Letzter aufrüstet. ☺

²³ Für die Erzeugung von Kampagnen in SuStII bietet es sich an, Slyders Kampagnentutorial zu Rate zu ziehen, da sich die Vorgehensweise bei SuStII in Details von der hier beschriebenen unterscheidet.

War\Run\maps ein Verzeichnis MY_CAMPAIGN anlegen²⁴. Dorthin kopiert man dann auch die Kampagnendateien vom Typ *.ssc.

In der ersten Zeile der Textdatei gibt man den Namen der Kampagne an, der frei wählbar ist. In der zweiten Zeile legt man fest, mit welcher Mission die Kampagne beginnen soll. Dazu gibt man die Nummer der Karte und, getrennt durch ein Leerzeichen, die Nummer der Mission an. Ein Beispiel sieht man in Abb. 39.

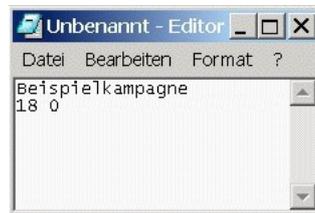


Abb. 39: Beispiel für die Datei info.txt

Nun sind alle benötigten Dateien vorhanden, so dass die spielbare Kampagne erzeugt werden kann. Dazu braucht man die passende Version des Programms sue.exe, die z.B. bei dem Kampagnentutorial von Slyder mitgeliefert wird. Die Programmdatei sue.exe kopiert man ins Verzeichnis C:\Programme\Sudden Strike - Resource War\Run\maps.

Anschließend wird die Eingabeaufforderung von Windows geöffnet, z.B. indem man Start > Ausführen... wählt und dann als zu öffnendes Programm cmd eingibt²⁵. Im Fenster der Eingabeaufforderung wird dann der aktuelle Pfad angezeigt (links von dem Zeichen >). Beginnt der angezeigte Pfad nicht mit C:, wechselt man mit c: auf das Systemlaufwerk²⁶. Hier muss man nun zunächst zu dem Verzeichnis wechseln, in dem die Datei sue.exe liegt. Dazu gibt es den Befehl cd (kurz für change directory, also Verzeichnis wechseln). Ist der angezeigte Pfad länger als C:, dann springt man durch Eingabe von cd\ in das Hauptverzeichnis von C:. Anschließend gibt man cd programme\sudden strike - resource war\run\maps ein. Als Pfad wird jetzt das gesuchte Verzeichnis angezeigt.

Um die spielbare Kampagne zu erzeugen, gibt man dann sue NAME.ssx my_campaign\ ein. Statt NAME gibt man natürlich den gewünschten Namen der Kampagnendatei an. Der gesamte Vorgang ist in Abb. 40 zu sehen.

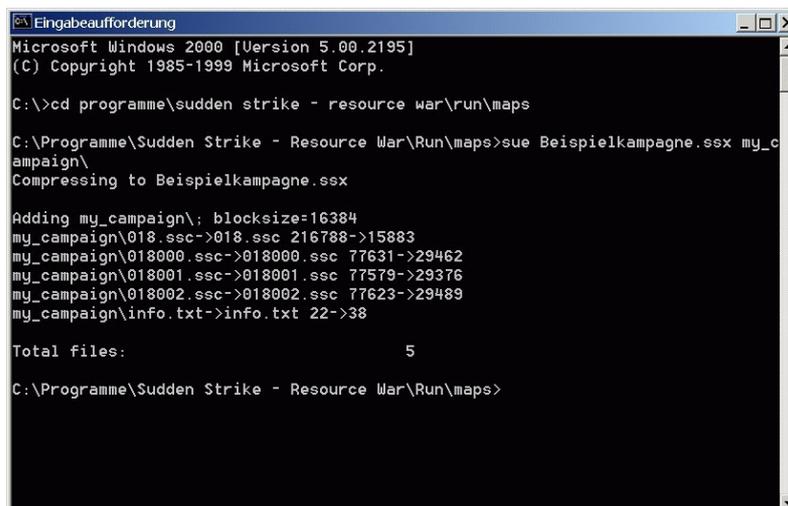


Abb. 40: Erzeugung der Kampagnendatei

Als Ergebnis findet man die Datei NAME.ssx im Verzeichnis C:\Programme\Sudden Strike - Resource War\Run\maps. Dabei handelt es sich um die spielbare Kampagnendatei, die nur noch in

²⁴ Die hier vorgeschlagene Vorgehensweise wird in den weiteren Ausführungen vorausgesetzt.

²⁵ Alternativ: Start > Programme > Zubehör > Eingabeaufforderung

²⁶ Ich gehe an dieser Stelle davon aus, dass SSRW auf dem Systemlaufwerk C: installiert wurde. Sollte das nicht der Fall sein, sind die weiteren Anweisungen entsprechend der Situation zu modifizieren.

das Verzeichnis C:\Programme\Sudden Strike - Resource War\Run\addons kopiert werden muss. Ist der Ordner addons nicht vorhanden, dann muss man ihn anlegen. Danach kann man die Kampagne spielen, indem man im Menü von SSRW nacheinander Einzelspieler > Kampagnen > Eigene und dann die neue Kampagne auswählt (vgl. Abb. 41).



Abb. 41: Auswahl der erstellten Kampagne im Spiel

5 Skriptgrundlagen

In diesem Kapitel geht es um die Vorbereitung der Skriptprogrammierung. Dazu werden die Elemente der Mission beschrieben, auf die bei der Programmierung zurückgegriffen wird. Um sich die Arbeit zu erleichtern, kann man über diese Elemente in Tabellen Buch führen. Auf diese Weise hat man bei der Arbeit an Skripten immer alle Informationen zur Hand. Entsprechende Tabellen findet man als Kopier- vorlagen am Ende dieses User Guides.

5.1 Einheiten

Da bei der Skriptprogrammierung auf *Gruppen* von *Einheiten* Bezug genommen wird, ist es oft von großer Bedeutung, zu welcher Gruppe man neue Einheiten hinzufügt. Idealerweise sollte man bei Beginn der Arbeit mit Einheiten bereits eine recht genaue Vorstellung haben, welche Einheiten benötigt werden und welche Aufgaben sie im Rahmen der Mission erfüllen sollen.

Zum Bearbeiten der Einheiten wählt man im Werkzeugbaum die Kategorie Units aus. Im Hilfsfenster erscheint eine tabellarische Übersicht über die Gruppen (vgl. Abb. 42). Es stehen 100 Gruppen zur Verfügung (A0 bis A9, B0 bis B9, ..., J0 bis J9). Die aktuell ausgewählte Gruppe wird blau hinterlegt dargestellt. Die Zahl gibt an, wie viele Einheiten zu der Gruppe gehören. Der farbige Balken deutet auf das Team, zu dem die Einheiten der Gruppe gehören. Wenn man in den Spielereinstellungen (vgl. 4.2.1) nichts geändert hat, steht grün für Einheiten des Spielers, gelb für seine Verbündeten, rot für den Gegner und blau für neutrale Einheiten. Eine Gruppe kann auch Einheiten verschiedener Teams enthalten. Der farbige Balken wird dann entsprechend mehrfarbig dargestellt.

5.1.1 Hinzufügen neuer Einheiten

Bevor eine neue Einheit zu der Mission hinzugefügt wird, sollte – falls dies erforderlich ist²⁷ – die Gruppe und der Spieler festgelegt werden, zu denen die neue Einheit gehören soll. Die Gruppe kann im Hilfsfenster durch Linksklick auf eine der Gruppen festgelegt werden. Alle weiteren neuen Einheiten werden dann dieser Gruppe hinzugefügt. Die Auswahl des Spielers ist über das Auswahlm Menü Player unter dem Werkzeugbaum möglich.

	0	1	2	3	4	5	6	7	8	9
A		2								
B	15	15		94	14	40	15			
C										
D										
E										
F										
G	38		2	2				18		
H	6	2	1							
I	12	5	8	10	6					
J	22	8	6	8	4					

Abb. 42: Übersicht über die Gruppen (Gruppe A1 ist ausgewählt, platzierte Einheiten werden dieser Gruppe hinzugefügt)

Danach kann durch Klick auf Add new der Einheiteneditor (vgl. 4.3) geöffnet werden. Sind im *Einheiteneditor* alle erforderlichen Einstellungen vorgenommen worden, kann man durch Klick auf OK den Editor schließen und die neue Einheit durch Linksklick in einem oder mehreren Exemplaren auf der Karte platzieren. Mit Hilfe der Buttons << und >> oder mit den Tasten und kann noch die Ausrichtung der neuen Einheit verändert werden, bevor sie platziert wird. Ebenso können über die Eingabefelder unter dem Werkzeugbaum die Einstellungen Schaden (Damage), Munition (Ammo), Erfahrung (Experience), zusätzliche Leben (Lives), Moral (Morale) und Treibstoff (Fuel) für die Einheit eingestellt werden. Durch Rechtsklick auf gesetzte Einheiten können diese entfernt werden.

Neue Einheiten können auch direkt aus den Unterkategorien der Kategorie Units im Werkzeugbaum ausgewählt und – ggf. nach vorgenommenen Einstellungen – in der Missionszone platziert werden.

²⁷ Als Faustregel sollte man Einheiten, die in der Mission für eine bestimmte Aufgabe vorgesehen sind, in einer eigenen Gruppe zusammenfassen. Probleme können sich aus gemischten Gruppen ergeben, in denen Einheiten mit unterschiedlichen Verhaltensmöglichkeiten vorhanden sind, z.B. Infanterie und Panzer. In solchen Fällen sollte man mehrere Gruppen anlegen.

Manchmal kommt es vor, dass durch einen Bug im Editor eine platzierte Einheit mehrfach gezählt wird, was bei Beobachtung der Einheitenanzahl im Gruppenfenster leicht feststellbar ist – beim Platzierten einer Einheit erhöht sich die Anzahl der Einheiten der Gruppe stärker als erwartet (bei einer einzelnen Einheit z.B. oft um 3 statt um 1). In solchen Fällen hilft es, die Mission und die Karte zu speichern und den Editor neu zu starten.

5.1.2 Bearbeiten vorhandener Einheiten

Die bereits in der Missionszone platzierten Einheiten können weiter bearbeitet werden, wenn man auf den Button Edit existing klickt oder die Leertaste drückt. Man kann dann einzelne Einheiten durch Linksklick auswählen. Auch Mehrfachauswahl ist möglich, wenn man während der Auswahl die Umschalttaste gedrückt hält. Alternativ kann man mit gehaltener linker Maustaste einen rechteckigen Rahmen aufziehen. Lässt man die Maustaste los, werden alle Einheiten innerhalb des Rahmens in die Auswahl übernommen. Sollen alle Einheiten einer Gruppe ausgewählt werden, klickt man bei gehaltener Alt-Taste im Hilfsfenster auf die gewünschte Gruppe.

Die ausgewählten Einheiten werden durch Anzeige der Gruppe gekennzeichnet, zu der sie gehören. Über die Bedienungselemente unter dem Werkzeugbaum lassen sich die Einstellungen Schaden (Damage), Munition (Ammo), Erfahrung (Experience), zusätzliche Leben (Lives), Moral (Morale) und Treibstoff (Fuel) für die ausgewählten Einheiten einstellen. Auch die Ausrichtung der Einheiten kann verändert werden.

Die Zuordnung der ausgewählten Einheiten zu einer Gruppe kann ebenfalls noch modifiziert werden. Um die ausgewählten Einheiten in eine andere Gruppe zu übertragen, klickt man auf die entsprechende Gruppe im Hilfsfenster. Wird dabei die Umschalttaste gedrückt, werden nur die Passagiere der ausgewählten Einheiten in die neue Gruppe übertragen.

5.1.3 Verhalten von Einheiten

Alle Einheiten, die nicht der Kontrolle durch den Spieler unterliegen, folgen in ihrem Verhalten einem sogenannten *Verhaltensmodell*, das festlegt, wie die Einheiten im Spiel handeln. Um das Verhalten einer Gruppe von Einheiten bearbeiten zu können, öffnet man durch einen Rechtsklick auf die Gruppe im Hilfsfenster das Fenster mit den Gruppeneigenschaften (vgl. Abb. 43). Neben der Bezeichnung der Gruppe (Group) und der Anzahl der Einheiten (unter Troops) wird im Rahmen Behavior das eingestellte KI-Verhaltensmodell²⁸ der Einheiten angezeigt. Eine ausführliche Erläuterung der verschiedenen Einstellungen und Optionen findet man in Kapitel 6.

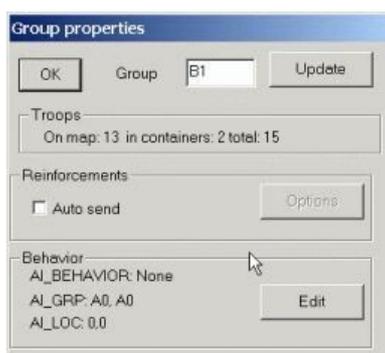


Abb. 43: Fenster mit den Gruppeneigenschaften

Der Button Edit öffnet ein weiteres Fenster, in dem das Verhaltensmodell verändert werden kann (vgl. Abb. 44). In den Feldern AI_GRP1 und AI_GRP2 können zwei Gruppen, die sog. *KI-Gruppen*, angegeben werden, auf die sich das Verhalten der bearbeiteten Gruppe beziehen soll. Standardwert ist in beiden Fällen die Gruppe A0. Entsprechend können in den Feldern AI_LOC1 und AI_LOC2 zwei Zonen (vgl. 5.3.1) eingetragen werden. Standardwert für diese sog. *KI-Zonen* ist die Zone 0, d.h. die gesamte Missionszone. Das eigentliche Verhaltensmodell wird nach Mausklick auf den Button AI_BEHAVIOR eingestellt.

²⁸ KI = Künstliche Intelligenz

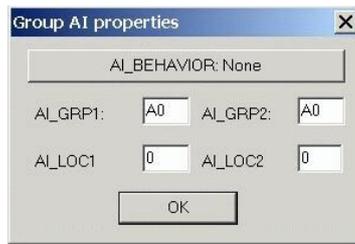


Abb. 44: Einstellen des KI-Verhaltens

Durch Mausklick auf diesen Button öffnet sich das in Abb. 45 gezeigte Fenster. In der Auswahlbox Behavior wird das grundlegende Verhaltensmodell der Gruppe (Standard None) eingestellt. Das große Feld darunter enthält zu dem gewählten Verhaltensmodell einige Optionen, die das Verhalten genauer bestimmen. Die aktiven Optionen sind durch [##] gekennzeichnet.

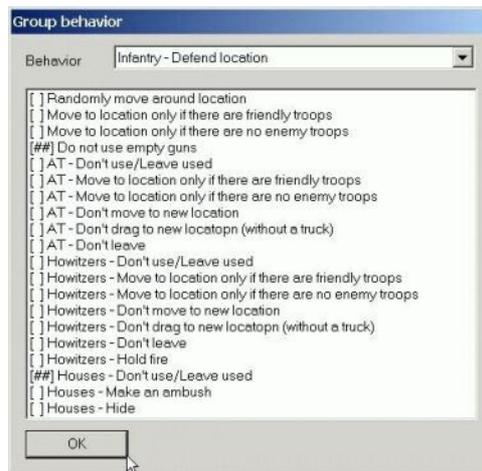


Abb. 45: Einstellen von Verhaltensmodell und Verhaltensoptionen

In der Abbildung ist als Beispiel angegeben, dass die Gruppe eine Zone gegen Feinde verteidigen, dabei aber weder freie Geschütze noch Häuser besetzen soll. Die Zone, die verteidigt werden soll, muss dann im Fenster aus Abb. 44 unter AI_LOC1 bzw. AI_LOC2 angegeben werden.

5.1.4 Einstellung der Wiederbelebungsoptionen

Wie bereits erwähnt, können Einheiten wiederbelebt werden, wenn man ihre Eigenschaft Lives im Einheiteneditor auf einen Wert größer Null gesetzt wurde (vgl. 4.3). Dazu setzt man im Dialog für die Gruppeneigenschaften ein Häkchen unter Auto send im Rahmen Reinforcements. Nach einem Klick auf Options öffnet sich ein weiterer Dialog, in dem man die Art und Weise der Wiederbelebung genauer festlegen kann (vgl. Abb. 46). Bei Verwendung dieser Einstellungen erfolgt die Wiederbelebung automatisch. Mehr Kontrolle über die Wiederbelebung kann man ausüben, wenn man stattdessen die für diesen Zweck zur Verfügung stehenden Skriptbefehle benutzt (vgl. 7.5.8).

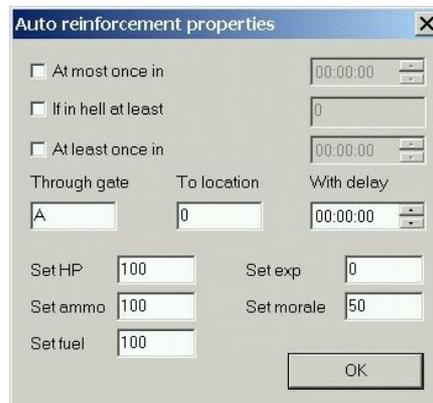


Abb. 46: Einstellung der Optionen für die automatische Wiederbelebung

5.1.5 Löschen von Einheiten

Sollen Einheiten wieder gelöscht werden, so wählt man die entsprechenden Einheiten aus wie im Abschnitt 5.1.2 beschrieben (Edit existing). Danach klickt man auf den Button Delete. Es sind u.U. mehrere Klicks notwendig, bis alle markierten Einheiten gelöscht sind. Vorsicht: Beim Löschen der letzten Einheit aus der Mission stürzt der gesamte Editor oft durch Einfrieren ab. Daher ist es sicherer (und meiner Meinung nach sowieso praktischer), beim Löschen bereits gesetzter Einheiten mit Rechtsklick auf die jeweilige Einheit zu arbeiten.

5.1.6 Arbeit mit Gruppen

Bei der Bearbeitung von Einheiten wird im Hilfsfenster eine Tabelle mit 100 Gruppen angezeigt (vgl. Abb. 42). Die aktuelle Gruppe wird blau hervorgehoben. Zu jeder Gruppe wird die Anzahl der Einheiten angegeben, die zu ihr gehören. Zusätzlich wird ein farbiger Balken eingeblendet, der andeutet, zu welchem Spieler die Einheiten der Gruppe gehören. Die Bedeutung der Farben ist der folgenden Tabelle zu entnehmen.

Farbe	grün	rot	gelb	blau
zugehöriger Spieler	Spieler	Gegner	Alliierte	neutral

Tabelle 2: Farbcodierung der Spieler in der Gruppenansicht

Die Zuordnung von Einheiten zu einer Gruppe wurde bereits beschrieben (s. 5.1.1 und 5.1.2). Durch Rechtsklick auf eine Gruppe im Hilfsfenster wird das Eigenschaftensfenster der Gruppe geöffnet, in dem vor allem das Verhalten der Gruppe festgelegt werden kann.

5.2 Verstärkungen

5.2.1 Verstärkungen bearbeiten

Die während der Mission benötigten *Verstärkungen* kann man über Mission options > Reinforcements oder durch Drücken von Umschalttaste und **R** bearbeiten. Es erscheint der nachstehend abgebildete Dialog, in dem man durch Mausklick auf einen Eintrag der Liste die zu bearbeitende Verstärkung auswählt.

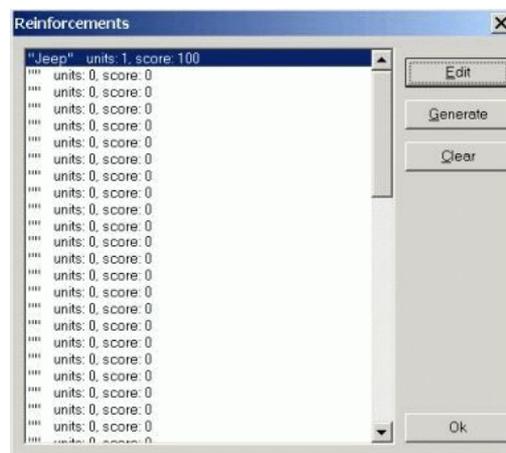


Abb. 47: Dialog für die Auswahl von Verstärkungen

Durch einen Klick auf Clear wird die ausgewählte Verstärkungstruppe gelöscht.

Klickt man auf Edit, kann man die Einheiten für diese Verstärkung im *Einheiteneditor* (vgl. 4.3) bearbeiten. Bei der Bearbeitung von Verstärkungen ist im Einheiteneditor auch das Textfeld am unteren Rand aktiv. Hier kann (und sollte) man der bearbeiteten Verstärkung einen Namen geben, anhand dessen man die Verstärkung bei der Skriptprogrammierung leicht identifizieren kann (vgl. Abb. 48).

Will man die Verstärkung einer bestimmten Gruppe zuweisen, dann sollte man darauf achten, beim Bearbeiten der Verstärkung im Einheiteneditor die Gruppeneinstellung (Group) der enthaltenen Einheiten passend zu setzen.



Abb. 48: Bearbeiten von Verstärkungen

Für viele Einsatzfälle gibt es auch vorgefertigte Verstärkungen für die verschiedenen Nationen, z.B. leichte oder schwere Panzerkampfgruppen, Infanterieverbände usw., wobei diese Verstärkungen ausbalanciert sind, so dass man für die verschiedenen Nationen annähernd gleichwertige Verstärkungen zur Verfügung hat (nützlich besonders für Multiplayer-Missionen).

Will man die vorgefertigten Verstärkungen nutzen, so klickt man statt auf Edit auf Generate. Es öffnet sich ein Dialog, in dem man die gewünschten Verstärkungen auswählen kann (vgl. Abb. 49). In dem Feld neben dem X gibt man zu jeder Truppe an, wie oft sie in der Verstärkung vorhanden sein soll.

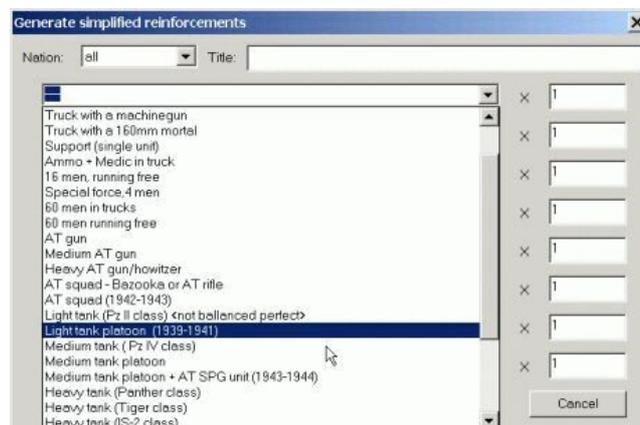


Abb. 49: Vorgefertigte Verstärkungen

Unter Nation stellt man die Nation ein, aus deren Einheiten die Verstärkung bestehen soll. Im Feld Title gibt man der Verstärkung einen Namen für die spätere Identifikation bei der Skriptprogrammierung.

Folgende Einschränkungen sind bei der Erzeugung von Verstärkungen zu beachten:

- Manche Einheiten können nicht als Verstärkung definiert werden. Dazu gehören z.B. stationäre Geschütze wie die Küstenartillerie.
- Einheiten der Verstärkung sollten genug Besatzung haben, um sich bewegen zu können, i.d.R. reicht dazu der Fahrer aus.
- Große Einheiten wie Schiffe und Züge können nicht zusammen mit anderen Einheiten in einer Verstärkung zusammengefasst werden.

Flugzeuge kann man auf zwei Arten als Verstärkung in eine Mission einfügen. Eine Möglichkeit ist die Einstellung entsprechender Optionen in den Spielereinstellungen (vgl. 4.2.1). Flugzeuge können aber auch als normale Verstärkung auf die Karte gelangen. Die Bearbeitung entspricht dann der hier beschriebenen Vorgehensweise.

5.2.2 Tore

Tore (Gates) sind die Orte, an denen Verstärkungen auf die Karte gelangen. Sie werden mit Buchstaben von A bis Z bezeichnet, d.h. es können maximal 26 Tore existieren. Nach Auswahl der Kategorie Gates im Werkzeugbaum kann man in dem Auswahlfeld Gate unterhalb des Werkzeugbaums das zu bearbeitende Tor einstellen. Das Tor wird dann durch Linksklick auf die Karte gesetzt und als rote Fahne mit dem entsprechenden Buchstaben dargestellt.

Die übliche Methode, Verstärkungen auf die Karte zu bringen, ist es, sie am Rand der Karte auftauchen zu lassen. Dazu setzt man das Tor auf ein direkt an die Missionszone angrenzendes Feld, d.h. auf eines der inneren Felder in dem heller dargestellten Rahmen, der die Missionszone begrenzt. Die Breite des Tors kann dann über den Regler Width angepasst werden. Je breiter das Tor ist, desto mehr Einheiten können gleichzeitig die Missionszone erreichen. Die korrekte Platzierung zeigt die folgende Abbildung.



Abb. 50: Fünf Felder breites Tor B am Rand der Missionszone

Wird das Tor innerhalb der Missionszone plaziert, z.B. als Tor für Verstärkungen aus Fabriken bei SSRW, so ist die Breite automatisch auf ein Feld beschränkt. Der Regler Width hat dann keine Wirkung.

Um den Einheiten, die durch das Tor die Missionszone erreichen, ein Ziel zu geben, zu dem sie sich bewegen, gibt man in der Skriptaktion (vgl. 7.5.8, 7.6) oder im Dialog für die automatische Wiederbelebung (vgl. 5.1.4) eine Zone an. Gibt man die Zone 0 an, die die gesamte Missionszone repräsentiert, ist nicht eindeutig bestimmt, wohin sich die Einheiten bewegen sollen. Daher setzt man zu einem Tor eine weitere Markierung mit einem Rechtsklick in die Missionszone. Es erscheint eine blaue Fahne mit dem Buchstaben des zugehörigen Tors. In der Umgebung dieser Fahne sammeln sich dann die Verstärkungen, wenn die Missionszone als Ziel angegeben wird.

5.3 Orte

Es gibt zwei unterschiedliche Möglichkeiten, Orte auf der Missionszone zu definieren. Außerdem gibt es weitere ortsabhängige Optionen, die ebenfalls nachfolgend beschrieben werden.

5.3.1 Locations

Größere Bereiche der Missionszone, die für die Mission von Bedeutung sind, werden als *Zonen* oder *Orte (Locations)* bezeichnet. Man kann bis zu 63 Orte definieren, die von 1 bis 63 durchnummeriert werden. Der Ort mit der Nummer 0 ist vorgegeben und repräsentiert die gesamte Missionszone.

	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55
56	57	58	59	60	61	62	63

Abb. 51: Übersicht über die Zonen

Bei der Auswahl der Kategorie Locations im Werkzeugbaum wird im Hilfsfenster eine tabellarische Übersicht der Zonen 1 bis 63 eingeblendet, wobei die aktuelle Zone blau hervorgehoben ist (vgl. Abb. 51).

Orte werden nach Auswahl der Kategorie Locations im Werkzeugbaum mit einem Pinsel auf die Karte gemalt. Die Vorgehensweise ist analog zur Definition von Terrain (vgl. 3.2.1): Setzen mit Linksklick, löschen mit Rechtsklick. Die zu der Zone gehörenden Felder werden blau dargestellt und mit der Nummer der Zone versehen (vgl. Abb. 52). Eine Zone muss übrigens nicht zusammenhängend sein, sondern kann auch aus mehreren voneinander getrennten Teilzonen bestehen.

Die Nummer der zu bearbeitenden Zone kann man entweder durch Linksklick auf die Zone in der Übersicht im Hilfsfenster oder im Auswahlfeld Location unterhalb des Werkzeugbaums festlegen. Durch Klick auf Delete lässt sich die gesamte Zone bequem wieder von der Karte entfernen.

Um Zonen schnell wiederfinden zu können, werden sie bei der Bearbeitung auch in der Minikarte in blau angezeigt. Die ausgewählte Zone wird in der Minikarte in weiß dargestellt und ist so relativ leicht zu lokalisieren.

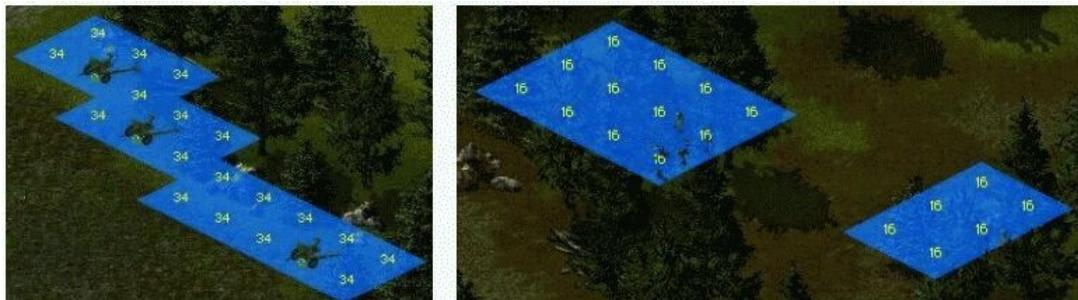


Abb. 52: Darstellung von Zonen auf der Karte (rechts: Zone mit zwei Teilzonen)

5.3.2 Marker

Sollen bestimmte Stellen auf der Karte, z.B. bestimmte Gebäude, in Skripten verwendet werden, so kann man diese Stellen mit *Markern* versehen, von denen es 100 gibt, die von 0 bis 99 durchnummeriert sind. Im Skript kann man die Nummer des Markers dann benutzen, um z.B. ein Gebäude darüber anzusprechen. Wählt man im Werkzeugbaum die Kategorie Markers aus, so kann man den zu setzenden Marker in der Tabelle im Hilfsfenster oder im Auswahlfeld unterhalb des Werkzeugbaums wählen. Mit einem Linksklick auf die Karte wird der aus einem einzigen Feld bestehende Marker gesetzt. Er kann durch Rechtsklick an die gleiche Stelle wieder entfernt werden.

Brücken, die aus mehreren Brückenteilen bestehen, werden nicht als ein einzelnes Gebäude gezählt, sondern jeder Brückenteil stellt ein Gebäude dar. Daher braucht man für jeden Brückenteil einen Marker, wenn man die gesamte Brücke über Marker ansprechen will.

5.3.3 Minenfelder

Minenfelder können bei der Erstellung einer Mission angelegt werden, wenn man im Werkzeugbaum die Kategorie Mines auswählt. Unterhalb des Werkzeugbaums kann man dann die Art der Minen einstellen. Es gibt Panzerminen (Anti-tank), Minen mit Zeitzünder (Time bomb) und Personenminen (Anti-personell). Die Teams, für die die Minen sichtbar sind, kann man ebenfalls einstellen (Visible to team 1, 2, 3, 4), wobei die Teamnummer sich auf die Teams bezieht, die in den Spielereinstellungen (vgl. 4.2.1) angegeben wurden.

Die Minenfelder werden dann mit einem Pinsel wie bei der Definition von Orten oder von Terrain auf die Karte gemalt. In der Darstellung auf der Karte kann man an der hellen Darstellung der vier winzigen Felder in dem roten Minensymbol erkennen, für welche Teams die Mine sichtbar ist. In Abb. 54 sind die oberen beiden Minen für Team 1 bzw. 2 sichtbar, die untere kann von Einheiten der Teams 3 und 4 gesehen werden.

5.3.4 Ortsabhängige Sounds

Man kann in der Mission an ausgewählten Orten *Soundeffekte* (z.B. Wasserrauschen, Grillenzirpen, Vogelgezwitscher) abspielen lassen. Dazu wählt man im Werkzeugbaum die Kategorie Sounds aus. Nach einem Klick auf den Button Add new unterhalb des Werkzeugbaums öffnet sich ein Dialog, in dem man den gewünschten Soundeffekt auswählen kann (vgl. Abb. 53).

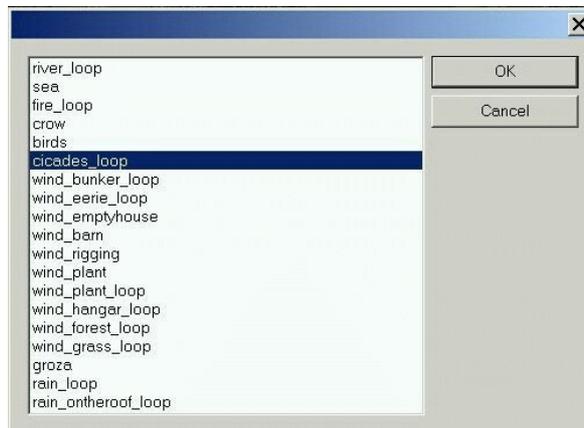


Abb. 53: Auswahl von Soundeffekten

Nach einem Klick auf OK kann man den Sound in der Missionszone plazieren, indem man auf die gewünschte Stelle klickt. Es erscheint dort ein Lautsprechersymbol.

Nach einem Klick auf Add existing und anschließend auf eines der Lautsprechersymbole kann man ggf. noch die Einstellungen des Soundeffekts verändern. Dazu verändert man unterhalb des Werkzeugbaums die Werte Radius (Anzahl der Felder im Umkreis um das Symbol, in denen der Soundeffekt zu hören sein soll), Priority (je niedriger der Wert, desto eher wird der Soundeffekt nicht abgespielt, wenn in einer Spielsituation u.U. nicht genügend Soundkanäle für alle Sounds zur Verfügung stehen), Min delay und Max delay (minimale bzw. maximale Dauer der Pause zwischen zwei Wiederholungen des Soundeffekts).

Durch einen Bug im Editor werden die Soundinformationen nicht gespeichert, so dass beim erneuten Laden der Mission alle bereits gesetzten Soundeffekte wieder verschwunden sind. Aus diesem Grund sollte man die Soundeffekte erst nach Abschluss aller anderen Arbeiten einfügen und dann sofort die spielbare Mission erzeugen. Um die Soundeffekte ggf. wieder rekonstruieren zu können, notiere ich mir, welche Sounds ich an welcher Stelle eingesetzt habe.

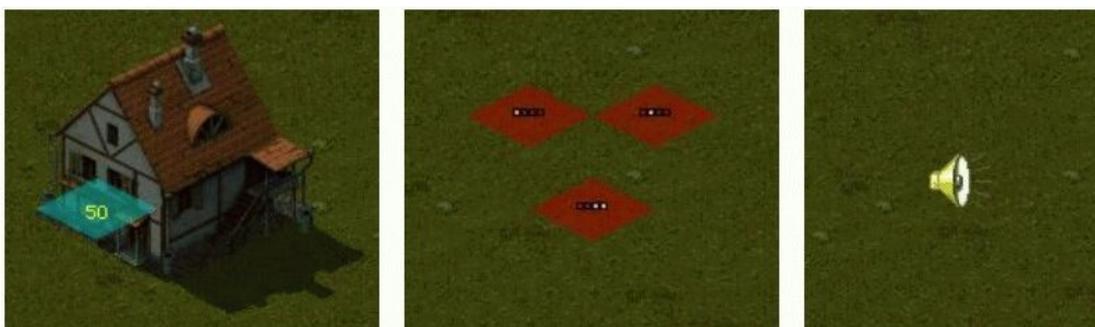


Abb. 54: Darstellung von Markern (links), Minen (Mitte) und Soundeffekten (rechts) im Editor

5.4 Texte

Im Verlauf einer Mission werden oft Texte angezeigt, die Informationen für den Spieler enthalten. Neben der in 4.2.2 behandelten Beschreibung der Missionsziele sind dies Texte, die von Skripten eingeblendet werden. Diese Texte werden in einem Fenster bearbeitet, das man über Mission options > Phrases öffnet.

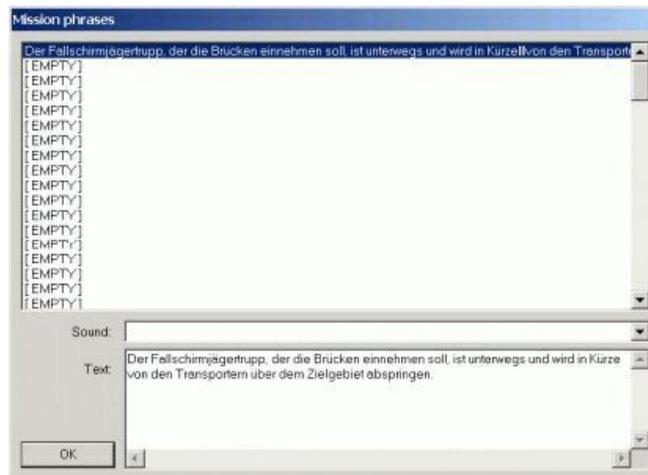


Abb. 55: Der Editor für Missionsinformationen

Im oberen Teil des Dialogs sind die Slots zu sehen, die jeweils einen Text enthalten können. Ein noch leerer Slot wird als [EMPTY] angezeigt. Enthält ein Slot bereits einen Text, so wird der Anfang dieses Textes eingeblendet. Insgesamt stehen 100 Slots zur Verfügung. Durch Linksklick auf einen Slot kann ein Text ausgewählt werden, der dann im mit Text bezeichneten Eingabefeld bearbeitet werden kann. Soll die Anzeige des Textes mit einem Soundeffekt unterlegt werden, kann der passende in dem Auswahlmeneü Sound festgelegt werden. Der Name des Soundeffekts wird ebenfalls im oberen Teil des Dialogs eingeblendet (vor dem Anfang des eigentlichen Textes).

6 Verhalten von Einheiten

Wie man das Verhalten von Einheiten einstellt, wurde bereits in 5.1.3 beschrieben. An dieser Stelle findet man nun die genaue Erläuterung der möglichen *Verhaltensmodelle* und der jeweils zur Verfügung stehenden *Verhaltensoptionen*, die das Verhalten einer Gruppe von Einheiten genauer festlegen. In der folgenden Abbildung ist die Liste aller Verhaltensmodelle zu sehen, aus denen man wählen kann.

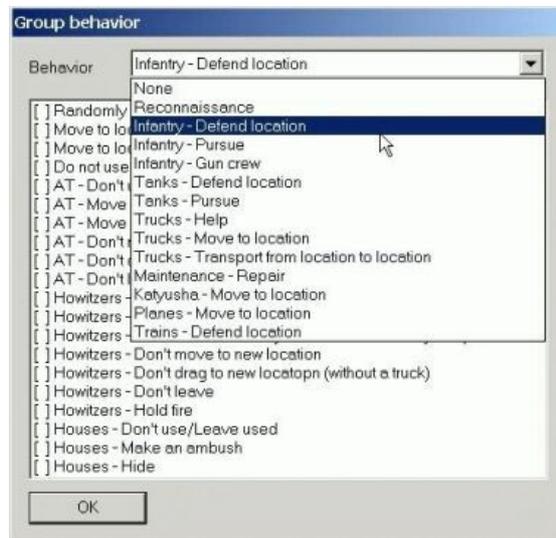


Abb. 56: Auswahl eines Verhaltensmodells

Die Liste mit den Verhaltensmodellen ist danach geordnet, für welche Einheitentypen die Verhaltensmodelle gedacht sind. Für Infanterie-, Panzer- und Lkw-Einheiten sind jeweils mehrere Verhaltensmodelle vorhanden. Für andere Einheitentypen (Versorgungsfahrzeuge, Raketenwerfer, Flugzeuge und Züge) gibt es nur jeweils ein spezifisches Verhaltensmodell. Die Verhaltensmodelle None und Reconnaissance sind nicht an bestimmte Einheitentypen gebunden.

Das Verhaltensmodell beschreibt das allgemeine Verhalten der Einheiten. Dazu gibt es im Normalfall eine Reihe von Optionen, die das Verhalten, das durch das Verhaltensmodell vorgegeben ist, in gewissem Umfang verändern können.

Eine Bemerkung vorab zu diesen Verhaltensoptionen: Nach meinen Erfahrungen ist es nicht immer ganz klar, auf welche Arten von Einheiten sich die Option tatsächlich auswirkt. Mir war es aber zu mühsam, in jedem Verhaltensmodell zu jeder Kombination von Verhaltensoptionen auszuprobieren, mit welchen Einheiten sie evtl. nicht so funktioniert, wie man das erwarten würde. Im Zweifelsfall hilft da nur ausprobieren. Die Punkte, die mir selbst noch nicht klar sind, habe ich entsprechend kommentiert. Wer mir da mit Informationen weiterhelfen kann, der möge das bitte tun.

6.1 Nicht an Einheitentypen gebundene Verhaltensmodelle

Wie oben bereits erwähnt sind nur zwei Verhaltensmodelle nicht an bestimmte Einheitentypen gebunden: None und Reconnaissance.

6.1.1 Das Verhaltensmodell None

Das Verhaltensmodell None ist das standardmäßig voreingestellte Verhaltensmodell für jede Gruppe von Einheiten. Gleichzeitig ist es das einfachste und unspezifischste Modell der gesamten Liste. Einheiten, die dieses Verhaltensmodell besitzen, bleiben an ihrer Position stehen. Sie greifen feindliche Einheiten nur dann an, wenn diese in ihren Sichtbereich kommen.

Die Einstellungen AI_GRP1, AI_GRP2, AI_LOC1 und AI_LOC2 haben in diesem Verhaltensmodell keine Auswirkungen.

Zu diesem Verhaltensmodell gibt es keine Verhaltensoptionen.

6.1.2 Das Verhaltensmodell Reconnaissance

Das Verhaltensmodell Reconnaissance soll dem Namen nach dazu wohl dazu dienen, Einheiten auf Aufklärung zu schicken. Auch in der deutschen Version des Originalhandbuchs wird das so beschrieben, allerdings ohne jedes weitere Detail. In der englischen Version des Handbuchs ist das Verhaltensmodell gar nicht erst erwähnt.

Trotz unterschiedlicher Versuche habe ich kein anderes Verhalten feststellen können als beim Verhaltensmodell None. Wer Reconnaissance schon erfolgreich eingesetzt hat, kann mir ja mal ein paar Infos zukommen lassen.

Zu diesem Verhaltensmodell gibt es keine Verhaltensoptionen.

6.2 Verhaltensmodelle für Infanterie-Einheiten

Die in diesem Abschnitt beschriebenen Verhaltensmodelle beziehen sich auf Gruppen, die aus Infanterie-Einheiten bestehen. Zur Infanterie im Sinne der Infanterie-Verhaltensmodelle zählen auch Geschütze (Flak, Pak, Haubitzen, Mörser, MG) und Jeeps (auch Pkw). Die Gruppe kann auch andere Einheiten (z.B. Panzer oder Lkw) enthalten, die dann aber nicht das Verhalten der Infanterie übernehmen.

6.2.1 Das Verhaltensmodell Infantry – Defend location

Das Verhaltensmodell Infantry – Defend location wird verwendet, wenn eine Infanterie-Gruppe eine bestimmte Zone verteidigen soll. Die Gruppe begibt sich zu der angegebenen Zone und verteidigt sie gegen feindliche Einheiten. Es können auch zwei Zonen angegeben werden, auf die die Einheiten sich aufteilen.

Die Einheiten besetzen Häuser in der angegebenen Zone, wenn diese relativ nahe gelegen sind. Sie benutzen Geschütze und Jeeps, die zur Gruppe gehören, in jedem Fall und verlassen diese nicht. Geschütze und Jeeps anderer Gruppen werden auch verwendet, falls sie sich in einer der zu verteidigenden Zonen befinden und in den Optionen zum Verhaltensmodell nichts anderes eingestellt wurde.

Einstellungen

Die zu verteidigenden Zonen werden als AI_LOC1 und AI_LOC2 angegeben. Soll die Infanterie den Ort verteidigen, an dem sie steht, dann setzt man sowohl für AI_LOC1 als auch für AI_LOC2 die Nummer der Zone ein, in der die Einheiten sich befinden. Das kann auch der Wert 0 (für die gesamte Missionszone) sein.

Gibt man sowohl für AI_LOC1 als auch für AI_LOC2 eine andere Zone als den aktuellen Standort der Einheiten an, dann bewegen sich die Einheiten dorthin.

Man kann die Gruppe auch zwei Zonen bewachen lassen. Dazu trägt man die Nummer der einen Zone als AI_LOC1 und die der anderen als AI_LOC2 ein. Aber Vorsicht! Wenn die Einheiten bereits in einer der beiden KI-Zonen stehen, dann bewegen sich die Einheiten der Gruppe nicht mehr in die andere Zone. Die Einheiten bleiben dann einfach an dem Ort stehen, wo sie sich gerade befinden, weil sie sich ja damit schon in einer der beiden zu verteidigenden Zonen befinden und somit ihren Auftrag bereits erfüllen.

Die Einstellungen für AI_GRP1 und AI_GRP2 werden nicht verwendet.

Verhaltensoptionen der beweglichen Infanterie-Einheiten

Mit beweglichen Infanterie-Einheiten meine ich hier die Infanteristen, die sich entweder zu Fuß oder als Passagiere von Jeeps der Gruppe frei bewegen können, also nicht bereits Häuser oder Geschütze besetzen.

- Randomly move around location: Die Einheiten stehen nicht einfach herum, sondern bewegen sich von Zeit zu Zeit zu einer anderen Stelle in der angegebenen Zone.

- Move to location only if there are friendly troops: Bei dieser Option bewegen sich die Einheiten nur dann in die angegebene Zone, wenn sich dort bereits freundlich gesinnte Einheiten befinden.
- Move to location only if there are no enemy troops: Die Einheiten bewegen sich nur zu der angegebenen Zone, wenn diese feindfrei ist.
- Do not use empty guns: Da die Optionen der Verhaltensmodelle ja angeblich selbsterklärend sein sollen, hatte ich diese Option so verstanden, dass die Einheiten der Gruppe keine unbesetzten Geschütze anderer Gruppen benutzen. Die Realität sieht nach meinen Erfahrungen aber anders aus. Freie Geschütze anderer Gruppen werden sehr wohl trotzdem besetzt. Will man das verhindern, kann man aber die beiden Optionen AT – Don't use/Leave used und Howitzers – Don't use/Leave used setzen.

Verhaltensoptionen für den Umgang mit Panzerabwehrgeschützen

Die nachfolgend erläuterten Optionen beziehen sich ausschließlich auf die Panzerabwehrgeschütze der Gruppe (AT=Anti-Tank). Dabei ist es anscheinend nicht einheitlich geregelt, welche Waffen in diesem Zusammenhang als Pak gelten (vgl. einzelne Erläuterungen zu den Optionen).

Wenn nicht mittels der folgenden Optionen etwas anderes eingestellt wird, dann benutzen die Infanteristen der Gruppe die Geschütze. Wird die zu bewachende Zone verändert, dann warten die Geschützbesatzungen eine kurze Zeit (ca. 15 Sekunden) auf einen Lkw, der das Geschütz zu der neuen Zone bringt. Nach dieser Wartezeit ziehen sie das Geschütz zu Fuß in die neue Zone.

- AT – Don't use/Leave used: Die Einheiten sollen bei der Verteidigung der angegebenen Zone keine Panzerabwehrgeschütze anderer Gruppen benutzen. Darunter fallen nach meinen Erfahrungen nicht nur die als Pak bezeichneten Geschütze, sondern alle, die nicht ausschließlich ballistisch feuern, also auch Flak (keine Flakpanzer oder Selbstfahrlafetten!) und MG. Auch Haubitzen, die direkten Beschuss ermöglichen, sind in diesem Sinne Panzerabwehrgeschütze.
- AT – Move to location only if there are friendly troops: Wie die entsprechende Option oben, aber für Panzerabwehrgeschütze.
- AT – Move to location only if there are no enemy troops: Wie die entsprechende Option oben, aber für Panzerabwehrgeschütze.
- AT – Don't move to new location: Wird diese Option angegeben, dann verlassen die Geschützbesatzungen bei einem Zonenwechsel die Geschütze, statt sie zu der neuen Zone zu ziehen (oder ggf. von einem Lkw ziehen zu lassen). Wird jedoch gleichzeitig auch die Option AT - Don't leave angegeben, dann halten die Geschützbesatzungen immer ihre Position, auch bei einem Stellungswechsel.

Bei dieser Option scheint mir einer der Fälle vorzuliegen, in denen nicht ganz klar ist, auf welche Einheiten die Option nun genau wirkt. Zum Beispiel wird die deutsche Feldhaubitze 18 (105mm) nicht wie andere Paks verlassen, sondern zu der neuen Zone gezogen, obwohl sie z.B. im Einheiteneditor und auch bei der Option AT – Don't use/Leave used als Panzerabwehrgeschütz eingeordnet wird. Bei dieser Option wird sie dagegen offenbar als Haubitze betrachtet.

- AT – Don't drag to new location (without a truck): Diese Option wirkt ähnlich wie AT – Don't move to new location. Der Unterschied liegt darin, dass das Geschütz zwar nicht von Hand, wohl aber von einem Lkw zu der Zone gezogen werden darf. Ist nicht relativ schnell (innerhalb von ca. 15 Sekunden) ein Lkw zum Transport zur Stelle, verlässt die Besatzung das Geschütz und bewegt sich zur zu verteidigenden Zone.
- AT – Don't leave: Die Geschützbesatzungen verlassen das Geschütz grundsätzlich nicht. Ist eine der beiden Optionen AT – Don't move to new location bzw. AT – Don't drag to new location (without a truck) angegeben, dann bleibt die Besatzung mit dem Geschütz an der bisherigen Position.

Verhaltensoptionen für den Umgang mit Haubitzen

Die für Panzerabwehrgeschütze geltenden Optionen gibt es jeweils auch in einer Version für Haubitzen. Dabei wird nur das vorangestellte „AT“ der Optionsbezeichnung durch „Howitzer“ ersetzt. Die Wirkung ist jeweils entsprechend.

Es gibt für Haubitzen aber noch eine weitere Option:

- Howitzers – Hold fire: Bei dieser Option bin ich mir bzgl. der Auswirkungen nicht sicher. Nach unterschiedlichen Versuchen scheint es mir, als ob die Haubitzen der Gruppe nur dann feuern, wenn sie in der zu verteidigenden Zone stehen.

Verhaltensoptionen für den Umgang mit Häusern

Die folgenden Optionen regeln das Verhalten der Gruppe in Bezug auf Häuser in der zu bewachenden Zone.

- Houses – Don't use/Leave used: Normalerweise besetzen die Infanteristen der Gruppe die Häuser der Zone, die sie verteidigen sollen, da sie dort besser geschützt sind. Will man keine „Hausbesetzer“, dann muss man diese Option aktivieren.
- Houses – Make an ambush: Die Infanteristen besetzen Häuser, schießen aber nicht auf feindliche Einheiten, wenn diese nicht das Haus betreten. Wenn der Sieg in einer Mission davon abhängt, dass alle gegnerischen Einheiten vernichtet werden, kann das zu langwierigen Suchaktionen führen, bis man die letzten in Häusern versteckten Soldaten findet.
- Houses – Hide: Der Unterschied zu Houses – Make an ambush ist mir bei meinen Versuchen nicht klar geworden. Erleuchtet mich!

6.2.2 Das Verhaltensmodell Infantry – Pursue

Das Verhaltensmodell Infantry – Pursue entspricht im wesentlichen dem Verhaltensmodell Infantry – Defend location. Der wesentliche Unterschied besteht darin, dass die Einheiten der Gruppe einer Einheit aus den KI-Gruppen AI_GRP1 und AI_GRP2 folgen, wenn diese gesichtet wird. Die Verfolgung wird beendet, wenn die verfolgte Einheit außer Sicht gerät. Die Einheiten der Verfolgergruppe kehren dann wieder in die zu verteidigende Zone zurück.

Die von der KI gesteuerten Einheiten stellen sich bei der Verfolgung allerdings ziemlich dämlich an, so dass das gezeigte Verhalten nicht besonders gut einzusetzen ist.

Einstellungen

Die zu verteidigenden Zonen werden als AI_LOC1 und AI_LOC2 angegeben. Die Anmerkungen aus 6.2.1 gelten auch hier.

Die Einstellungen für AI_GRP1 und AI_GRP2 legen die Gruppen fest, die ggf. verfolgt werden sollen.

Verhaltensoptionen

Gegenüber Infantry – Defend location ist der Umfang der Verhaltensoptionen erheblich geringer. Folgende Optionen stehen zur Auswahl:

- Randomly move around location: Die Einheiten stehen nicht einfach herum, sondern bewegen sich von Zeit zu Zeit zu einer anderen Stelle in der angegebenen Zone.
- Randomly select pursued target: Die verfolgte Einheit wird zufällig ausgewählt, wenn mehrere zu sehen sind.
- Move to location only if there are friendly troops: Bei dieser Option bewegen sich die Einheiten nur dann in die angegebene Zone, wenn sich dort bereits freundlich gesinnte Einheiten befinden.
- Move to location only if there are no enemy troops: Die Einheiten bewegen sich nur zu der angegebenen Zone, wenn diese feindfrei ist.

- Do not use empty guns: Wie bei Infantry – Defend location. Die Option verhindert auch hier nicht das Besetzen von freien Geschützen anderer Gruppen.
- AT – Don't use/Leave used: Wie bei Infantry – Defend location.
- Houses – Don't use/Leave used: Wie bei Infantry – Defend location.

6.2.3 Das Verhaltensmodell Infantry – Gun crew

Beim Verhaltensmodell Infantry – Gun crew ergeben sich lediglich zwei Änderungen gegenüber dem Verhaltensmodell Infantry – Defend location. Als zu bewachende Zone dient ausschließlich die als AI_LOC1 angegebene Zone, während AI_LOC2 als Beobachtungszone²⁹ für die Haubitzen gilt. Als zweite Änderung werden Jeeps grundsätzlich nicht benutzt. Mit Ausnahme dieser beiden Änderungen gelten alle Erläuterungen zu Infantry – Defend location auch hier.

6.3 Verhaltensmodelle für Panzer-Einheiten

6.3.1 Das Verhaltensmodell Tanks – Defend location

Das Verhaltensmodell Tanks – Defend location wird verwendet, wenn eine Gruppe von Panzern eine bestimmte Zone verteidigen soll. Die Gruppe begibt sich zu der angegebenen Zone und verteidigt sie gegen feindliche Einheiten. Es können auch zwei Zonen angegeben werden, auf die die Einheiten sich aufteilen.

Zu Panzern zählen neben Kampfpanzern, Jagdpanzern, Schützen- und Spähpanzern auch Flakpanzer sowie Haubitzen und Flak auf Selbstfahrlafette. Raketenwerfer (Nebelwerfer, Calliope usw.) zählen nicht zu den Panzern im Sinne dieses Verhaltensmodells. Gehören auch Einheiten zu der Gruppe, die nicht zu den Panzern gezählt werden, dann ignorieren sie das Verhalten der Panzer-Einheiten. Für sie gilt also im Prinzip das Verhaltensmodell None.

Einstellungen

Die Einstellungen für dieses Verhaltensmodell entsprechen denen von Infantry – Defend location. Die zu verteidigenden Zonen werden als AI_LOC1 und AI_LOC2 angegeben. Die Einstellungen für AI_GRP1 und AI_GRP2 werden nicht verwendet.

Verhaltensoptionen

- Randomly move around location: Die Panzer stehen nicht einfach herum, sondern bewegen sich von Zeit zu Zeit zu einer anderen Stelle in der angegebenen Zone. Bei SSRW sollte man dabei den Treibstoffverbrauch beachten und ggf. einige Tankwagen zur Versorgung in der Nähe bereithalten.
- Move to location only if there are friendly troops: Bei dieser Option bewegen sich die Panzer nur dann in die angegebene Zone, wenn sich dort bereits freundlich gesinnte Einheiten befinden.
- Move to location only if there are no enemy troops: Die Panzer bewegen sich nur zu der angegebenen Zone, wenn diese feindfrei ist.

6.3.2 Das Verhaltensmodell Tanks – Pursue

Das Verhaltensmodell Tanks – Pursue überträgt das Verhaltensmodell Infantry – Pursue auf Panzer-Einheiten. Die Panzer der Gruppe folgen also einer Einheit aus den KI-Gruppen AI_GRP1 und AI_GRP2, wenn diese gesichtet wird. Die Verfolgung wird beendet, wenn die verfolgte Einheit außer Sicht gerät. Die Panzer der Verfolgergruppe kehren dann wieder in die zu verteidigende Zone zurück.

Da sich die Panzer bei diesem Verhaltensmodell u.U. oft bewegen, sollte man bei SSRW an die Treibstoffversorgung denken.

²⁹ Nach den vorhandenen Unterlagen und eigenen Versuchen bin ich mir bzgl. der Bedeutung der Beobachtungszone nicht sicher. Wer kann mich da aufklären?

Einstellungen

Die zu verteidigenden Zonen werden als AI_LOC1 und AI_LOC2 angegeben. Die Anmerkungen aus 6.2.1 gelten auch hier.

Die Einstellungen für AI_GRP1 und AI_GRP2 legen die Gruppen fest, die ggf. verfolgt werden sollen.

Verhaltensoptionen

Die folgenden Optionen stehen zur Auswahl:

- Randomly move around location: Die Panzer stehen nicht einfach herum, sondern bewegen sich von Zeit zu Zeit zu einer anderen Stelle in der angegebenen Zone.
- Randomly select pursued target: Die verfolgte Einheit wird zufällig ausgewählt.
- Move to location only if there are friendly troops: Bei dieser Option bewegen sich die Panzer nur dann in die angegebene Zone, wenn sich dort bereits freundlich gesinnte Einheiten befinden.
- Move to location only if there are no enemy troops: Die Panzer bewegen sich nur zu der angegebenen Zone, wenn diese feindfrei ist.

6.4 Verhaltensmodelle für Transport-Einheiten

6.4.1 Das Verhaltensmodell Trucks – Help

Das Verhaltensmodell Trucks – Help wird verwendet, wenn Geschütze zu transportieren sind. Transportfahrzeuge (Lkw und Schützenpanzer, die Geschütze ziehen können) helfen freundlich gesinnten Einheitengruppen, ihre Geschütze zu transportieren. Infanterie, die sich in den Transportern aufhält, wird ausgeladen, sobald das Verhaltensmodell auf die Transporter angewendet wird.

Während ein Geschütz angehängt ist, unterliegt der Transporter dem Verhaltensmodell der Geschützbesatzung. Sobald das Geschütz abgehängt wird, begibt sich der Transporter in eine der angegebenen Zonen, wo er auf weitere Transporte wartet.

Da sich die Transporter bei diesem Verhaltensmodell u.U. oft bzw. über längere Strecken bewegen, sollte man bei SSRW an die Treibstoffversorgung denken. Dies gilt besonders beim Einsatz von Schützenpanzern, weil diese im Vergleich zu Lkw oft eine relativ geringe Reichweite haben.

Einstellungen

Die KI-Zonen AI_LOC1 und AI_LOC2 bezeichnen die Ruhezonen für die Transporter, in denen diese auf den nächsten Transportauftrag warten. Man kann dabei eine oder zwei Zonen festlegen, wobei die Transporter sich im zweiten Fall auf die beiden Ruhezonen verteilen.

Die Einstellungen für AI_GRP1 und AI_GRP2 legen die Gruppen fest, die unterstützt werden sollen. Mit Hilfe der Option Help only the selected groups lässt sich die Unterstützung auf diese Gruppen beschränken.

Verhaltensoptionen

Die folgenden Optionen stehen zur Auswahl:

- Randomly move around location: Die Transportfahrzeuge stehen nicht einfach herum, sondern bewegen sich von Zeit zu Zeit zu einer anderen Stelle in der angegebenen Zone.
- Randomly select pursued target: Die unterstützte Einheit wird zufällig ausgewählt.
- Move to location only if there is no one to pursue: Bei dieser Option bewegen sich die Transporter nur dann in die angegebene Zone, wenn es in der Nähe keine Geschütze gibt, die transportiert werden müssen.
- Move to location only if there are friendly troops: Bei dieser Option bewegen sich die Transportfahrzeuge nur dann in die angegebene Zone, wenn sich dort bereits freundlich gesinnte Einheiten befinden.

- Move to location only if there are no enemy troops: Die Transportfahrzeuge bewegen sich nur zu der angegebenen Zone, wenn diese feindfrei ist.
- Help only the selected groups: Die Transporter unterstützen nur die angegebenen Gruppen. Ist diese Option nicht aktiviert, werden alle freundlich gesinnten Einheiten unterstützt, die sich in der Nähe der Transporter befinden.

6.4.2 Das Verhaltensmodell Trucks – Move to location

Das Verhaltensmodell Trucks – Move to location wird verwendet, wenn Transportfahrzeuge sich zu einer bestimmten Zone bewegen sollen. Die Transportfahrzeuge der Gruppe fahren zu der angegebenen Zone. Falls sie beladen sind, wird die Ladung (Infanterie, stationäre MG, Mörser, Ersatzteile und Ölfässer) in der Zielzone ausgeladen.

Einstellungen

Die KI-Zonen AI_LOC1 und AI_LOC2 bezeichnen die Zonen, zu denen die Transporter sich bewegen sollen. Werden zwei verschiedene Zonen angegeben, fahren die Fahrzeuge laut der englischen Version des Originalhandbuchs zu der näher gelegenen Zone. Nach meinen Versuchen scheint es aber eher so zu sein, dass bei Angabe verschiedener Zonen immer die erste, also AI_LOC1, angesteuert wird.

Die Einstellungen für AI_GRP1 und AI_GRP2 werden in diesem Verhaltensmodell nicht verwendet.

Verhaltensoptionen

Die folgenden Optionen stehen zur Auswahl:

- Randomly move around location: Die Transportfahrzeuge stehen nicht einfach herum, sondern bewegen sich von Zeit zu Zeit zu einer anderen Stelle in der angegebenen Zone.
- Move to location only if there are friendly troops: Bei dieser Option bewegen sich die Transportfahrzeuge nur dann in die angegebene Zone, wenn sich dort bereits freundlich gesinnte Einheiten befinden.
- Move to location only if there are no enemy troops: Die Transportfahrzeuge bewegen sich nur zu der angegebenen Zone, wenn diese feindfrei ist.

6.4.3 Das Verhaltensmodell Trucks – Transport from location to location

Das Verhaltensmodell Trucks – Transport from location to location wird verwendet, wenn Infanterie von einer Zone zu einer anderen transportiert werden soll. Man kann der Infanterie zwar z.B. im Verhaltensmodell Infantry – Defend location eine andere Zone zuweisen, bei großen Entfernungen dauert der Stellungswechsel aber relativ lange. Ein Stellungswechsel mit Hilfe von Transportfahrzeugen ist u.U. weniger zeitintensiv.

Die Transportfahrzeuge der Gruppe fahren zu der ersten angegebenen Zone, nehmen Infanterie auf und transportieren sie zur zweiten angegebenen Zone, sobald der Transporter mindestens zu 70% besetzt ist. Dieser Ablauf wiederholt sich, bis in der ersten Zone keine Infanterieeinheiten mehr zu transportieren sind.

Eine Warnung: Man sollte sich von diesem Verhaltensmodell nicht zu viel versprechen. In der Theorie hört sich das ja sehr schön an: Lkw zu Zone 1, Soldaten drauf, ab zur Zone 2. In der Praxis stellen sich die von der KI gesteuerten Einheiten aber wieder mal ziemlich dämlich an. Da wird planlos durcheinandergewuselt, weil die Zuordnung der Soldaten zu den Transportern anscheinend zufällig erfolgt. Dann kommt erschwerend die in Sudden Strike ja noch nie besonders gelungene Wegfindung dazu. Einigen Infanteristen dauert es dann – verständlicherweise – zu lange und so machen sie sich dann doch zu Fuß auf den Weg. Zu guter Letzt werden u.U. eine ganze Reihe Soldaten gar nicht erst in Zone 2 ankommen, weil die Transporter auf 70% Auslastung warten – bei der letzten Fuhre leider oft vergeblich. Da stehen dann manchmal mehrere Transporter mit jeweils zwei, drei oder noch mehr Soldaten besetzt herum und warten auf weitere, damit die 70% erreicht werden, was aber nie passiert, weil alle anderen schon in der Zielzone stehen. OK, Meckermodus aus, vielleicht habe ich ja auch irgendwas falsch gemacht...

Einstellungen

Die Zone, in der Infanterie abgeholt werden soll, wird als AI_LOC1 angegeben. Die Zielzone, zu der die Infanterie transportiert werden soll, trägt man als AI_LOC2 ein. Das Handbuch schweigt sich allerdings darüber aus, dass die Infanterie anscheinend eines der Infanterie-Verhaltensmodelle haben muss und dass eine der Zonen diejenige sein muss, die für die Transporter als AI_LOC2 festgelegt wurde. Das ist zwar verständlich, weil sonst u.U. der Auftrag der Infanterie abgebrochen würde, es hätte meiner Meinung nach aber trotzdem erwähnt werden müssen. Hat die Infanterie das Verhaltensmodell None, passiert nämlich gar nichts.

Die Einstellungen für AI_GRP1 und AI_GRP2 werden in diesem Verhaltensmodell nicht verwendet.

Verhaltensoptionen

Es stehen die gleichen Optionen zur Auswahl wie bei Trucks – Move to location. Was außerdem die Option Repair railway an dieser Stelle soll, wird mir wohl immer schleierhaft bleiben.

6.5 Verhaltensmodelle für andere Einheitentypen

6.5.1 Das Verhaltensmodell Maintenance – Repair

Das Verhaltensmodell Maintenance – Repair regelt das Verhalten von Versorgungsfahrzeugen (Versorgungs-Lkw, Tankwagen). Normalerweise versorgen bzw. reparieren Versorgungsfahrzeuge alle freundlich gesinnten Einheiten in Sichtweite bzw. in der angegebenen Zone.

Einstellungen

Die Versorgungsfahrzeuge der Gruppe fahren zu der als AI_LOC1 angegebenen Zone, wenn an ihrem ursprünglichen Standort keine zu versorgenden Einheiten zu sehen sind. Die KI-Zone ist eigentlich nur in dem Fall von Bedeutung, dass eine Brücke repariert werden soll. Dies ist nur möglich, wenn die Brücke in der Zone AI_LOC2 liegt. Damit sich die Versorgungseinheiten dorthin begeben, muss man also für beide KI-Zonen die Zone eintragen, in der die Brücke liegt (eine genauere Beschreibung folgt weiter unten).

Die Einstellungen für AI_GRP1 und AI_GRP2 werden in diesem Verhaltensmodell nicht verwendet.

Verhaltensoptionen

Die folgenden Optionen stehen zur Auswahl:

- Randomly move around location: Die Versorgungsfahrzeuge stehen nicht einfach herum, sondern bewegen sich von Zeit zu Zeit zu einer anderen Stelle in der angegebenen Zone. Das kann bei größeren Zonen sinnvoll sein, um die Wahrscheinlichkeit zu erhöhen, dass zu versorgende Einheiten in der Zone in Sichtweite geraten.
- Move to location only if there are friendly troops: Bei dieser Option bewegen sich die Versorgungsfahrzeuge nur dann in die angegebene Zone, wenn sich dort bereits freundlich gesinnte Einheiten befinden.
- Move to location only if there are no enemy troops: Die Versorgungsfahrzeuge bewegen sich nur zu der angegebenen Zone, wenn diese feindfrei ist.
- Repair only inside the location: Es werden nur in den beiden KI-Zonen befindliche Einheiten versorgt. Ist diese Option nicht aktiviert, werden alle freundlich gesinnten Einheiten in Sichtweite versorgt.
- Repair objects: Wenn diese Option aktiviert wird, dann reparieren die Versorgungsfahrzeuge auch Eisenbahnlinien und Flugplätze in den beiden KI-Zonen. Brücken und Gebäude werden nur in Zone AI_LOC2 repariert. Dazu muss außerdem noch ein Marker auf der Brücke bzw. auf dem Gebäude angebracht werden. So kann man z.B. dafür sorgen, dass der Gegner Fabrikanlagen und Krankenhäuser auch bei Beschuss durch den Spieler instand halten kann.

- Build pontoons: Die Versorgungsfahrzeuge bewegen sich zu der angegebenen Zone und bauen eine *Pontonbrücke*. Dazu muss die Zone, in der die Pontonbrücke gebaut werden soll, als AI_LOC1 und AI_LOC2 angegeben werden. Der Punkt, an dem die Brücke errichtet werden soll, muss in dieser Zone liegen und mit einem Marker gekennzeichnet werden. Natürlich muss diese Stelle für das Versorgungsfahrzeug erreichbar sein und auch wirklich den Bau einer Brücke erlauben (gerade Ufer, vgl. 3.6.5). Die korrekte Platzierung des Markers zeigt die folgende Abbildung.

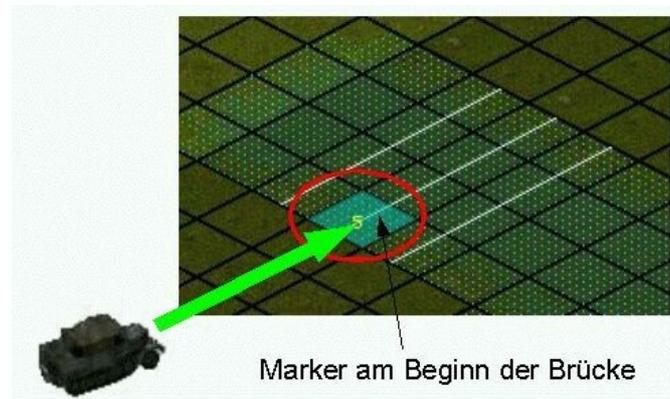


Abb. 57: Bau von Pontonbrücken durch KI-gesteuerte Einheiten (Plazierung des Markers)

Zu den drei letzten Optionen ist eine Anmerkung notwendig. Wenn man das Verhaltensmodell Maintenance – Repair für eine Gruppe zum ersten Mal einstellt, dann werden alle Optionen angezeigt. Will man später eine Änderung vornehmen, dann werden die letzten drei Optionen nicht mehr angezeigt, so dass man auf diese Optionen nicht mehr zugreifen kann. Die Einstellungen bleiben aber erhalten, auch wenn sie nicht mehr sichtbar sind. Das gleiche Problem tritt übrigens auch im Skripteditor auf, wenn man das KI-Verhalten setzen will. Den Effekt sieht man in der folgenden Abbildung.

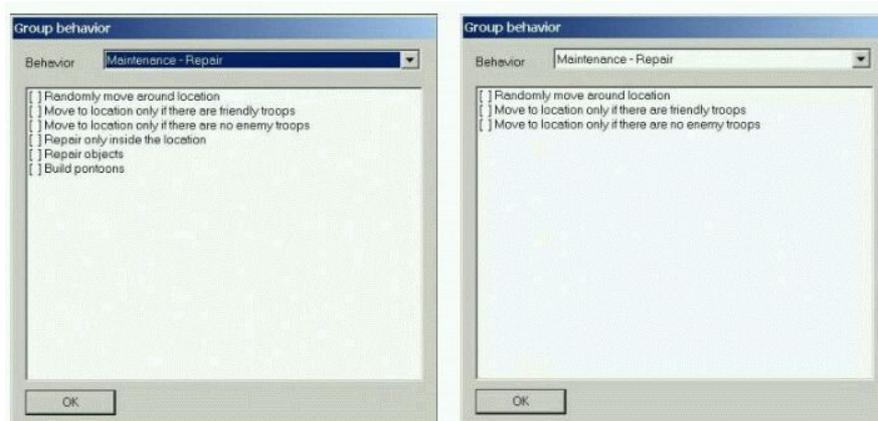


Abb. 58: Bug beim Einstellen der Optionen zum Verhaltensmodell Maintenance – Repair

Will man an den drei nicht angezeigten Optionen Änderungen vornehmen, kann man in dem Auswahlfeld einmal ein anderes Modell und direkt im Anschluss wieder Maintenance – Repair auswählen. Dann sind wieder alle Optionen sichtbar. Bei diesem Vorgehen muss man allerdings alle Optionen für die aktuelle Gruppe neu einstellen und bei der nächsten Anzeige verschwinden die letzten drei Optionen natürlich wieder. Lästig, aber glücklicherweise nicht weiter problematisch.

6.5.2 Das Verhaltensmodell Katyusha – Move to location

Das Verhaltensmodell Katyusha – Move to location entspricht dem Verhaltensmodell Trucks – Move to location, bezieht sich aber auf Raketenwerfer. Die Einstellungen und Optionen sind die gleichen.

In meiner Version des Editors tritt hier übrigens ein weiterer Bug auf. Es werden außer beim ersten Mal bei jedem Anzeigen der Optionen fälschlicherweise die Optionen für Versorgungsfahrzeuge angezeigt.

6.5.3 Das Verhaltensmodell **Planes – Move to location**

Auch für Flugzeuge gibt es ein Verhaltensmodell, das dafür sorgt, dass ein Flugzeug sich zu einer Zone begibt. Dort angekommen führt es den von der Art des Flugzeugs (Aufklärer, Jäger, Bomber, Transporter) abhängigen Auftrag aus und landet schließlich wieder auf dem Flugplatz.

Einstellungen

Die Einstellungen entsprechen denen des Verhaltensmodells Trucks – Move to location.

Verhaltensoptionen

Die folgenden Optionen stehen zur Auswahl:

- Fly to location only if there are friendly troops: Bei dieser Option bewegen sich die Flugzeuge nur dann in die angegebene Zone, wenn sich dort bereits freundlich gesinnte Einheiten befinden.
Für Bomber gilt eine modifizierte Option. Die Zone wird nur dann bombardiert, wenn **keine** eigenen Einheiten dort sind.
- Fly to location only if there are no enemy troops: Die Flugzeuge bewegen sich nur zu der angegebenen Zone, wenn diese feindfrei ist.
Auch hier hat die Option für Bomber eine andere Bedeutung. Die Zone wird nur dann bombardiert, wenn dort feindliche Einheiten **vorhanden** sind.
- Transports – Fly when more than 70% loaded: Transportflugzeuge fliegen erst dann ab, wenn sie zu mindestens 70% beladen sind.

Auch bei diesem Verhaltensmodell kommt es in meiner Version des Editors zu Problemen bei der Anzeige. Die hier genannten Optionen werden nur beim ersten Mal angezeigt. Bei späterer Anzeige sieht man die Optionen des Verhaltensmodells Trucks – Move to location. Da sich jedoch nur die Darstellung der Optionen ändert, kann man trotzdem Änderungen vornehmen.

6.5.4 Das Verhaltensmodell **Trains – Defend location**

Das Verhaltensmodell Trains – Defend location wird verwendet, wenn ein Zug einen bestimmten Ort verteidigen soll. Der Zug bewegt sich zu der angegebenen Zone und verteidigt diese gegen feindliche Einheiten.

Einstellungen

Die Zone, die durch den Zug verteidigt werden soll, wird als AI_LOC1 angegeben. Bei Verwendung der Option Repair railway wird der Zug zunächst die Schienen in der Zone AI_LOC2 reparieren und sich dann in die Zone AI_LOC1 begeben.

Die Einstellungen für AI_GRP1 und AI_GRP2 werden in diesem Verhaltensmodell nicht verwendet.

Verhaltensoptionen

Die folgenden Optionen stehen zur Auswahl:

- Randomly move around location: Der Zug bleibt nicht an einer Stelle stehen, sondern bewegt sich von Zeit zu Zeit zu einer anderen Stelle in der angegebenen Zone.
- Move to location only if there are friendly troops: Bei dieser Option fährt der Zug nur dann in die angegebene Zone, wenn sich dort bereits freundlich gesinnte Einheiten befinden.
- Move to location only if there are no enemy troops: Der Zug fährt nur dann zu der angegebenen Zone, wenn diese feindfrei ist.
- Repair railway: Sind in der als AI_LOC2 eingetragenen Zone die Schienen beschädigt, dann werden sie von der Besatzung repariert. Ist die Option deaktiviert, bleibt der Zug an der Beschädigung stehen, auch wenn die Zielzone hinter der Beschädigung liegt.

Auch bei diesem Verhaltensmodell kommt es in meiner Version des Editors zu Problemen bei der Anzeige. Die hier genannten Optionen werden nur beim ersten Mal angezeigt. Bei späterer Anzeige sieht man die Optionen des Verhaltensmodells Planes – Move to location.

7 Skriptprogrammierung

In diesem Kapitel wird die *Skriptprogrammierung* ausführlich beschrieben. Da die künstliche Intelligenz der Einheiten in SSRW (wie schon in den Vorgängerversionen) auf relativ niedriger Ebene angesiedelt ist, wird einigermaßen komplexes Verhalten der vom Computer gesteuerten Einheiten nur dadurch erreicht, dass man dieses Verhalten in dem *Skript* einer Mission festlegt. Das Skript einer Mission besteht dabei aus einer mehr oder weniger umfangreichen Menge von Teilskripten, die in dem mitgelieferten Handbuch zum Editor auch als *Trigger* bezeichnet werden. Ich werde allerdings im folgenden auch die Trigger als Skripte bezeichnen³⁰.

Ein Skript in SSRW besteht aus zwei Teilen, einer *Bedingung* und einer Reihe von *Aktionen*. Die Bedingung legt die Umstände fest, unter denen die Aktionen ausgeführt werden. Ein Skript kann man im Prinzip als Wenn-dann-Satz wiedergeben, z.B. „**Wenn** alle feindlichen Einheiten vernichtet wurden, **dann** wird für den Spieler eine Meldung ausgegeben und die Mission mit einem Sieg beendet.“

Der erste Abschnitt dieses Kapitels beschreibt den Aufbau von Bedingungen. Dazu werden auch Grundlagen der Booleschen Logik erläutert. In den weiteren Abschnitten werden die in SSRW zur Verfügung stehenden Bedingungen und Aktionen thematisch geordnet behandelt. Dabei wird das jeweilige Thema zunächst allgemein beschrieben. Daran schließen sich dann die Erläuterungen zu den entsprechenden Aktionen und Bedingungen an.

Die farbliche Darstellung der Bedingungen und Aktionen in diesem Kapitel entspricht dabei der Darstellung in den Auswahllisten des Skripteditors und soll die Orientierung im Skripteditor erleichtern. Ich habe jeweils die Bezeichnungen aus dem Bedingungs- bzw. Aktionsfenster des Skripteditors angegeben. In einigen Fällen weichen die Bezeichnungen von Bedingungen und Aktionen in der Auswahlliste, wohl aus Gründen der Lesbarkeit, davon ab. Bei besonders starken Abweichungen habe ich auch die Schreibweise aus der Auswahlliste angegeben. Da einige der Bedingungen und Aktionen vom Missionsentwickler um weitere Angaben ergänzt werden müssen, habe ich diese in die Erläuterungen mit aufgenommen. Eine Übersicht der dabei verwendeten Schreibweisen und Abkürzungen findet man am Ende des Kapitels.

Da ich noch nicht alle Bedingungen und Aktionen ausprobiert habe, kann es sein, dass die eine oder andere Erklärung ungenau, unvollständig oder schlicht falsch ist. In solchen Fällen bitte ich a) um Nachsicht und b) um Nachricht.

7.1 Bedingungen und Boolesche Logik

Die Verwendung von Bedingungen beruht auf der *Booleschen Logik*³¹, die auch in fast jeder Programmiersprache verwendet wird und auf der letztlich auch die Funktionsweise des Computers beruht. Eine Bedingung kann entweder erfüllt (*wahr*) oder nicht erfüllt sein (*falsch*). Um es an dem oben angeführten Beispiel zu erläutern: Die Bedingung „alle feindlichen Einheiten vernichtet“ ist erfüllt, wenn tatsächlich keine feindlichen Einheiten mehr vorhanden sind. Solange auch nur eine einzige feindliche Einheit existiert, ist die Bedingung nicht erfüllt.

7.1.1 Der Operator not

Die Bedingung aus dem obigen Beispiel (und natürlich jede andere Bedingung) kann man durch die Verwendung des *Operators* not in ihr Gegenteil verkehren. Wenn z.B. noch eine feindliche Einheit existiert, dann ist die Bedingung „alle feindlichen Einheiten vernichtet“ – wie bereits erwähnt – nicht erfüllt. Die verneinte Bedingung „not alle feindlichen Einheiten vernichtet“ ist dagegen erfüllt. Indem man also den Operator not vor eine Bedingung setzt, dreht sich das Ergebnis der Bedingung um. Aus wahr wird falsch, aus falsch wird wahr (vgl. Tabelle 3).

³⁰ Da ein Teilskript neben einer Bedingung auch Aktionen beschreibt, fällt es mir schwer, die Teilskripte als Trigger zu bezeichnen. Mit dem Begriff Trigger, also Auslöser oder Abzug, verbinde ich eher etwas, das die Aktionen auslöst, also die Bedingungen, unter denen eine Aktion ausgeführt wird.

³¹ nach dem britischen Mathematiker George Boole (1815-1864)

B	not B
wahr	falsch
falsch	wahr

Tabelle 3: Wirkung des Operators not (B steht für eine Bedingung)

7.1.2 Die Operatoren and und or

Eine Bedingung kann auch aus mehreren Teilbedingungen bestehen. Diese Teilbedingungen müssen dann zueinander in Beziehung gesetzt werden. Dafür gibt es in der Booleschen Logik eine ganze Reihe von Möglichkeiten, in SSRW (wie auch in vielen Programmiersprachen) begnügt man sich mit zwei davon.

Der Operator and

Eine Bedingung mit mehreren Teilbedingungen kann z.B. erfüllt sein, wenn jede der Teilbedingungen erfüllt ist. Das wird durch den *Operator* and beschrieben, durch den man die Teilbedingungen miteinander verknüpft. So könnte man z.B. die weiter oben bereits als Beispiel verwendete Siegbedingung erweitern zu „alle feindlichen Einheiten vernichtet and mindestens eine Einheit von Gruppe A1 in Zone 15“.

Bei Verwendung dieser Bedingung hätte der Spieler noch nicht gewonnen, wenn er jede einzelne feindliche Einheit vernichtet hätte. Zusätzlich müsste wegen der zweiten Teilbedingung mindestens eine der Einheiten aus der Gruppe A1 die Zone mit der Nummer 15 erreichen, um die Mission siegreich zu beenden. Umgekehrt reicht es für einen Sieg auch nicht aus, wenn eine Einheit von Gruppe A1 die Zone 15 erreicht, aber sich noch irgendwo feindliche Einheiten in der Missionszone aufhalten.

Erst wenn beide Teilbedingungen erfüllt sind, also „alle feindlichen Einheiten vernichtet“ **und** „mindestens eine Einheit von Gruppe A1 in Zone 15“, ist die Mission siegreich abgeschlossen. Der and-Operator funktioniert also im Prinzip so wie das „und“ in unserer Umgangssprache.

Die folgende Tabelle zeigt die vier Möglichkeiten, die bei einer Bedingung mit zwei Teilbedingungen (wie in dem hier verwendeten Beispiel) auftreten können. Eine Bedingung kann aber auch aus mehr als zwei Teilbedingungen bestehen. Wenn alle Teilbedingungen mit dem and-Operator verknüpft sind, dann muss jede der Teilbedingungen erfüllt sein, damit die gesamte Bedingung ebenfalls erfüllt ist.

B₁	B₂	B₁ and B₂
falsch	falsch	falsch
falsch	wahr	falsch
wahr	falsch	falsch
wahr	wahr	wahr

Tabelle 4: Wirkung des Operators and: Die Teilbedingungen B₁, B₂ werden mit and zu der Bedingung B₁ and B₂ verknüpft. Nur wenn beide Teilbedingungen wahr sind, ist auch die Verknüpfung wahr.

Der Operator or

Wie erwähnt gibt es noch eine andere Möglichkeit, mehrere Teilbedingungen miteinander zu verknüpfen, nämlich den *Operator* or. Dieser Operator funktioniert wie das „oder“ in unserer Umgangssprache (vorausgesetzt, dass mit „oder“ nicht „entweder – oder“ gemeint ist).

Bei Verwendung dieses Operators würde unser Beispiel für die Siegbedingung so lauten: „alle feindlichen Einheiten vernichtet or mindestens eine Einheit von Gruppe A1 in Zone 15“. Für einen Sieg würde es in diesem Fall schon ausreichen, wenn eine der beiden Teilbedingungen erfüllt wäre. Natürlich dürfen auch beide Teilbedingungen erfüllt sein. Nur wenn keine der Teilbedingungen erfüllt ist, kann sich der Spieler noch nicht zufrieden zurücklehnen. Der Spieler hat also gewonnen, wenn „alle

feindlichen Einheiten vernichtet“ sind **oder** „mindestens eine Einheit von Gruppe A1 in Zone 15“ steht **oder** sogar beides erfüllt ist.

Auch hier können wie schon bei der and-Verknüpfung vier Möglichkeiten auftreten, die in der folgenden Tabelle aufgelistet sind. Besteht die Bedingung aus mehr als zwei Teilbedingungen, die alle mit dem **or**-Operator verknüpft sind, dann muss mindestens eine der Teilbedingungen erfüllt sein, damit die gesamte Bedingung ebenfalls erfüllt ist.

B₁	B₂	B₁ or B₂
falsch	falsch	falsch
falsch	wahr	wahr
wahr	falsch	wahr
wahr	wahr	wahr

Tabelle 5: Wirkung des Operators or: Die Teilbedingungen B₁, B₂ werden mit or zu der Bedingung B₁ or B₂ verknüpft. Wenn eine der Teilbedingungen wahr ist, ist auch die Verknüpfung wahr.

7.1.3 Klammern in Bedingungen

Anders als in den bisherigen Beispielen kann man in einer Bedingung auch mehrere der drei Booleschen Operatoren not, and und or gleichzeitig verwenden. Dabei kann es aber zu Missverständnissen kommen. So ist z.B. nicht eindeutig, was mit der folgenden Bedingung gemeint ist: not „mehr als 3 Einheiten in Gruppe A1“ and „Gruppe A2 vernichtet“. Soll sich der not-Operator nur auf die erste der Teilbedingungen beziehen oder auf das Ergebnis der and-Verknüpfung? Das hängt davon ab, welchen der beiden Operatoren man zuerst betrachtet. Normalerweise wird zuerst der Operator not, dann and und zum Schluss or angewendet.

Leider ist im mitgelieferten Handbuch nicht angegeben, ob sich SSRW an diese Konvention hält und ich war, ehrlich gesagt, zu faul, das für alle denkbaren Kombinationen der drei Operatoren auszutesten. Man kann aber die Reihenfolge der Auswertung durch die Verwendung von *Klammern* festlegen. Was in Klammern zusammengefasst wird, wird zuerst ausgewertet. Bei mehreren ineinander geschachtelten Klammerebenen arbeitet man sich von innen nach außen.

Würde man in der obigen Bedingung die Klammern so setzen, dass man die Bedingung not („mehr als 3 Einheiten in Gruppe A1“ and „Gruppe A2 vernichtet“) erhält, dann wäre zuerst der and-Operator anzuwenden. Die Bedingung in den Klammern wäre erfüllt, wenn beide Teilbedingungen erfüllt sind. Die gesamte Bedingung wäre wegen des not-Operators also dann erfüllt, wenn das Gegenteil zutrifft, d.h. wenn keine der Bedingungen erfüllt ist.

Indem man die Klammern anders setzt, kann man natürlich auch die Bedingung (not „mehr als 3 Einheiten in Gruppe A1“) and „Gruppe A2 vernichtet“ bekommen. Diese Bedingung ist dann erfüllt, wenn die erste Teilbedingung, also „mehr als 3 Einheiten in Gruppe A1“, nicht erfüllt, die zweite Teilbedingung „Gruppe A2 vernichtet“ aber erfüllt ist.

7.2 Einmalige und wiederholte Ausführung von Skripten

Während des Spiels werden die Bedingungen aller in der Mission vorhandenen Skripte ständig überwacht. Im Handbuch ist die Rede von mehreren Malen pro Sekunde. Sobald bei dieser Überwachung festgestellt wird, dass die Bedingung eines Skripts erfüllt ist, werden die in dem Skript angegebenen Aktionen ausgeführt. Ist die gleiche Bedingung bei der nächsten Überprüfung immer noch erfüllt, werden die Aktionen des Skripts erneut ausgeführt. Diese Wiederholung der Ausführung erfolgt, bis die Bedingung nicht mehr erfüllt ist.

Das ist in manchen Fällen nützlich, kommt aber nicht so häufig vor und kann auch zu Problemen führen, wie man in Abb. 59 sehen kann. In dem dargestellten Fall wurde eine Verstärkung per Skript geschickt. Die Bedingung dieses Skripts war aber dauerhaft erfüllt, so dass das Skript immer wieder ausgeführt wurde. Dadurch wurden immer neue Verstärkungen auf die Karte geschickt.



Abb. 59: Beispiel für ein Problem aufgrund wiederholter Ausführung eines Skripts

In den meisten Fällen soll ein Skript jedoch nur einmal ausgeführt werden. Das erreicht man, indem man den Aktionen des Skripts die Aktion `exclude this condition` hinzufügt. Diese Aktion sorgt dafür, dass die Bedingung des Skripts bei der nächsten Überprüfung nicht mehr berücksichtigt wird. Das bedeutet aber auch, dass alle Skripte, die von der gleichen Bedingung ausgelöst werden, ebenfalls nicht mehr überprüft werden. Wenn man also mehrere Skripte mit gleichen Bedingungen erstellt, sollte man wissen, was man tut. Erfahrungsgemäß sind Fehler aufgrund solcher Konstellationen (sog. Seiteneffekte) relativ schwer zu finden.

Die Aktion `exclude this condition` kann prinzipiell an jeder Stelle des Skripts stehen. Ich habe mir jedoch angewöhnt, diese Aktion an den Anfang des Skripts zu stellen, damit man auch bei längeren Skripten sofort und ohne im Skripteditor zu scrollen sehen kann, ob das Skript einmal oder mehrmals ausgeführt wird.

7.3 Informationen für den Spieler

In vielen Situationen ist es notwendig, dem Spieler Informationen über den Ablauf der Mission zukommen zu lassen. Dafür gibt es unterschiedliche Möglichkeiten, die jeder im Spiel schon mal erlebt haben dürfte.

7.3.1 Textausgaben

Die einfachste Möglichkeit zur Vermittlung von Informationen besteht darin, einen Text auf dem Bildschirm auszugeben. Dabei wird auf die *Missionstexte* (Phrases) zurückgegriffen, deren Erstellung in 5.4 beschrieben wurde. Um einen dieser Missionstexte auszugeben, verwendet man die Aktion `say phrase <<Missionstext>>`. Der Missionstext erscheint dann im linken unteren Teil des Bildschirms (vgl. Abb. 60).



Abb. 60: Textausgabe mit `say phrase`

Geht es bei der Textausgabe um Informationen zu einer bestimmten Stelle auf der Karte, kann man den Spieler zusätzlich auf diese Stelle hinweisen. Dazu setzt man an die gewünschte Stelle auf der Karte einen Marker (vgl. 5.3.2). Für die Textausgabe verwendet man in diesem Fall die Aktion `say phrase <<Missionstext>> with pointer to object <<#Marker>>`. Im Spiel erscheint dann bei der Ausgabe des Missionstextes zusätzlich ein kleiner werdender grüner Kreis in der Minimap, dessen Mittel-

punkt die durch den Marker bezeichnete Stelle ist (vgl. Abb. 61). Dadurch erhält der Spieler einen Hinweis auf die Stelle, auf die sich die Meldung bezieht.



Abb. 61: Textausgabe mit say phrase with pointer to object

7.3.2 Dialogboxen

Die im letzten Abschnitt beschriebene Textausgabe hat einen Nachteil, der bei wichtigen Informationen nicht zu unterschätzen ist: Der Text wird nach einiger Zeit wieder ausgeblendet. Es besteht daher die Möglichkeit, dass der Spieler diese Textmitteilung aufgrund von Aktionen im Spiel übersieht. Informationen, die für den Spieler von besonderer Bedeutung sind, sollte man daher nicht als reinen Text ausgeben.

Stattdessen verwendet man die Aktion `show modal dialog <<Missionstext>>` für die Anzeige der Information. Der Text wird dann in einer *Dialogbox* angezeigt (vgl. Abb. 62). Das Spiel wird dabei so lange unterbrochen, bis der Spieler die Dialogbox durch einen Mausklick auf OK schließt. Auf diese Weise ist sichergestellt, dass er die Information nicht übersieht.



Abb. 62: Textausgabe in einer Dialogbox

Auch bei der Verwendung von Dialogboxen kann man den Spieler mit Hilfe eines Markers auf eine Stelle der Karte hinweisen. Die entsprechende Aktion lautet `show modal dialog <<Missionstext>> with pointer to object <<#Marker>>`.

7.3.3 Anzeige bestimmter Kartenbereiche

Es gibt Situationen, in denen der oben beschriebene Hinweis auf eine Stelle der Karte nicht ausreicht. Wenn der Spieler z.B. auf ein einzelnes Gebäude hingewiesen werden soll, ist die Darstellung der Minimap zu ungenau. Einzelne Gebäude kann man auf ihr im Normalfall nur schlecht oder gar nicht erkennen. Außerdem verschwindet der in der Minimap eingeblendete grüne Kreis relativ schnell

wieder, so dass die Stelle auf der Karte nicht genau identifiziert werden kann. Im schlimmsten Fall übersieht der Spieler den Hinweis.

In solchen Fällen kann man den angezeigten Kartenausschnitt auch an die entsprechende Stelle verschieben, wenn man dort eine Zone (Location) angelegt hat (vgl. 5.3.1). Dazu verwendet man die Aktion `move screen to location <<#Zone>>`. Der Kartenausschnitt wird dann so verschoben, dass die Mitte der Zone in der Bildschirmmitte liegt.

Wenn die Einheiten des Spielers sich noch nicht in Sichtweite der angegebenen Stelle befinden, bleibt die Zone vom Kriegsnebel verdeckt. Manchmal ist dies aber nicht erwünscht. In solchen Fällen kann man den *Kriegsnebel* an der entsprechenden Stelle aufheben, so dass alles an der Stelle sichtbar wird. Dazu platziert man einen Marker an der Stelle, die sichtbar werden soll, und fügt die Aktion `show object <<#Marker>> in radius <<#>> for player <<Spielername>>` in das Skript ein.

Der Radius gibt dabei an, in welchem Umkreis um den angegebenen Marker der Kriegsnebel aufgehoben werden soll. Bei der Verwendung dieser Aktion ist zu beachten, dass die Wirkung nur bis zur nächsten Überprüfung der Bedingungen aller Skripte der Mission anhält. Will man also eine Stelle der Karte länger sichtbar lassen, dann muss man dafür sorgen, dass die auslösende Bedingung entsprechend lange erfüllt bleibt. Ein Beispiel dafür zeigt die folgende Abbildung.

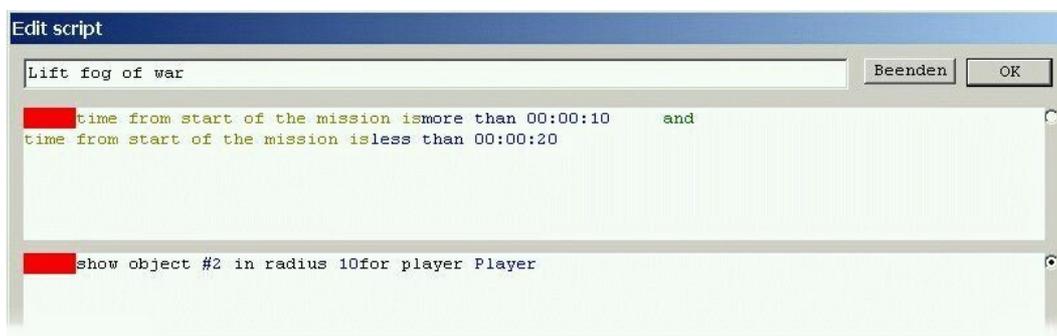


Abb. 63: Aufheben des Kriegsnebels für zehn Sekunden

7.4 Zeitgesteuerte Skripte

Viele Abläufe während einer Mission sind in irgendeiner Form zeitabhängig. Zur Realisierung solcher Abläufe bietet SSRW eine ganze Reihe von unterschiedlichen Möglichkeiten.

7.4.1 Abhängigkeit vom Zeitpunkt des Missionsbeginns

Soll ein Skript bei Beginn der Mission ausgeführt werden, fragt man die Bedingung `mission start` ab. Ein Skript, das bei Erfüllung dieser Bedingung ausgeführt wird, sollte die Aktion `exclude this condition` enthalten, da die Bedingung im Normalfall nur einmal während der gesamten Missionsdauer erfüllt sein kann³². Man kann so die Skript-Engine von SSRW von der Überprüfung einer überflüssigen Bedingung entlasten.

Man kann die Skriptausführung auch von der seit Missionsbeginn verstrichenen Zeit abhängig machen, indem man die Bedingung `time from start of the mission is <<VglOp>> <<Zeit>>` verwendet. Ein Beispiel ist in Abb. 63 zu sehen.

7.4.2 Timer

Der Bezug auf den Zeitpunkt des Missionsbeginns ist nicht die einzige Möglichkeit zur zeitgesteuerten Ausführung von Skripten. SSRW stellt 100 sogenannte *Timer* zur Verfügung, die von 0 bis 99 durchnummeriert sind. Ein Timer funktioniert im Prinzip wie ein Countdown. Er wird auf eine bestimmte Zeitdauer eingestellt und gestartet. Ist die eingestellte Zeit verstrichen, wird ein Signal ausgelöst. Jeder der 100 Timer kann sowohl als *einfacher* wie auch als *periodischer Timer* verwendet werden.

³² Eine Ausnahme bilden zusammengesetzte Bedingungen, in denen die Bedingung `mission start` durch `or` mit anderen Teilbedingungen verknüpft wird. Solche Bedingungen können auch im weiteren Verlauf noch erfüllt sein.

Einfache Timer

In vielen Fällen soll eine Aktion ausgeführt werden, sobald nach dem Eintreten einer bestimmten Bedingung eine gewisse Zeit vergangen ist. In diesen Fällen setzt man einen einfachen Timer ein, der durch die Aktion `start timer <<#Timer>> to <<Zeit>>` gestartet wird. Ist die angegebene Zeitspanne vergangen, dann ist die Bedingung `timer <<#Timer>> elapsed` erfüllt. Sie wird solange als erfüllt betrachtet, bis der Timer erneut gestartet wird. Aus diesem Grund muss ein Skript, das von dieser Bedingung ausgelöst wird, mit der Aktion `exclude this condition` versehen werden, wenn es nur einmal ausgeführt werden soll.

Ist der Timer gestartet, kann er ggf. mit der Aktion `stop timer <<#Timer>>` angehalten werden. Dadurch wird verhindert, dass die Bedingung `timer <<#Timer>> elapsed` erfüllt wird.

Periodische Timer

Soll eine Aktion in regelmäßigen Abständen, z.B. alle fünf Minuten, ausgeführt werden, kann man das ebenfalls mit Hilfe eines Timers erreichen. Dazu führt man vor dem Starten des Timers die Aktion `set period of timer <<#Timer>> into <<Zeit>>` aus. Sobald der Timer anschließend gestartet wird, wird immer nach der angegebenen Zeit für einen Moment die Bedingung `timer <<#Timer>> elapsed` erfüllt. Danach beginnt der Timer wieder bei dem Startwert und die Bedingung ist nicht mehr erfüllt.

7.4.3 Countdown

In Situationen, in denen der Spieler ein bestimmtes Ziel in begrenzter Zeit erreichen oder eine Stellung für einen bestimmten Zeitraum halten muss, kann man den *Countdown-Timer* von SSRW verwenden. Es handelt sich dabei um einen speziellen Timer, der durch die Aktion `start countdown from <<Zeit>>` gestartet wird. In der oberen linken Ecke des Bildschirms wird dann – beginnend mit der angegebenen Zeitdauer – die rückwärts laufende Restzeit eingeblendet (vgl. Abb. 64). Auf diese Weise ist der Spieler immer darüber informiert, wie viel Zeit ihm für die Erfüllung seiner Aufgabe noch bleibt. Die Anzeige bleibt nach Ablauf des Countdowns sichtbar.

Der Countdown kann über die Bedingung `countdown <<VglOp>> <<Zeit>>` abgefragt werden.

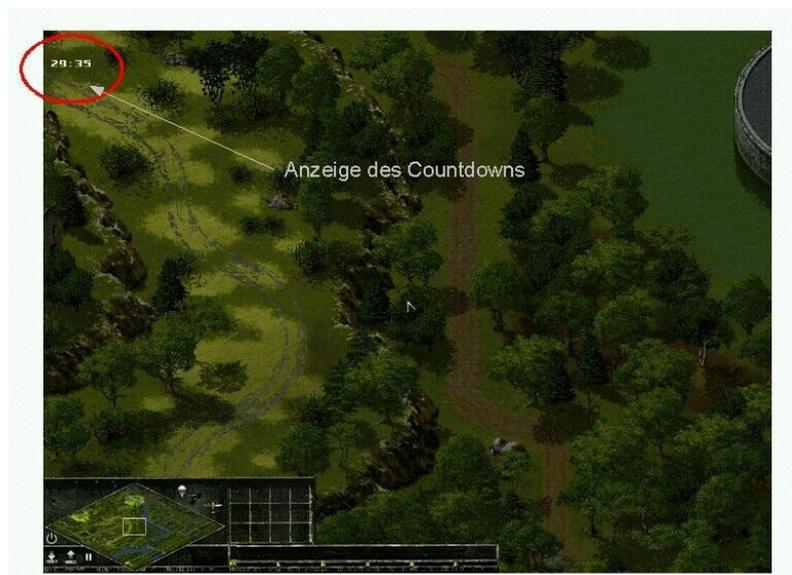


Abb. 64: Anzeige des Countdowns während der Mission

7.5 Skriptsteuerung von Einheiten

Häufig hängt die Ausführung einer Aktion davon ab, in welchem Zustand bestimmte Einheiten sind oder an welchem Ort sie sich befinden. Einheiten können auch mit Hilfe von Skripten manipuliert oder zu bestimmten Aktionen veranlasst werden.

7.5.1 Aufenthaltsort von Einheiten

Manche Aktionen sollen ausgeführt werden, wenn eine bestimmte Stelle auf der Karte von Einheiten einer Gruppe oder eines Teams erreicht, besetzt oder verlassen wird. Um solche Situationen erkennen zu können, gibt es die folgenden Bedingungen.

Durch die Bedingung `<<VglOp>> <<#>> units of group <<#Gruppe>> in location <<Zone>>` wird überprüft, wie viele Einheiten der angegebenen Gruppe sich in der angegebenen Zone aufhalten. Statt der absoluten Anzahl der Einheiten in der angegebenen Zone kann man auch den prozentualen Anteil der in der Zone stehenden Einheiten an der Größe der Gruppe abfragen. Dazu dient die Bedingung `<<VglOp>> <<#>> % of units of group <<#Gruppe>> in location <<Zone>>`.

Varianten dieser beiden Bedingungen gibt es auch für den Fall, dass nicht die Einheiten einer bestimmten Gruppe, sondern die eines Teams von Interesse sind. In diesem Fall lauten die Bedingungen `<<VglOp>> <<#>> units of Player <<Spielername>> in location <<Zone>>` bzw. `<<VglOp>> <<#>> % of units of Player <<Spielername>> in location <<Zone>>`.

7.5.2 Anzahl von Einheiten

In vielen Fällen ist es für die Ausführung von Aktionen wichtig zu wissen, wie viele Einheiten einer Gruppe oder eines Teams insgesamt vorhanden sind. Das gilt z.B. für die Siegbedingung „alle feindlichen Einheiten vernichtet“.

Dabei wird zwischen verschiedenen Situationen unterschieden, in denen eine Einheit sich befinden kann. *Aktive* (*active*) Einheiten befinden sich in der Missionszone, während *unterwegs* befindliche (*on the way*) Einheiten als Verstärkung auf die Karte geschickt wurden, die Missionszone aber noch nicht erreicht haben. Besetzte und unbesetzte Geschütze (Guns) werden als eigenständiger Fall gewertet. Zuletzt gibt es noch den Fall, dass Einheiten bereits getötet oder zerstört wurden, aber noch nicht ihre maximale Anzahl an „Wiederbelebungen“ (vgl. 4.3) erreicht haben. Diese befinden sich in einem Zustand, der als „*in hell*“ („in der Hölle“) bezeichnet wird. Diese vier Fälle und ihre möglichen Kombinationen werden als *Einheitenklasse* bezeichnet. Sie werden in den folgenden Bedingungen verwendet.

Die Abb. 65 zeigt den Dialog, der im Skripteditor zum Einstellen der Einheitenklasse verwendet wird. Durch Setzen eines oder mehrerer Häkchen kann man jede Kombination der vier grundlegenden Einheitenklassen festlegen.

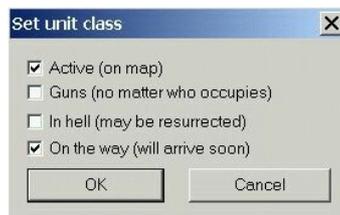


Abb. 65: Dialog zum Einstellen der Einheitenklasse

Durch die Bedingung `<<EinhK1>> units in group <<#Gruppe>> <<VglOp>> <<#>>` wird überprüft, wie viele Einheiten der angegebenen Einheitenklasse zu der angegebenen Gruppe gehören. Entsprechend dient die Bedingung `<<EinhK1>> units of player <<Spielername>> <<VglOp>> <<#>>` zur Überwachung der Anzahl der Einheiten eines bestimmten Teams. Eine Kombination beider Bedingungen steht ebenfalls zur Verfügung: `<<EinhK1>> units of player <<Spielername>> in group <<#Gruppe>> <<VglOp>> <<#>>`. Damit lässt sich die Anzahl der Einheiten eines Teams innerhalb einer bestimmten Gruppe überwachen.

Auch Vergleiche zwischen zwei Gruppen bzw. zwei Teams sind möglich. Dazu dienen die beiden folgenden Bedingungen. Mit `<<EinhK1>> units in group <<#Gruppe1>> <<VglOp>> <<#>> % of <<EinhK2>> units in group <<#Gruppe2>>` kann man zwei Gruppen miteinander vergleichen, während der Vergleich von zwei Teams mit Hilfe der Bedingung `<<EinhK1>> units of player <<Spielername1>> <<VglOp>> <<#>> % of <<EinhK2>> units of player <<Spielername2>>` realisiert wird. Anzumerken ist, dass diese Bedingung in der Auswahlliste des Skripteditors nicht ganz vollständig angegeben ist. Sie ist dort als `units of player in per cents` aufgeführt. Korrekt wäre analog

zu der Bedingung für Gruppen units of player in per cents of units of player. Im Bereich für Bedingungen wird dann aber die korrekte Schreibweise angezeigt. Ein Anwendungsbeispiel zeigt die folgende Abbildung: Hat der Spieler mit allen Verstärkungen, die unterwegs sind, weniger als halb so viele Einheiten, wie der Gegner auf der Karte hat, wird eine Meldung ausgegeben.

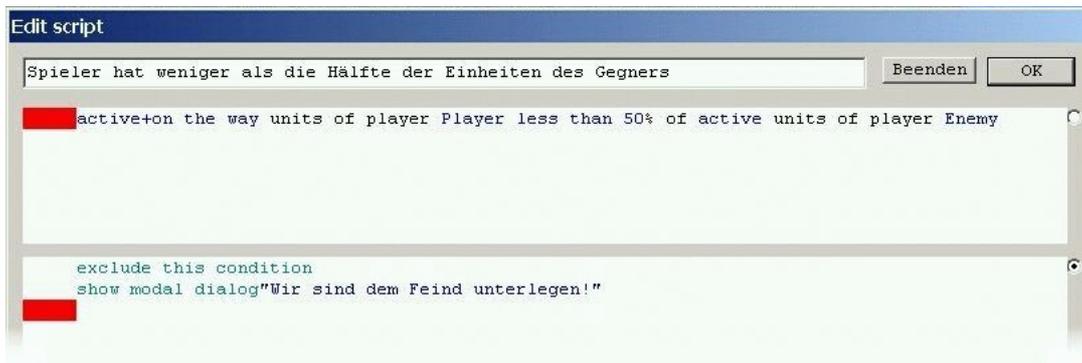


Abb. 66: Vergleich der Einheiten von Spieler und Gegner

7.5.3 Zustand von Einheiten

Um den Munitions- oder Treibstoffvorrat der Einheiten einer Gruppe zu überprüfen, kann man die Bedingung `for units of group <<#Gruppe>> <<Ammo/Fuel>> <<VglOp>> <<#>> %` verwenden, die in der Auswahlliste als `group ammo/fuel` angezeigt wird.

Mit der Aktion `set <<Attr>> <<LimitOp>> <<#>> percent for units of group <<#Gruppe>>` kann man die Vorräte sowie die anderen Attribute der Einheiten (Schaden, Erfahrung, Moral sowie die Anzahl der Wiederbelebungen (Lives, vgl. 4.3) verändern. Der Wert lässt sich entweder nach oben (at most = höchstens) oder nach unten (at least = mindestens) begrenzen.

Verwendet man die Aktion `set <<Attr>> <<LimitOp>> <<#>> percent for units in zone <<#Zone>>`, so bezieht sich die Veränderung der Attribute nicht auf die Einheiten einer Gruppe, sondern auf die Einheiten, die sich in der angegebenen Zone aufhalten.

Von beiden Aktionen gibt es zusätzlich eine Variante, die neben der Gruppe bzw. Zone noch den Spieler festlegen, dessen Einheiten in ihren Attributen verändert werden sollen. Die Aktionen lauten `set <<Attr>> <<LimitOp>> <<#>> percent for units of player <<Spielername>> of group <<#Gruppe>>` bzw. `set <<Attr>> <<LimitOp>> <<#>> percent for units of player <<Spielername>> in zone <<#Zone>>`. Die letztgenannte Aktion wird in der Auswahlliste als `set attribute for units in zone` angezeigt. Korrekt wäre `set attribute for units of player in zone`.

7.5.4 KI-Verhalten

Das in Kapitel 6 beschriebene *Verhalten* von Einheitengruppen kann auch per Skript abgefragt und verändert werden, so dass flexible Reaktionen auf den Verlauf der Mission möglich werden. Die entsprechende Bedingung zur Abfrage des *Verhaltensmodells* einer bestimmten Gruppe lautet `AI.group behaviour <<#Gruppe>> now <<VModell>>`. Dabei kann man auch die Verhaltensoptionen des Verhaltensmodells überprüfen, indem man sie in dem Dialog angibt. Mit den Bedingungen `AI.group location1 <<#Gruppe>> now <<#Zone>>` und `AI.group location2 <<#Gruppe>> now <<#Zone>>` fragt man die *KI-Zonen* ab. Zur Überprüfung der KI-Gruppen dienen die Bedingungen `AI.group group1 <<#Gruppe>> now <<#Gruppe>>` und `AI.group group2 <<#Gruppe>> now <<#Gruppe>>`. Bei den beiden letzten Bedingungen bezeichnet jeweils die erste angegebene Gruppe die Einheitengruppe, deren *KI-Gruppe* abgefragt werden soll.

Die Aktionen, mit denen die Änderungen des Verhaltens bewirkt werden, funktionieren wie das Einstellen des Verhaltens über die in Kapitel 6 beschriebenen Dialoge. Zur Einstellung des Verhaltensmodells einer Gruppe (einschließlich der Verhaltensoptionen) führt man die Aktion `set <<#Gruppe>> AI.group into <<VModell>>` aus. In der Auswahlliste heißt diese Aktion `set AI.group behaviour`. Die beiden Aktionen `set <<#Gruppe>> AI.group location1 into <<#Zone>>` und `set <<#Gruppe>> AI.group location2 into <<#Zone>>` dienen zum Festlegen der KI-Zone. Entsprechend ändert man mit

den Aktionen `set <<#Gruppe>> AI.group group1 into <<#Gruppe>>` und `set <<#Gruppe>> AI.group group2 into <<#Gruppe>>` die KI-Gruppen.

7.5.5 Patrouillen

Um eine *Patrouille* zu realisieren, kann man natürlich per Skript zu bestimmten Zeitpunkten die KI-Zonen der Patrouillengruppe ändern. Es gibt aber auch eine Reihe von Aktionen, die den gleichen Effekt mit weniger Aufwand ermöglichen. Jede Gruppe hat einen Puffer für eine Patrouillenroute. Mit `add patrol locations <<#Zone1>> <<#Zone2>> for group <<#Gruppe>>` fügt man ein Paar von Zonen zu diesem Puffer hinzu. Die angegebenen Zonen können unterschiedlich, aber auch gleich sein. Bei Verwendung unterschiedlicher Zonen teilt sich die Patrouillengruppe auf die beiden Zonen auf. So kann man ganz oder teilweise getrennte Routen realisieren.

Hat man mehrere Paare von Zonen zu der Patrouillenroute der Gruppe hinzugefügt, kann man durch die Aktion `shift patrol location for group <<#Gruppe>>` dafür sorgen, dass für die Patrouille das jeweils nächste Zonenpaar aus dem Puffer als `AI_LOC1` und `AI_LOC2` gesetzt werden, so dass sich die Patrouille dorthin bewegt. Ist das letzte Paar im Puffer erreicht, wird wieder das erste Zonenpaar aus dem Puffer verwendet. Mit Hilfe dieser Aktion kann man in Verbindung mit einem periodischen Timer (vgl. 7.4.2) eine dauerhafte Patrouille einrichten.

Mit der Aktion `clear patrol locations for group <<#Gruppe>>` kann man den Puffer einer Gruppe löschen, wenn man z.B. eine neue Patrouillenroute definieren möchte.

7.5.6 Langstreckenfeuer

Per Skript lassen sich auch einzelne Schüsse aus Haubitzen und Raketenwerfern auslösen. Als Ziel gibt man bei der Aktion `fire a volley by Katyusha of group <<#Gruppe>> into object <<#Marker>>` ein Objekt an, das durch einen Marker gekennzeichnet wurde. Bei der Aktion `fire a volley by Katyusha of group <<#Gruppe>> into zone <<#Zone>>` ist das Ziel eine Zone, in der zufällig ein Feld als Ziel bestimmt wird. Diese Aktionen beziehen sich auf Raketenwerfer der angegebenen Gruppe. Sind mehrere Raketenwerfer in der Gruppe vorhanden, dann wird einer davon zufällig ausgewählt.

Die entsprechenden Aktionen für das Abfeuern von Haubitzen lauten `fire a volley by Howitzer of group <<#Gruppe>> into object <<#Marker>>` bzw. `fire a volley by Howitzer of group <<#Gruppe>> into zone <<#Zone>>`.

Auch für V2-Raketen gibt es eine entsprechende Aktion, allerdings nur für den Beschuss einer Zone: `fire a volley by V2 of group <<#Gruppe>> into zone <<#Zone>>`.

7.5.7 Transfer von Gruppen

Die Einheiten einer Gruppe, die sich an einem bestimmten Ort befinden, können bei Bedarf mit Hilfe der Aktion `set new group <<#Gruppe>> for units of group <<#Gruppe>> in location <<#Zone>>` einer anderen Gruppe zugeordnet werden. Die übertragenen Einheiten übernehmen das Verhalten ihrer neuen Gruppe. Als Ort kann man auch die gesamte Missionszone (Zone 0) angeben, wenn man z.B. alle Einheiten der Gruppe übertragen will.

Eine Gruppe von Einheiten kann auch einem anderen Spieler übertragen werden. Dazu dient die Aktion `deliver group <<#Gruppe>> to player <<Spielername>>`. So kann man z.B. neutrale oder alliierte Einheiten an einen bestimmten Ort schicken und dem Spieler durch Verwendung dieser Aktion übergeben, sobald sie dort ankommen.

7.5.8 Spezielle Bedingungen und Aktionen

Die folgenden Bedingungen und Aktionen sind auf bestimmte Anwendungsfälle hin konzipiert. Da sie nach meiner Auffassung nicht recht in einen anderen Abschnitt passen, habe ich sie hier zusammengefasst.

Gruppe wurde angegriffen

Wurde die angegebene Gruppe vor einer gewissen Zeitspanne angegriffen, dann ist die Bedingung `group <#Gruppe>> was attacked <<VglOp>> <<Zeit>> ago` erfüllt. Durch Abfrage dieser Bedingung lassen sich z.B. Gegenangriffe zu einem geeigneten Zeitpunkt starten.

Einheiten von der Karte entfernen

In manchen Fällen ist es erforderlich oder wünschenswert, bestimmte Einheiten wieder von der Karte zu entfernen. Dies kann man mit der Aktion `kill units of group <#Gruppe>> through the flag <<#Tor>>` realisieren. Alle Einheiten der genannten Gruppe bewegen sich dann zu dem Tor und verschwinden dort. Liegt das Tor am Rand der Missionszone, sieht es so aus, als verließen die Einheiten die Karte.

Eine etwas andere Version dieser Aktion wirkt auf die Einheiten in einer bestimmten Zone. Diese Aktion lautet `kill units in location <#Zone>> through the flag <<#Tor>>`.

Mit Zügen und Kreuzern scheint diese Aktion nicht zu funktionieren.

Einheiten per Skript wiederbeleben

Will man nicht auf die automatische Wiederbelebung (vgl. 5.1.4) zurückgreifen, kann man die Wiederbelebung von Einheiten auch per Skript steuern. Dazu gibt es die Aktion `resurrect units of group <#Gruppe>> via flag <<#Tor>> into zone <<#Zone>> with delay <<Zeit>> health <<#>> shells <<#>> experience <<#>> morale <<#>> fuel <<#>>`.

Die Einheiten der Gruppe, die noch nicht ihre maximale Anzahl von Wiederbelebungen ausgeschöpft haben, werden nach der angegebenen Zeit durch das angegebene Tor in die festgelegte Zone geschickt. Die letzten fünf Angaben, die in der Aktion ergänzt werden müssen, legen fest, welchen Prozentanteil ihrer Lebensenergie (health), Munition (shells), Erfahrung (experience), Moral (morale) und ihres Treibstoffs (fuel) sie bei der Wiederbelebung haben sollen.

7.6 Verstärkungen

7.6.1 Verstärkungen in Einzelspielermissionen

Zum Senden von *Verstärkungen* müssen zunächst Verstärkungen definiert werden (vgl. 5.2.1). Außerdem muss ein Tor (vgl. 5.2.2) vorhanden sein, durch das die Verstärkungen auf die Karte gelangen können. Ist das der Fall, kann man die Verstärkung durch Ausführen der Aktion `send to player <<Spielername>> reinforcement of type <<Verstärkung>> via flag <<#Tor>> into zone <<Zone>> with delay <<Zeit>>` auf die Karte schicken. In dieser Anweisung wird festgelegt, welcher Spieler die Verstärkung bekommt, um welche Art von Verstärkung es sich handelt, durch welches Tor und in welche Zone sie geschickt wird und mit welcher zeitlichen Verzögerung dies geschehen soll.

Gibt man als Zone die gesamte Missionszone, also Zone 0 an, so bewegen sich die Einheiten der Verstärkung zu der blauen Fahne, die zu dem angegebenen Tor gehört.

Welche Verstärkungen unterwegs sind, kann man durch Abfrage der Einheitenklasse „on the way“ feststellen. Die entsprechenden Bedingungen wurden bereits in 7.5.2 erklärt.

7.6.2 Verstärkungen in Mehrspielermissionen

Für die Regelung von Verstärkungen in Mehrspielermissionen gibt es in der Auswahlliste die folgenden Aktionen:

- ... `starting reserve`
- ... `reserve for flags`
- ... `reserve for flags new`
- ... `planes for flags`
- ... `flights of planes for flags`
- ... `message for flags`
- ... `set infinite reinforcements mode`

Da sich dieser User Guide auf Einzelspielermissionen beschränkt, verweise ich zu diesem Thema noch einmal auf die Editor-Fibel von Irrläufer.

7.7 Skriptsteuerung von Flugzeugen

7.7.1 Flugzeuge zuteilen

Um einem Team *Flugzeuge* (vgl. 4.2.1) zuzuteilen, kann man die Aktion `add to player <<Spielername>> <<#>> <<FTyp>>` verwenden. Dadurch erhält das genannte Team die angegebene Anzahl von Flugzeugen des eingestellten Typs. Für den Spieler sind seine Flugzeuge dann im Menü sichtbar, wo er sie bei Bedarf abrufen kann. Damit die Flugzeuge abgerufen werden können, muss auch die Anzahl der möglichen Flüge für diesen Flugzeugtyp entsprechend eingestellt werden. Dazu benutzt man die Aktion `add to player <<Spielername>> <<#>> flights of <<FTyp>>`. Ein Beispiel zeigt die folgende Abbildung.



Abb. 67: Skript zum Hinzufügen eines Bombers

Nach der Ausführung des Skripts kann der Spieler im Menü den Bomber auswählen und verwenden. Das Ergebnis ist in der folgenden Abbildung dargestellt.



Abb. 68: Menü nach Hinzufügen eines Bombers

Mit den beiden Bedingungen `player <<Spielername>> kept <<VglOp>> <<#>> <<FTyp>>` (in der Auswahlliste fälschlicherweise als `player kept players` statt `player kept planes` aufgeführt) und `player <<Spielername>> kept <<VglOp>> <<#>> flights of <<FTyp>>` kann man die Anzahl der Flugzeuge bzw. der möglichen Flüge für das angegebene Team und den angegebenen Flugzeugtyp überprüfen.

7.7.2 Flugzeuge auf die Karte schicken

Eine andere Möglichkeit der Zuteilung von Flugzeugen besteht darin, die Flugzeuge auf die Karte zu schicken und auf einem Flugplatz landen zu lassen. Das realisiert man mit Hilfe der beiden Aktionen `send <<#>> <<FTyp>> of player <<Spielername>> into location <<#Zone>> and land to airfield <<#Flugplatz>>` bzw. `send <<#>> <<FTyp>> of player <<Spielername>> to object <<#Marker>> and land to airfield <<#Flugplatz>>`. Zunächst muss der Spieler mit den bereits beschriebene Aktionen die entsprechende Anzahl von Flugzeugen bekommen haben. Die Flugzeuge fliegen dann zur angegebenen Zone bzw. zum angegebenen Marker und führen dort ihren Auftrag (Aufklärung, Bombardierung usw.) aus. Anschließend landen sie auf dem in der Aktion festgelegten Flugplatz. Wird als Flugplatz der Wert 0 vergeben, landen die Flugzeuge nicht, sondern verlassen die Karte wieder.

Mit Hilfe der zweiten Aktion kann man z.B. einzelne mit einem Marker versehene Objekte (z.B. Brücken oder Gebäude) gezielt bombardieren lassen.

7.7.3 Flugzeuge zwischen Flugplätzen wechseln lassen

Um Flugzeuge einer bestimmten Gruppe, die sich bereits auf einem Flugplatz der Karte befinden, zu einem anderen Flugplatz zu bewegen, verwendet man die Aktion `send <<#>> planes of group <<#Gruppe>> to airfield <<#Flugplatz>>`. Befinden sich die Flugzeuge zum Zeitpunkt der Skriptausführung in der Luft, dann wird die Aktion nach ihrer Landung ausgeführt. Bei dieser Aktion kann man nur Flugplätze auf der Karte angeben, der Wert 0 ist nicht verfügbar.

7.7.4 Verwendung von Flugrouten

In manchen Fällen benötigt man eine genauere Kontrolle über die *Flugroute* von Flugzeugen. Dazu kann man per Skript eine Route festlegen. Zunächst wird die Route mit der Aktion `start avia track` angelegt. War bereits eine Flugroute definiert, so wird diese gelöscht.

Die Punkte der Flugroute werden mit `add route point over object <<#Marker>>` festgelegt. Dazu muss an dem entsprechenden Punkt der Route ein Marker auf der Karte angebracht werden. Soll das Flugzeug an einem Punkt der Route einen Auftrag (z.B. Bombardierung) ausführen, verwendet man stattdessen die Aktion `add strike point over object <<#Marker>>`. Der Auftrag ist von der Art des Flugzeugs abhängig.

Solange keine neue Route definiert wird, kann man durch die Aktion `start <<#>> <<FTyp>> of player <<Spielername>> on avia track and land to airfield <<#Flugplatz>>` Flugzeuge entlang der Route fliegen lassen. Sie landen danach auf dem angegebenen Flugplatz. Wird als Flugplatz der Wert 0 angegeben, verlassen die Flugzeuge die Karte.

7.8 Missionsende und Kampagnen

7.8.1 Sieg und Niederlage

Eine Mission kann je nach Verlauf als *Sieg* (victory), *Niederlage* (defeat) oder *Unentschieden* (tie game) gewertet werden. Um eine Mission mit einem dieser Ergebnisse zu beenden, wird die Aktion `terminate mission <<Ergebnis>>` ausgeführt, sobald die entsprechenden Bedingungen erfüllt sind. Häufig wird z.B. die Anzahl aktiver Einheiten des Spielers und des Gegners überwacht. Hat der Gegner keine Einheiten mehr, wird die Mission mit einem Sieg beendet. Hat dagegen der Spieler alle Einheiten verloren, wird die Mission als Niederlage beendet.

Wird für Sieg und Niederlage nicht jeweils ein Skript angelegt, kann es passieren, dass die Mission nie beendet wird. In Mehrspielermissionen wird diese Aktion nicht verwendet, da sie automatisch beendet werden.

7.8.2 Folgemission in Kampagnen

Für *Kampagnenmissionen* gibt es die Aktion `set next mission to <<Folgemission>>`, die dazu dient, die *Folgemission* (Next in campaign sequence oder `Map<<#>>:Mission <<#>>`) festzulegen. Neben einer Mission kann man auch die Beendigung der Kampagne mit Sieg (Campaign complete) oder Niederlage (Campaign failed) sowie eine erzwungene Wiederholung der aktuellen Mission (Forced replay) einstellen (vgl. Abb. 69).

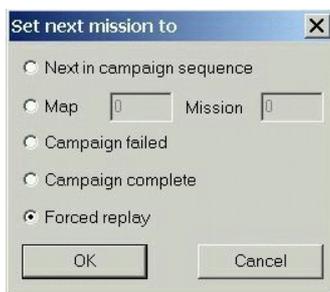


Abb. 69: Festlegen der Folgemission einer Kampagnenmission

Da es möglich ist, Einheiten aus einer Mission in eine andere zu übertragen (vgl. 7.8.3), ist es sinnvoll, zu Beginn der neuen Mission den Spielstand zu speichern, da andernfalls bei einem Neustart dieser

Mission die übertragenen Einheiten verloren gehen. Das Speichern des Spielstandes kann man auch per Skript auslösen, indem man die Aktion **Save mission** verwendet.

7.8.3 Übertragung von Einheiten zwischen Missionen

Um Einheiten, die am Ende einer Mission noch vorhanden sind, in eine andere Mission übernehmen zu können, gibt es in SSRW acht sogenannte *Forces* (im folgenden *Container*³³). Dabei handelt es sich um temporäre Aufbewahrungsorte für Einheiten. Am Ende einer Mission können vorhandene Einheiten in einen solchen Container übertragen und zu Beginn der folgenden Mission aus diesem Container als Verstärkung auf die Karte gebracht werden.

Um Einheiten in einen Container zu übertragen, gibt es verschiedene Aktionen. Durch **add all units to <<Force>>** werden alle Einheiten in den angegebenen Container übertragen. Etwas differenzierter geht es aber auch: **add all units of player <<Spielername>> to <<Force>>** überträgt alle Einheiten des genannten Spielers in den angegebenen Container. Mit **add group <<#Gruppe>> to <<Force>>** lässt sich das gleiche für die Einheiten einer bestimmten Gruppe bewerkstelligen. Entsprechend überträgt **add from location <<#Zone>> to <<Force>>** alle Einheiten, die sich an einem bestimmten Ort befinden, in den Container.

In der folgenden Mission geht man den umgekehrten Weg und setzt die Einheiten aus einem Container an einem festgelegten Ort ab. Dazu kann man die Aktion **put units from <<Force>> into location <<#Zone>>** verwenden. Da die Gruppenzuordnung der Einheiten erhalten bleibt, die Gruppen aber in der neuen Mission im Normalfall andere Bedeutungen haben, ist es günstiger die folgende Version zu verwenden: **put units from <<Force>> into location <<#Zone>> and set group <<#Gruppe>> to them.** Dadurch hat man die Möglichkeit, den übertragenen Einheiten eine passende Gruppe zuzuordnen.

Diese beiden Aktionen lassen sich eigentlich nur zu Beginn der Mission einsetzen, da die übertragenen Einheiten plötzlich auf der Karte erscheinen. Im weiteren Verlauf würde es so aussehen, als ob sie sich aus dem Nichts materialisierten. Trotzdem kann man übertragene Einheiten auch später in die Mission einbringen. Dazu sollte man für jeden verwendeten Container eine leere Verstärkung definieren und mit einem passenden Namen versehen. Durch die Aktionen **form <<Verstärkung>> from <<Force>>** und – wegen der Gruppenzuordnung besser – **form <<Verstärkung>> from <<Force>> and set <<#Gruppe>> to them** wird die angegebene Verstärkung durch die Einheiten aus dem angegebenen Container ersetzt (und im zweiten Fall einer Gruppe zugeordnet). Diese Verstärkung kann man dann wie eine ganz normale Verstärkung auf die Karte schicken.

Der Inhalt eines Containers kann mit Hilfe der Aktion **clear <<Force>>** auch wieder gelöscht werden. Man sollte diese Aktion immer ausführen, bevor die Einheiten für die Folgemission in die Container übernommen werden, da die Container nach der Übertragung nicht automatisch geleert werden. Andernfalls werden einmal in den Container verschobene Einheiten nicht nur in die nächste, sondern auch in alle darauf folgenden Missionen übertragen.

Die Bedingung **units of force <<Force>> <<VglOp>> <<#>>** kann man verwenden, um die Anzahl der Einheiten zu bestimmen, die in dem Container vorhanden sind. Mit Hilfe dieser Bedingung kann man z.B. Verstärkungen von der Anzahl der Einheiten abhängig machen, die aus der vorangegangenen Mission übernommen wurden.

7.9 Zellen (Variablen)

Durch das Konzept der *Zellen*³⁴ werden die Möglichkeiten der Skriptprogrammierung erheblich erweitert, auch wenn das auf den ersten Blick nicht unbedingt ersichtlich wird. Zellen entsprechen im Prinzip den Variablen, die man aus Programmiersprachen kennt. Es handelt sich bei den Zellen, die in SSRW zur Verfügung stehen, um Speicherplätze, von denen jeder eine ganze und nicht negative Zahl enthalten kann. Insgesamt gibt es 100 Zellen, die mit den Nummern 0 bis 99 versehen sind.

³³ Da das Wort Force für mein Sprachgefühl in einem deutschen Satz etwas merkwürdig klingt, werde ich im folgenden den Begriff Container verwenden, der den Sachverhalt auch ganz gut beschreibt.

³⁴ Damit sind nicht die Teilfelder gemeint, in die die Felder der Karte unterteilt sind, obwohl diese ebenfalls als Zellen bezeichnet werden (vgl. 3.6).

7.9.1 Bedingungen und Aktionen

Die Aktion `let cell <<#Zelle>> contain <<Wert>>` weist der angegebenen Zelle einen Wert zu, d.h. dieser Wert bildet nach Ausführung der Aktion den Inhalt der Zelle. Der Wert kann dabei entweder eine Zahl oder der Inhalt einer weiteren Zelle sein. Das wird in dem Dialog eingestellt, der in der Abb. 70 zu sehen ist.

Zur Manipulation von Zelleninhalten gibt es noch weitere Aktionen. Die Aktion `<<#Zelle>> <<MathOp>> <<Wert>>` wird in der Auswahlliste als mathematical operation aufgeführt. Dahinter verbergen sich eigentlich zwei Operationen, nämlich Addition und Subtraktion. Der Inhalt der Zelle wird um den angegebenen Wert (Zahl oder Zelleninhalt) erhöht bzw. verringert. Interessant ist auch die Aktion `add to cell <<#Zelle>> number of <<EinhKl>> units of group <<#Gruppe>>`, die eine Zählung von Einheiten der genannten Gruppe ermöglicht. Die ermittelte Anzahl wird dann in der angegebenen Zelle gespeichert. Auf diese Weise kann man z.B. die Gesamtzahl der vorhandenen Einheiten bestimmen, um zu verhindern, dass Einheitenlimit durch Verstärkungen zu überschreiten.

Allerdings ist die Bezeichnung der Aktion etwas irreführend. Die Anzahl der Einheiten wird nicht zu dem Wert der Zelle addiert, wie der Name andeutet. Der bisher in der Zelle gespeicherte Wert wird mit der Anzahl der Einheiten überschrieben.

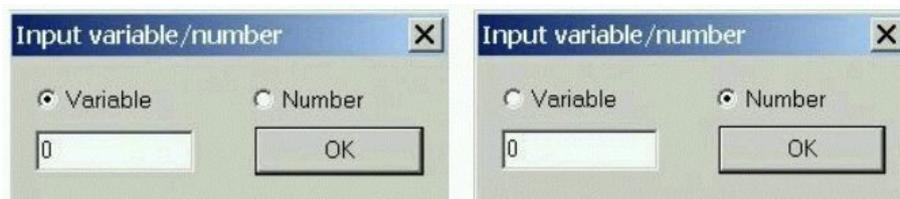


Abb. 70: Wert einer Zelle (links: Inhalt der Zelle C_0, rechts: die Zahl 0)

Der Inhalt einer Zelle kann durch die Bedingung `value in cell <<#Zelle>> <<VglOp>> <<Wert>>` abgefragt werden. Der Inhalt der Zelle kann dabei mit einem Zahlenwert oder dem Inhalt einer weiteren Zelle verglichen werden.

7.9.2 Einsatz von Zellen

Auf den ersten Blick erscheint das Konzept der Zellen vielleicht nicht allzu hilfreich. Man kann in der Tat auch ohne Zellen eine ganz ansprechende Mission erstellen. Bei näherer Betrachtung erleichtern Zellen aber viele Abläufe ganz erheblich. Einige Beispiele findet man in den folgenden Abschnitten.

Zellen als Flags

Eine naheliegende Einsatzmöglichkeit von Zellen ist die Verwendung als Flags³⁵, die bestimmte Bedingungen signalisieren. Da Bedingungen recht umfangreich werden können (ein noch ziemlich harmloses Beispiel ist in Abb. 71 zu sehen), wird die Eingabe der Bedingungen bei der etwas umständlichen Benutzerschnittstelle des Skripteditors mühsam, wenn man diese Bedingung in mehreren Skripten verwenden will. Zudem werden lange Bedingungen schnell unübersichtlich.



Abb. 71: Beispiel für eine Bedingung mit mehreren Teilbedingungen

³⁵ In diesem Zusammenhang ist mit Flag eine Möglichkeit gemeint, die Erfüllung (oder auch Nichterfüllung) einer Bedingung anzuzeigen. Der Begriff Flag in dieser Bedeutung kommt eigentlich aus dem Bereich der Programmierung in Maschinensprache bzw. der Prozessortechnik. Im Skripteditor werden zwar auch die Tore für Verstärkungen sowie die Zeppeline in Mehrspielermissionen als Flags bezeichnet, diese sind hier aber nicht gemeint.

Durch Verwendung von Zellen lässt sich Abhilfe schaffen. Dazu sucht man eine Zelle aus, die die Bedingung repräsentieren soll. Ist die Bedingung nicht erfüllt, bekommt die Zelle den Wert 0. Sobald die Bedingung erfüllt ist, wird dieser Wert auf 1 geändert.

Statt nun in jedem Skript die lange Bedingung auszuwerten, kann man ein Skript zur Auswertung der Bedingung anlegen, in dem der Wert der Zelle von 0 auf 1 geändert wird. In allen anderen Skripten, die diese Bedingung verwenden sollen, reicht es dann aus, lediglich den Wert der Zelle mit 1 zu vergleichen, denn die Zelle bekommt ja nur dann den Wert 1, wenn die Bedingung erfüllt ist.

In ähnlicher Weise kann man Zellen einsetzen, wenn mehrere unterschiedliche Bedingungen die gleichen Auswirkungen haben. Ich verwende z.B. immer die Zelle C0 als Flag für eine Niederlage. Zu Beginn der Mission trage ich den Wert 0 ein. Sobald eine der möglichen Bedingungen für eine Niederlage eintritt, wird der Wert von C0 auf 1 geändert. Sobald C0 den Wert 1 hat, wird die Mission mit einer Niederlage beendet. Auf diese Weise kann je nach Ursache der Niederlage eine jeweils unterschiedliche Meldung ausgegeben werden. Trotzdem brauche ich die Aktionen zum Beenden der Mission, zum Festlegen der Folgemission und zum Übertragen der Einheiten in die Folgemission nur in einem Skript einzugeben.

Zellen als Zähler

Da man mit den mathematischen Operationen den Wert der Zelle auch erhöhen und verringern kann, lassen sich Zellen als Zähler einsetzen. So kann man z.B. eine Zelle benutzen, um die erreichten Teilziele mitzuzählen. Zu Beginn bekommt die Zelle den Wert 0. Bei Erreichen eines Teilziels wird der Wert um 1 erhöht. Den Sieg in der Mission kann man dann daran erkennen, dass der Wert der Zelle gleich der Anzahl der Teilziele ist.

Auch die Anzahl von Einheiten, die auf verschiedene Gruppen verteilt sind, kann man mit Hilfe von Zellen bestimmen. Dazu weist man einer Zelle (z.B. C5) zunächst den Wert 0 zu. In einer weiteren Zelle (z.B. C6) wird mit der Aktion `add to cell <<#Zelle>> number of <<Einhkl>> units of group <<#Gruppe>>` die Anzahl der Einheiten einer Gruppe gespeichert. Dadurch wird der bisherige Wert der Zelle C6 überschrieben. Anschließend erhöht man den Wert von C5 mit `<<#Zelle>> <<MathOp>> <<Wert>>` um den Wert der Zelle C6. Diese Aktionen führt man für jede Gruppe aus, so dass am Ende die Zelle C5 die gesamte Anzahl der Einheiten dieser Gruppen enthält.

7.10 Weitere Bedingungen und Aktionen

7.10.1 Zustand von Objekten

Die Bedingung `<<#Marker>> object is dead` wird verwendet, wenn man überprüfen möchte, ob das mit einem bestimmten Marker gekennzeichnete Objekt zerstört wurde. Entsprechend fragt man mit `<<#Marker>> object is fully repaired` ab, ob das Objekt vollständig repariert wurde.

7.10.2 Zünden von Minen

Die auf der Karte verteilten Minen lassen sich per Skript auslösen, wobei auch eine zeitverzögerte Auslösung möglich ist. Die Aktion `for all mines in zone <<#Zone>> set timer to <<#>> seconds` lässt alle Minen in der angegebenen Zone nach der eingestellten Zeit explodieren³⁶.

7.11 Verwendete Abkürzungen

Nachfolgend sind überblicksartig noch einmal die Schreibweisen aufgeführt und erläutert, die in diesem Kapitel verwendet wurden.

<code><<Ammo/Fuel>></code>	kann die Werte <code>Ammo for 1st weapon</code> , <code>Ammo for 2nd weapon</code> und <code>Fuel</code> annehmen
<code><<Attr>></code>	eines der Attribute einer Einheit (<code>hp</code> , <code>ammo</code> , <code>experience</code> , <code>morale</code> , <code>fuel</code> , <code>lives</code>)

³⁶ Für die Minenart `time bomb` funktioniert das leider nicht. Diese Minen explodieren fröhlich nach der üblichen Zeit, auch wenn man per Skript etwas anderes einstellt.

<<EinhKl>>	eine Einheitenklasse (Active, Gun, In hell, On the way und beliebige Kombinationen davon)
<<Ergebnis>>	eines der möglichen Ergebnisse einer Mission (Sieg, Niederlage oder unentschieden)
<<Folgemission>>	Folgemission einer Kampagnenmission oder Kampagnenende
<<Force>>	einer der acht möglichen Container zum Übertragen von Einheiten zwischen zwei Missionen
<<FTyp>>	einer der möglichen Flugzeugtypen
<<LimitOp>>	entweder at most (höchstens) oder at least (mindestens)
<<MathOp>>	eine mathematische Operation, entweder Addition (increase by) oder Subtraktion (decrease by)
<<Missionstext>>	einer der Missionstexte (Phrases)
<<Spielername>>	der Name eines der vier möglichen Teams (Player, Enemy, Ally, Neutral)
<<Verstärkung>>	eine der definierten Verstärkungen
<<VglOp>>	ein Vergleichsoperator (more than, exactly, less than). Je nach Art der Bedingung stehen nicht alle Operatoren zur Verfügung.
<<Vmodell>>	eines der zur Verfügung stehenden Verhaltensmodelle
<<Wert>>	entweder eine ganze und nicht negative Zahl oder die Nummer einer Zelle zum Speichern solcher Zahlen
<<Zeit>>	eine Zeitangabe in der Form hh:mm:ss, also Stunden:Minuten:Sekunden
<<#>>	eine ganze (i.d.R. nicht negative) Zahl
<<#Flugplatz>>	Nummer eines Flugplatzes auf der Karte
<<#Gruppe>>	Bezeichnung einer Gruppe, z.B. A0 oder F6
<<#Marker>>	die Nummer eines Markers, Wertebereich 0..99
<<#Timer>>	die Nummer eines Timers, Wertebereich 0..99
<<#Tor>>	der Buchstabe eines Tors für Verstärkungen
<<#Zelle>>	die Nummer einer Zelle zum Speichern von Zahlen
<<#Zone>>	die Nummer einer Zone (Location), Wertebereich 0..63

8 Der Briefing-Editor

Der *Briefing-Editor* *anied.exe* wird bei SSRW zwar nicht mitgeliefert, wohl aber bei SuStII. Da die mit dem Programm erstellten Briefings auch mit SSRW kompatibel sind, erschien es mir sinnvoll, auf diesen Editor kurz einzugehen.

Die Erstellung von Briefing-Animationen wird oft abgelehnt. Dafür scheint es vor allem zwei Gründe zu geben. Zum einen wird der Aufwand für die Erzeugung der Animation als zu hoch angesehen. Das ist meiner Meinung nach jedoch – Entschuldigung für die harten Worte – Unsinn. Eine vollständige Animation kann bei systematischem Vorgehen in weniger als einer halben Stunde erstellt werden. Auch komplizierte Animationen brauchen selten mehr als zwei Stunden, wenn man sich erst einmal an die Arbeitsweise des Editors gewöhnt hat. Der zweite Grund für die Ablehnung von Briefing-Animationen ist die nicht zu bestreitende Tatsache, dass die Missionsdatei durch die Animationsdaten größer wird³⁷.

Mir scheint, dass die Ursache für diese beiden Ablehnungsgründe in dem Konzept der *Keyframes* zu suchen ist. Wenn man dieses Konzept verstanden hat und richtig einsetzt, ist weder der Arbeitsaufwand noch der Zuwachs der Dateigröße besonders gravierend. Auch wenn dieses Konzept in 8.5.1 und 8.5.2 ausführlich erläutert wird, will ich hier schon einmal ein kleines Beispiel anführen. In Briefing-Animationen werden Truppenbewegungen oft dadurch dargestellt, dass ein Pfeil ausgehend von einem Punkt bis zum Ziel der Truppenbewegung verlängert wird. Um das zu erreichen legt man das Aussehen des Pfeils zu Beginn der Bewegung fest und trägt für dieses Aussehen einen Keyframe ein. Entsprechend passt man das Aussehen des Pfeils am Ende der Animation an und legt dafür einen zweiten Keyframe fest. Dann gibt man an, wie lang die Animation dauern soll, z.B. 100 Einzelbilder. Diese 100 Einzelbilder muss man nicht selbst erzeugen, das macht der Editor automatisch. Mit drei Angaben (Pfeil am Anfang, Pfeil am Ende und Animationsdauer) hat man also die Animation des Pfeils vollständig bearbeitet. Das geht tatsächlich genau so schnell und einfach, wie es sich anhört!

8.1 Die Benutzeroberfläche des Briefing-Editors

Der Editor *anied.exe* wird über Start > Programme > Sudden Strike II > Briefing-Editor gestartet. Die Benutzeroberfläche ist im Vergleich mit dem Karten- und Missionseditor *edit3.exe* vergleichsweise simpel aufgebaut (vgl. Abb. 72).

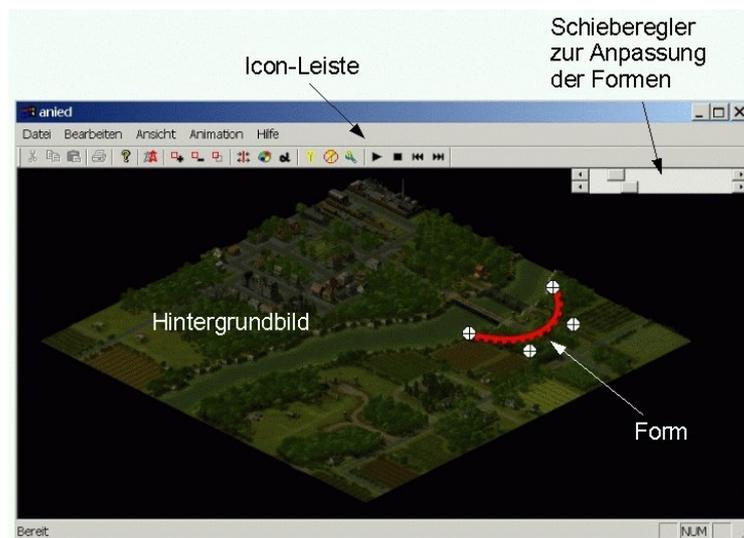


Abb. 72: Das Fenster des Briefing-Editors

Im Fenster sieht man ein bereits geladenes Hintergrundbild und eine der Formen, aus denen die Briefing-Animation besteht. Die weißen Kreise an der Form zeigen an, dass diese Form gerade ausgewählt ist. Sie dienen außerdem dazu, die Ausrichtung und Größe der Form zu verändern.

³⁷ Ein Beispiel: Größe einer Missionsdatei: 149KB ohne, 205KB mit Briefinganimation.

Die Schieberegler oben rechts beeinflussen das Aussehen der Form. Je nach Form werden hier unterschiedlich viele Regler eingeblendet, die jeweils einen Aspekt des Aussehens beeinflussen. Im abgebildeten Fall kann man z.B. über den oberen Regler die Anzahl der „Zähne“ der Verteidigungslinie und über den unteren die Länge dieser Zähne einstellen.

Über die Icons der *Icon-Leiste* kann man die Funktionen des Briefing-Editors auswählen. Die *Icons* im einzelnen (von links nach rechts)³⁸:



Info (Hilfe > Über Anied...): Zeigt die Copyright-Meldung des Editors an.



Zur nächsten Form wechseln: Die Formen lassen sich nicht durch Anklicken auswählen. Stattdessen wird mit diesem Icon zur nächsten Form weitergeschaltet.



Form hinzufügen (Bearbeiten > Form hinzufügen, Taste Einfg): Fügt eine weitere Form in die Briefing-Animation ein.



Form löschen (Bearbeiten > Form löschen, Taste Entf): Entfernt die aktuelle Form aus der Briefing-Animation.



Form kopieren (Bearbeiten > Form kopieren): Kopiert die aktuelle Form und fügt die Kopie an der gleichen Position ein.



Linienbreite: Ermöglicht die Veränderung der Linienbreite der aktuellen Form.



Farbe: Ermöglicht die Veränderung der Farbe der aktuellen Form.



Alpha-Wert: Ermöglicht die Veränderung des Alpha-Wertes der aktuellen Form. Durch Veränderung dieses Wertes kann man eine Form ein- oder ausblenden.



Keyframe hinzufügen (Animation > Keyframe hinzufügen, Taste  auf dem Ziffernblock): Fügt für die aktuelle Form einen Keyframe hinzu.



Keyframe löschen (Animation > Keyframe löschen, Taste  auf dem Ziffernblock): Der aktuelle Keyframe der markierten Form wird gelöscht.



Keyframes bearbeiten (Ansicht > Dialog bearbeiten, Tasten +): Blendet den Dialog zum Bearbeiten der Keyframes für die aktuelle Form ein.



Animation starten (Animation > Animieren, Leertaste): Startet die erstellte Briefing-Animation.



Animation anhalten (Animation > Animation stoppen, Taste Esc): Hält die gestartete Animation an der gerade gezeigten Stelle an.



Vorheriger Keyframe (Animation > Vorheriger Keyframe, Taste Bild↓): Springt um einen Keyframe der aktuellen Form zurück.



Vorheriger Keyframe (Animation > Nächster Keyframe, Taste Bild↑): Springt um einen Keyframe der aktuellen Form vorwärts.

³⁸ Die ersten vier Icons (Ausschneiden, Kopieren, Einfügen und Drucken) habe ich ausgelassen, da sie funktionslos zu sein scheinen. Jedenfalls habe ich sie bisher nur im inaktiven Zustand zu sehen bekommen.

8.2 Vorüberlegungen

Bevor man eine Briefing-Animation erstellt, sollte man überlegen, wie die Animation ablaufen soll. Das hängt natürlich von den Zielen ab, die der Spieler im Missionsverlauf erreichen soll. Die folgenden Aspekte sollten dabei bedacht werden:

- Welche Formen sollen in der Animation vorkommen? Der Briefing-Editor stellt eine ganze Reihe von Formen zur Verfügung. Dazu gehören z.B. Panzer, Flugzeuge, Schiffe, Linien, Pfeile und Verteidigungszonen.
- Welche Farben haben die Formen? Ändern manche Formen im Verlauf der Animation ihre Farbe? Bei den Verteidigungszonen ist z.B. ein Farbwechsel sinnvoll, um anzudeuten, dass eine Region von einem anderen Team besetzt wird.
- Sind alle Formen während der gesamten Dauer der Animation sichtbar? Formen lassen sich auch ein- und ausblenden.
- Stehen die Formen still oder bewegen sie sich? Ändern sie evtl. ihre Form?

8.3 Erstellen und Laden des Hintergrundbildes

Der erste Schritt bei der Arbeit an einer Briefing-Animation ist die Erstellung eines *Hintergrundbildes* für die Animation³⁹. Dazu erzeugt man zunächst im Missionseditor eine Bitmap der Karte (vgl. 3.1.3) bzw. der Missionszone (falls die Missionszone kleiner ist als die Karte, vgl. 4.1.4). Dabei sollte man als Dateityp *.bmp einstellen, da der Briefing-Editor beim Laden von jpg-Dateien abstürzt.

Die gespeicherte Bitmap muss nun noch auf die passende Größe gebracht werden. Beim Erzeugen der Bitmap im Missionseditor kann man zwar die Auflösung des Ergebnisses zwischen der Originalgröße 1:1 und einer Verkleinerung von 1:16 einstellen, die exakte Auflösung der Briefing-Karte erreicht man so aber i.d.R. nicht. An dieser Stelle kommt nun ein Bildbearbeitungsprogramm ins Spiel.

Ich werde die folgenden Arbeitsschritte am Beispiel des Programms IrfanView⁴⁰ erläutern, das ich für solche Zwecke gern einsetze, da es im Gegensatz zu anderen Bildbearbeitungsprogrammen nur relativ wenig Arbeitsspeicher belegt und außerdem sehr leicht zu bedienen ist. Jedes andere ernstzunehmende Bildbearbeitungsprogramm bietet aber ebenfalls die benötigten Funktionen.

Die im Missionseditor erzeugte Bitmap wird zuerst über Datei > Öffnen... in IrfanView geladen. Die während des Briefings im Spiel angezeigte Karte hat die Auflösung 780x390. Auf diese Größe muss das geladene Bild nun geändert werden. Dazu wählt man Bild > Größe ändern... aus. Es erscheint der in der folgenden Abbildung gezeigte Dialog.

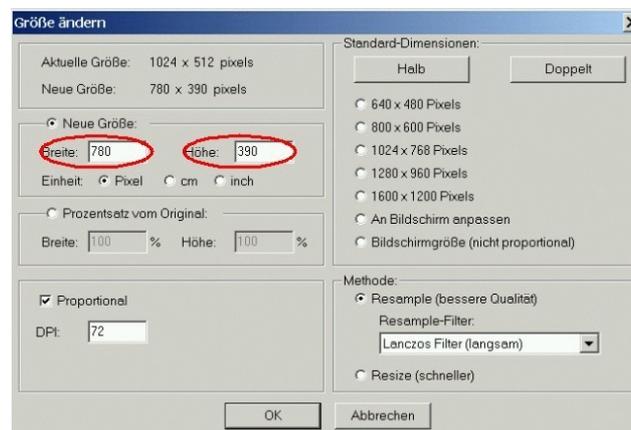


Abb. 73: Anpassung der Größe des Hintergrundbildes in IrfanView

³⁹ Das Hintergrundbild ist zwar für die Arbeit im Briefing-Editor nicht zwingend erforderlich, aber ohne Hintergrundbild wird man kaum eine Chance haben, die Bestandteile der Animation so zu plazieren, dass sie im fertigen Briefing an der korrekten Stelle angezeigt werden.

⁴⁰ Download z.B. unter <http://www.irfanview.de>. Die Software ist für den nichtkommerziellen Einsatz kostenlos.

Im Feld Breite gibt man nun die für das Briefing benötigte Breite von 780 Pixel⁴¹ an. Ist das Häkchen bei Proportional gesetzt, passt sich die Höhe automatisch so an, dass das Bild beim Verkleinern nicht verzerrt wird. Es erscheint also automatisch der Wert 390 im Feld Höhe. Ist das nicht der Fall, gibt man ihn von Hand ein. Anschließend wird die Größenänderung mit Klick auf OK bestätigt.

Nun wird das verkleinerte Bild noch über Datei > Speichern wieder als bmp-Datei abgespeichert und die Arbeit im Briefing-Editor kann beginnen. Über Datei > Hintergrund laden wird die verkleinerte Bitmap im Briefing-Editor geladen.

8.4 Arbeiten mit Formen

Der Briefing-Editor stellt eine ganze Reihe verschiedener *Formen* zur Verfügung, die in der Animation verwendet werden können (vgl. Abb. 74). Diese Formen können nach dem Einfügen noch an die Erfordernisse des erstellten Briefings angepasst werden (z.B. Größe, Farbe oder Linienstärke).

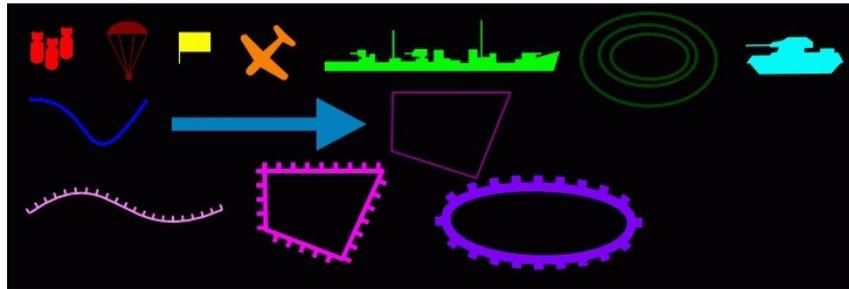


Abb. 74: Eine Auswahl der für Briefings zur Verfügung stehenden Formen

8.4.1 Formen einfügen, löschen und kopieren

Über Bearbeiten > Form hinzufügen wird der Auswahldialog für die Formen (vgl. Abb. 75) geöffnet. Nach Auswahl der gewünschten Form und Klick auf OK wird diese im Fenster plaziert und als aktuelle Form markiert (erkennbar an den weißen Kreisen an der Form, vgl. Abb. 72). Die Farbe und Linienstärke entspricht der zuletzt aktuellen Form.



Abb. 75: Auswahldialog für Formen

Bevor die Form verschoben oder in ihrer Größe verändert werden kann, muss mindestens ein *Keyframe* für sie angelegt werden. Dazu drückt man z.B. auf dem Ziffernblock die Taste $\boxed{0}$. In dem Dialog, der sich daraufhin öffnet (vgl. Abb. 76), sollte man zu Beginn der Bearbeitung die Nummer 0 für den ersten Keyframe vergeben. Anschließend kann man die Form weiter bearbeiten (vgl. 8.4.2).

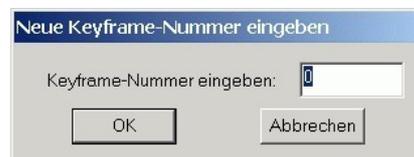


Abb. 76: Anlegen eines Keyframes für die aktuelle Form

Die aktuelle Form kann man über Bearbeiten > Form löschen auch wieder aus der Briefing-Animation entfernen.

Über Bearbeiten > Form kopieren kann man eine Kopie der aktuellen Form erstellen und in die Animation einfügen. Die Kopie wird an der gleichen Position eingefügt wie das Original und muss wie eine neu erstellte Form zunächst mit mindestens einem Keyframe versehen werden (s.o.).

⁴¹ Pixel = Bildpunkt

8.4.2 Formen bearbeiten

In den meisten Fällen sieht eine gerade eingefügte Form noch nicht so aus, wie man es gern hätte. Man kann die aktuelle Form aber entsprechend bearbeiten. Zu beachten ist dabei, dass sich die Veränderung der Einstellungen der aktuellen Form immer auf den gerade aktuellen Keyframe dieser Form beziehen! Genaueres dazu findet man in 8.5.

Die aktuelle Form ist daran zu erkennen, dass in ihrem Umfeld zwei oder vier weiße Kreise, sog. *Kontrollpunkte*, zu sehen sind. Klickt man auf einen dieser Kreise und hält die Maustaste gedrückt, kann man die Größe und Ausrichtung der Form verändern.

Bewegt man die Maus bei gedrückter linker Maustaste, ohne dass der Mauszeiger auf einem der Kontrollpunkte steht, kann man die aktuelle Form an eine andere Position verschieben.

Manche Formen sehen nach dem Einfügen zunächst etwas seltsam aus, wenn sie mit einer ungeeigneten Linienstärke versehen sind. Ein Beispiel zeigt die folgende Abbildung.

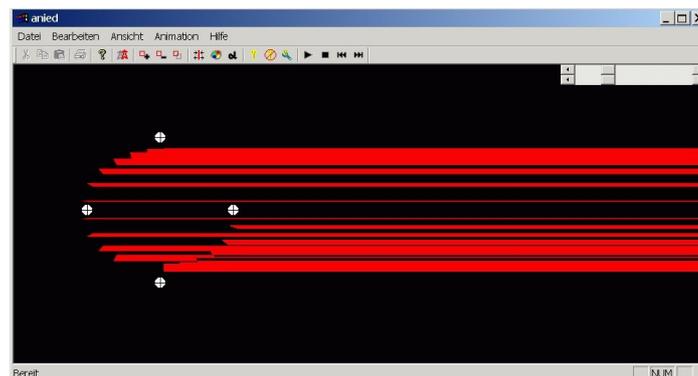


Abb. 77: Form mit ungeeigneter Linienstärke (hier Verteidigungsregion mit Linienstärke 0)

Die *Linienstärke*⁴² lässt sich aber nach einem Klick auf das Icon  ändern. Welche Linienstärke geeignet ist, hängt von der Art der Form ab (vgl. 8.4.3). Nach einem Klick auf OK wird die eingegebene Linienstärke übernommen.

Entsprechend ändert man die *Farbe* der aktuellen Form durch einen Klick auf das Icon . In dem Dialog, der sich daraufhin öffnet kann man eine der vorgegebenen Farben auswählen. Die gewählte Farbe wird nach einem Klick auf OK übernommen. Wenn man mit den Standardfarben nicht zufrieden ist, kann man auch eigene Farben definieren. Dazu klickt man auf Farben definieren >>, was zu einer Erweiterung des Dialogs zur Farbauswahl führt. In dem großen quadratischen Feld wählt man dann den gewünschten Farbton aus, dessen Helligkeit mit Hilfe des Farbstreifens am rechten Rand noch verändert werden kann. Nach Klick auf Farben hinzufügen wird die so erzeugte Farbe links unter Benutzerdefinierte Farben angezeigt. Sie steht dann zusätzlich zu den Standardfarben zur Verfügung.

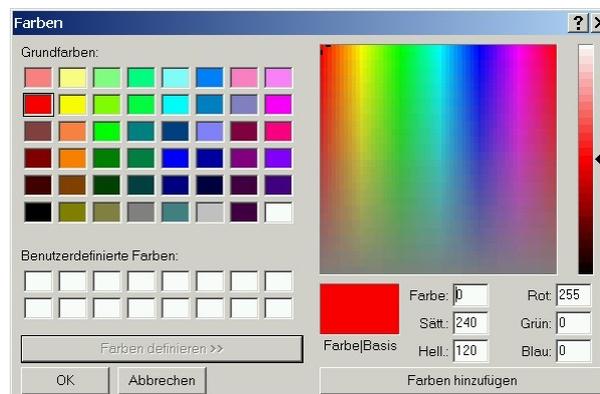


Abb. 78: Auswahl und Definition von Farben

⁴² Werte größer als Null werden als Linienbreite in Pixel interpretiert. Je größer der Wert ist, desto dicker ist die Linie.

Nach einem Klick auf das Icon  öffnet sich ein Dialog, in dem man den *Alpha-Wert* der aktuellen Form angeben kann. Der Alpha-Wert, der zwischen 0 und 1 (jeweils einschließlich) liegen muss⁴³, bestimmt die Transparenz der Form. Bei Eingabe von 1 ist die Form nicht transparent, bei 0 dagegen vollständig transparent und somit nicht sichtbar. Mit Hilfe dieses Wertes kann man also Formen im Verlauf der Animation ein- und ausblenden.

8.4.3 Formenspezifische Einstellungen

Da die Bearbeitung der Formen sich in einigen Punkten unterscheidet, werden nachfolgend jeweils die Einstellungsmöglichkeiten der unterschiedlichen Formen genauer beschrieben.

Figuren

In diesem Abschnitt werden die Einstellungsmöglichkeiten für *Figuren* wie Flugzeug, Panzer usw. behandelt. In Abb. 74 sieht man diese Formen in der oberen Reihe.

Die Figuren können über die beiden Kontrollpunkte in Größe und Ausrichtung verändert werden. Bis auf die Flugzeugfiguren werden sie waagrecht ausgerichtet, indem man beide Kontrollpunkte waagrecht nebeneinander schiebt.

Man kann die Figuren entweder als Umriss oder als ausgefüllte Figur darstellen. Für die ausgefüllte Darstellung gibt man als Linienstärke 0 an. Größere Werte führen zu einer Umrissdarstellung.

Beziérkurven

*Beziérkurven*⁴⁴ sind Linien, deren Länge und Krümmung durch vier Kontrollpunkte festgelegt werden. Zwei der Kontrollpunkte bestimmen die Enden der Linie, über die anderen beiden wird die Krümmung eingestellt. Wenn man das Prinzip nicht schon aus Grafik- oder CAD-Programmen kennt, versteht man die Funktionsweise nach einigen Experimenten sehr schnell, da die Bedienung recht intuitiv ist.

Die Endpunkte einer Bézierkurve werden durch die beiden Kontrollpunkte bestimmt, die nach dem Einfügen der Kurve oben liegen. Die beiden unteren Kontrollpunkte bestimmen die Krümmung der Kurve.

Eine gerade Linie bekommt man, wenn alle vier Kontrollpunkte auf einer Geraden liegen. Das erreicht man am einfachsten, indem man die beiden Kontrollpunkte für die Kurvenkrümmung auf die Endpunkte der Kurve schiebt.

Die Linienstärke 0 ist für Bézierkurven nicht geeignet, da es bei dieser Einstellung zu dem in Abb. 77 dargestellten Effekt kommt.

Pfeil

Bei der Pfeildarstellung gibt es außer den vier bereits von den Beziérkurven bekannten Kontrollpunkten noch sechs Schieberegler, mit denen man das Aussehen des Pfeils beeinflussen kann.

Zunächst zu den Kontrollpunkten: Die beiden mittleren bestimmen die Pfeilkrümmung, die beiden äußeren legen die Länge der Pfeillinie fest. Ein Pfeil ist im Prinzip nichts anderes als eine Bézierkurve mit aufgesetzter Pfeilspitze.

Es gibt allerdings einige Einstellmöglichkeiten, die bei Bézierkurven nicht verfügbar sind. Diese erreicht man mit Hilfe der Schieberegler, die oben rechts im Editorfenster zu sehen sind. Einen Eindruck von der Wirkung der Regler soll Abb. 79 vermitteln.

Der oberste Regler bestimmt die Breite der Pfeillinie am Ende des Pfeils, der zweite Regler entsprechend die Breite direkt am Ansatz der Pfeilspitze, wobei sich auch die Größe der Pfeilspitze verändert. Eine gleichmäßige Breite der Pfeillinie erreicht man, indem man beide Regler auf die gleiche Position einstellt.

⁴³ Bei der Eingabe von Werten zwischen 0 und 1 werden die Nachkommastellen durch einen Punkt, nicht durch ein Komma abgetrennt. Für einen Alpha-Wert von 0,5 muss man also 0.5 eingeben.

⁴⁴ nach Pierre Bézier, einem der Entwickler dieser Kurven

Der dritte Regler beeinflusst die Länge des Pfeils. Die Länge des Pfeils kann man zwar auch durch Verschieben der Endpunkte einstellen, aber die Veränderung wirkt im fertigen Briefing flüssiger, wenn man den Schieberegler verwendet. Dazu bestimmt man zunächst mit Hilfe der Kontrollpunkte die Form des Pfeils in der Länge, die er im Briefing maximal erreicht. Beim Erstellen der Animation regelt man dann die Länge nur noch über den Schieberegler.

Mit Hilfe der drei übrigen Schieberegler kann man die Form und Größe der Pfeilspitze beeinflussen. Der erste dieser Regler (der vierte von oben) dient zur Veränderung der Größe der Pfeilspitze, wobei alle Seiten der Pfeilspitze verlängert werden. Mit dem fünften Regler von oben kann man die Form der Pfeilspitze verändern. In der Stellung ganz links ist sie dreieckig. Durch Verschieben nach rechts werden die hinteren Ecken weiter nach hinten verschoben. Der letzte Regler dient zum Einstellen der Länge der Pfeilspitze.

Gibt man als Linienstärke 0 an, wird der Pfeil ausgefüllt dargestellt. Bei größeren Werten wird nur der Umriss gezeichnet.

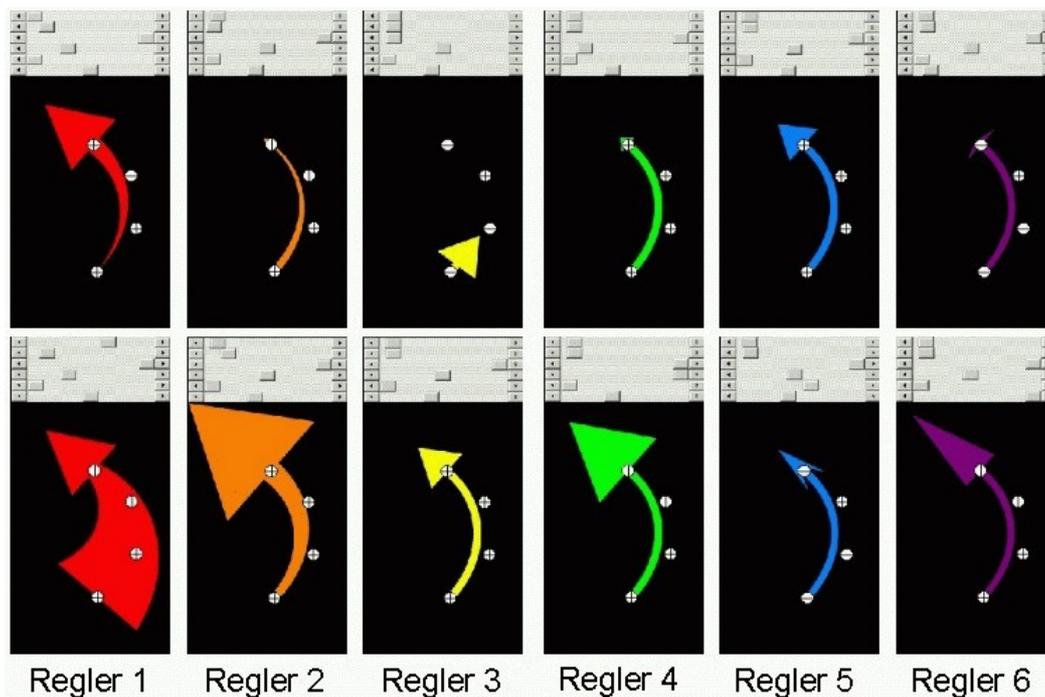


Abb. 79: Bearbeitung von Pfeilen (von links nach rechts: Oberer Regler – Breite am Pfeilende, Regler 2 – Breite am Pfeilanfang, Regler 3 – Pfeillänge, Regler 4 – Größe der Pfeilspitze, Regler 5 – Form der Pfeilspitze, Regler 6 – Länge der Pfeilspitze)

Quadrat

Hinter der Form, die in der Auswahlliste als Quadrat aufgeführt wird, verbirgt sich eine Form, die zur Darstellung beliebiger Vierecke verwendet werden kann.

Die Eckpunkte des Vierecks werden jeweils durch einen Kontrollpunkt festgelegt. Durch Verschieben der Kontrollpunkte verändert man Form und Größe des Vierecks.

Wie bei den Figuren wird das Viereck bei Wahl der Linienstärke 0 ausgefüllt, bei größeren Werten werden nur die Umrisslinien gezeichnet.

Verteidigungszonen

Zur Darstellung von *Verteidigungsstellungen* gibt es die Formen VERTEIDIGUNG_Linie, VERTEIDIGUNG_Quadrat und VERTEIDIGUNG_Region. Die Handhabung entspricht bei VERTEIDIGUNG_Linie derjenigen bei Bézierkurven, bei VERTEIDIGUNG_Quadrat der unter Quadrat beschriebenen.

Auch bei VERTEIDIGUNG_Region werden Ausrichtung und Form mit Hilfe von vier Kontrollpunkten festgelegt. Die beiden Kontrollpunkte, die direkt auf der Linie liegen, bestimmen Größe und Ausrich-

tung der Region. Mit den anderen beiden lässt sich die Krümmung der Linien einstellen. Nach kurzem Ausprobieren sollte das Prinzip klar werden.

Über den oberen der beiden Schieberegler kann man die Anzahl der „Zähne“ der Verteidigungslinie und über den unteren die Länge dieser Zähne einstellen.

Die Linienstärke 0 ist für die Verteidigungszonen nicht geeignet, da es bei dieser Einstellung zu dem in Abb. 77 dargestellten Effekt kommt.

Text

In der Auswahlliste für Formen ist auch ein Eintrag TEXT zu finden. Leider ist diese Form nicht sinnvoll verwendbar, da der Standardtext sich anscheinend nicht ändern lässt.

8.5 Animation erstellen

8.5.1 Prinzip der Animation

Hinter den *Briefing-Animationen* steht das gleiche Prinzip wie bei Filmen. Einzelbilder (sog. *Frames*) werden schnell nacheinander angezeigt⁴⁵, so dass aufgrund der Trägheit des menschlichen Auges für den Betrachter der Eindruck von fließender Bewegung entsteht. Dabei unterscheiden sich zwei aufeinander folgende Bilder nur unwesentlich.

Da es sehr aufwendig ist, die für eine längere Animation benötigte Anzahl von Einzelbildern von Hand zu zeichnen, bietet der Briefing-Editor die Möglichkeit, die Einzelbilder zum größten Teil automatisch zu erzeugen. Der Benutzer muss dann nur noch die Schlüsselszenen (*Keyframes*) der Animation festlegen. Die zwischen zwei Keyframes liegenden Bilder werden dann vom Computer automatisch berechnet.

Soll sich z.B. ein Flugzeug in gerader Linie fortbewegen, gibt man als Keyframes die Position des Flugzeugs am Anfang und am Ende der Bewegung an. Zusätzlich wird die Anzahl der gewünschten Zwischenbilder benötigt. Der Computer teilt dann die Entfernung zwischen den beiden Positionen in entsprechend viele Teilschritte auf und bewegt das Flugzeug von Bild zu Bild jeweils um einen solchen Schritt weiter.

Für das Briefing muss man also überlegen, welche Frames als Keyframes dienen können. Für kontinuierliche Veränderungen gibt man wie beschrieben Anfangs- und Endzustand sowie die Anzahl der Zwischenschritte an. Je mehr Zwischenschritte zwischen den Keyframes liegen, desto länger dauert die Animation. Dieses Vorgehen bietet sich z.B. für Bewegungen, Größen- und Formänderungen an.

Manchmal ist diese Art kontinuierlicher Veränderung jedoch nicht erwünscht. Ein Beispiel wäre eine Verteidigungszone, die ihre Farbe ändert, um die Einnahme durch einen anderen Spieler anzudeuten. Würde man hier mehrere Bilder zwischen den Keyframes erzeugen, würde sich ein Überblenden von einer Farbe zu einer anderen ergeben. Besser wäre in diesem Fall aber ein abrupter Übergang. Um das zu erreichen, setzt man einfach die beiden Keyframes direkt aneinander.

Ich verwende für den Entwurf von Briefing-Animationen ein Planungsraster, in dem die Keyframes eingetragen werden⁴⁶. Für jede Form gibt es eine Zeile, in der der jeweilige Zustand (z.B. die Farbe) der Form markiert wird. Einen Ausschnitt aus einem ausgefüllten Planungsraster sieht man in Abb. 80. In der Mission, für die dieses Briefing erstellt wurde, werden Fallschirmjäger in der Nähe einer feindlichen Verteidigungsstellung abgesetzt, die erobert werden soll.

⁴⁵ Ein üblicher Wert sind 24 Bilder pro Sekunde. Dieser Wert kann auch für die Briefing-Animationen als Richtwert dienen. Für eine 10 Sekunden dauernde Animation sind also ca. 240 Frames zu erzeugen.

⁴⁶ Die Verwendung des Planungsrasters ist wieder nur als Vorschlag zu verstehen. Natürlich kann man auch ganz anders vorgehen. Am Ende des User Guides ist das Raster als Kopiervorlage zu finden.

Ein entsprechender Dialog öffnet sich nach Auswahl von Animation > Keyframe löschen. Hier muss man die Nummer des zu löschenden Keyframes eingeben.

Manchmal kann es notwendig sein, einen Keyframe innerhalb der Animation zu verschieben. Dazu wählt man den Menüpunkt Animation > Keyframe bearbeiten aus. In dem daraufhin eingeblendeten Dialog (vgl. Abb. 83) gibt man links die Nummer des Keyframes an, der verschoben werden soll. In dem rechten Feld wird dann die Nummer der neuen Position in der Animation angegeben. Eine typische Situation ist z.B. eine Animation, die zu schnell abläuft. Soll sie auf die halbe Geschwindigkeit gebracht werden, reicht es aus, einfach die Nummern aller Keyframes mit zwei zu multiplizieren. Man verschiebt also z.B. den Keyframe mit der Nummer 20 an die Stelle des Frames mit der Nummer 40, wie es in Abb. 83 dargestellt ist.



Abb. 83: Dialog zum Verschieben eines Keyframes

Neben den gerade beschriebenen einzelnen Menübefehlen gibt es die gleichen Funktionen noch einmal im Dialog Keyframes bearbeiten (vgl. Abb. 84), der über Ansicht > Dialog bearbeiten dauerhaft eingeblendet werden kann und gegenüber den einzelnen Befehlen ein komfortableres Arbeiten erlaubt. Die Buttons Hinzufügen, Löschen und Bewegen entsprechen den oben beschriebenen Menübefehlen. Der mit Frame-Liste beschriftete Schieberegler ermöglicht die bequeme Auswahl eines der Keyframes der aktuellen Form, aber auch einzelner dazwischen liegender Frames. So kann man z.B. einen Frame auswählen und durch einen Klick auf Hinzufügen als Keyframe markieren.

Die Arbeit mit diesem Dialog ist u.a. deswegen so komfortabel, weil bei einem Verschieben des Reglers nicht nur für die aktuelle, sondern für alle in der Animation vorhandenen Formen jeweils der eingestellte Frame angezeigt wird. Für die aktuelle Form sind außerdem alle Keyframes (dargestellt als Markierungen an der Skala) auf einen Blick zu sehen.



Abb. 84: Dialog zum Bearbeiten von Keyframes

Die Änderungen an der aktuellen Form wirken sich immer auf den im Dialog eingestellten Keyframe aus. Mit Hilfe dieses Dialogs kann man so ganz bequem den Ablauf der Animation von Keyframe zu Keyframe „abspielen“ und die aktuelle Form jeweils nach Wunsch anpassen.

Den Ablauf der Animation im Originaltempo kann man sich über Animation > Animieren anschauen. Die laufende Animation kann über Animation > Animation stoppen abgebrochen werden.

8.5.3 Animation erzeugen und speichern

Über Datei > Sequenz speichern und Datei > Sequenz laden kann man die erstellte Animation speichern bzw. laden. Die auf diese Weise gespeicherte Datei kann aber nicht in die Mission eingebunden werden. Dazu gibt es einen weiteren Menüpunkt, nämlich Animation > Animation rendern.

Wählt man diesen Menüpunkt aus, erscheint der in Abb. 85 dargestellte Dialog. In dem oberen Feld wählt man nach Klick auf den Button Durchsuchen... das Verzeichnis der Mission aus, zu der das Briefing gehört⁴⁷. Dort werden die Dateien mit den Animationsdaten gespeichert.

⁴⁷ z.B. C:\Programme\SSRW\Run\Editor\maps.src\map.010\mis.001

In den grauen Feldern in dem mit Bis zu drei Teilungspunkte wählen beschrifteten Rahmen wird ganz links der erste, ganz rechts der letzte Frame der Animation angezeigt. Diese Werte kann man in dem Dialog nicht ändern. In den drei Feldern dazwischen kann man die *Teilungspunkte* der Animation festlegen. Diese Teilungspunkte geben die Einzelbilder an, an denen die Animation in Teilanimationen zerlegt werden soll.

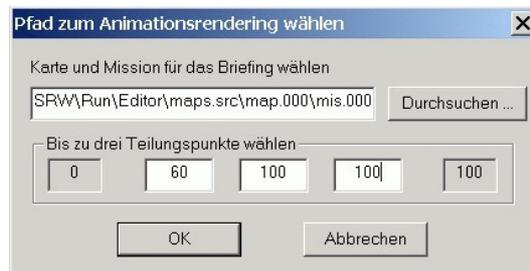


Abb. 85: Dialog zum Fertigstellen der Briefing-Animation

Teilungspunkte werden benötigt, wenn im Briefing vor dem Spiel die Teilziele über die Buttons rechts unten neben der Briefingkarte (vgl. Abb. 33) abgerufen und zusammen mit den entsprechenden Abschnitten der Animation angezeigt werden sollen.

Die Animation, die in Abb. 85 als Beispiel dient, besteht aus den Frames 0 bis 100. Es werden zwei Teilanimationen angelegt, von denen die erste bei Frame Nr. 0, die zweite bei Frame Nr. 60 beginnt. Die beiden Teilanimationen bestehen also aus den Frames 0 bis 59 bzw. 60 bis 100.

Werden weniger als drei Teilungspunkte benötigt, trägt man in den nicht verwendeten Feldern jeweils die Nummer des letzten Frames (hier also 100) ein. Dies ist notwendig, damit die Animation und die Buttons für die Teilziele im Spiel korrekt angezeigt werden.

8.5.4 Einbinden der Briefing-Animation in eine Mission

Wenn man die Animation über Animation > Animation rendern im richtigen Verzeichnis erzeugt hat, findet man dort für jede Teilanimation die Dateien partX.dat und partX.off. An der Stelle des X steht je nachdem, zu welcher Teilanimation die Dateien gehören, die Ziffer 0, 1, 2 oder 3. Diese werden automatisch eingebunden, wenn man im Karten- und Missionseditor die spielbare Mission erzeugt.

9 Anhang

9.1 Gegenüberstellung SSRW und SuStII

Da die Benutzeroberfläche des SSRW-Editors in Englisch, die des SuStII-Editors in Deutsch gehalten ist, habe ich hier eine „Übersetzung“ der Menüs angegeben (links: SSRW, rechts: SuStII). In SSRW sind einige Menüpunkte dazugekommen, zu denen es in SuStII keine Entsprechung gibt. Diese sind mit --- in der rechten Spalte gekennzeichnet. Kursiv gedruckte Menüeinträge sind in SSRW gegenüber SuStII in anderer Reihenfolge angeordnet.

File

Show map info
Wave map
Exit

Map

New map
Save map
Load map
Generate bitmap
Merge map

Mission

New mission
Build as campaign mission
Build as multiplayer mission
Build as single mission
Save mission
Load mission
Clear mission
Run game
Generate bitmap

Mission options

Players
Scripts
Reinforcements
Groups
Objectives
Phrases
Generate tactical map
Settings

View

Toolbar
Status bar
Minimap
Tool tree
Pic bar

Help

About edit3...

Datei

Karteninfo zeigen

Beenden

Karte

Neue Karte
Karte speichern
Karte laden
Bitmap generieren

Mission

Neue Mission
Als Kampagnenmission generieren
Als Mehrspielermission generieren
Als Einzelspielermission generieren
Mission speichern
Mission laden
Mission löschen

Missionsoptionen

Spieler
Skripte
Verstärkungen
Gruppen
Ziele
Meldungen
Taktische Karte generieren
Eigenschaften

Ansicht

Werkzeugleiste
Statusleiste
Minikarte
Werkzeuge
Symbolleiste

Hilfe

Über edit3...

9.2 Fehlermeldungen

In diesem Abschnitt sind einige Fehlermeldungen des Editors zu finden, bei denen ich rekonstruieren konnte, aus welchem Grund sie angezeigt wurden.

9.2.1 yy===-1 Internal Error

Diese Fehlermeldung (vgl. Abb. 86) ist mir begegnet, als ich versucht habe, aus einer Mission mit einem Skript mit unvollständiger Bedingung eine spielfähige Version zu erzeugen. Als das Skript gelöscht war, gelang dies ohne Probleme.

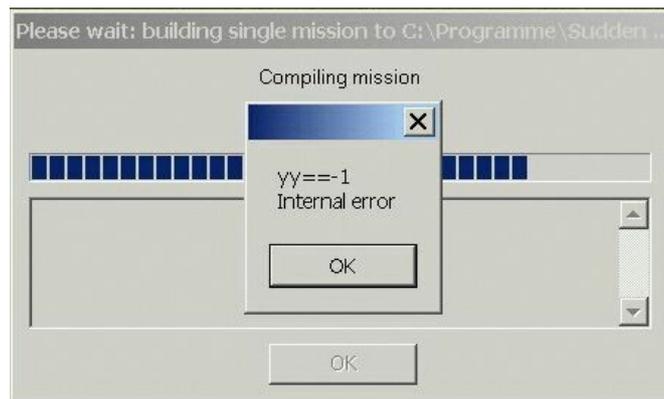


Abb. 86: Fehlermeldung beim Kompilieren einer Mission mit unvollständigem Skript

Es scheinen dabei aber auch noch andere Aspekte eine Rolle zu spielen, denn der Fehler ließ sich mit einem anderen unvollständigen Skript nicht reproduzieren.

9.2.2 `inpCur[curW]>=inpNum[curW]`

Ein angelegtes, aber weder mit Bedingungen noch Aktionen versehenes Skript führte zu der in Abb. 87 dargestellten Fehlermeldung.

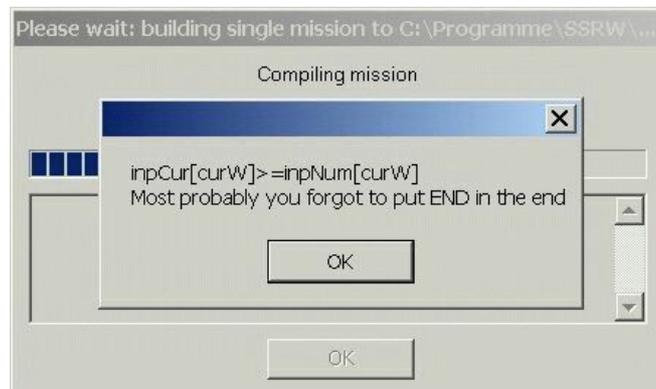


Abb. 87: Fehlermeldung beim Kompilieren mit einem „leeren“ Skript

9.2.3 Fehler beim Speichern

Wenn man eine Karte speichern will, aber das Verzeichnis `maps.src` fehlt, dann erscheint die folgende Fehlermeldung (vgl. Abb. 88). In diesem Fall legt man das Verzeichnis an und speichert die Karte anschließend erneut (vgl. 3.1.2).



Abb. 88: Fehlermeldung beim Speichern einer Karte, wenn `maps.src` fehlt

9.3 Bugs

Die hier aufgeführten Bugs wurden zum Teil bereits an anderer Stelle beschrieben. Sie sind hier nur der Vollständigkeit halber noch einmal erwähnt. Generell sei hier noch einmal die ausdrückliche Empfehlung gegeben, bei der Arbeit mit dem Editor möglichst oft zu speichern, da er nicht stabil läuft und aus vielen Gründen abstürzen kann.

9.3.1 Mehrfaches Zählen von Einheiten

Dieser Fehler tritt manchmal beim Erstellen von Ladungen im Einheiteneditor sowie beim Plazieren von Einheiten auf der Karte auf. Er wurde schon in 4.3 und 5.1.1 beschrieben.

9.3.2 Einfrieren beim Löschen der letzten Einheit

Beim Löschen der letzten Einheit in der Missionszone friert der Editor manchmal ein (vgl. 5.1.5).

9.3.3 Einfrieren bei Anlegen und Löschen großer Zonen

Wenn man große Zonen mit gedrückter Maustaste anlegt oder löscht, kann es ebenfalls zu einem Einfrieren des Editors kommen.

9.3.4 Optionen zum Verhaltensmodell Maintenance – Repair u.a.

Bei der Einstellung der Optionen zu einigen Verhaltensmodellen funktioniert die Anzeige der Optionen nicht korrekt (vgl. 6.5.1 und folgende Abschnitte).

Index

Ackerflächen.....	26	say phrase.....	62
Airfields (Auswahlfeld).....	20	say phrase with pointer to object.....	62
Aktion.....		send planes of group to airfield.....	71
add all player's units to force.....	72	send planes of player into location and land to airfield.....	70
add all units to force.....	72	send planes of player to object and land to airfield.....	70
add flights to player.....	70	send reinforcement.....	69
add from location to force.....	72	set AI.group behaviour.....	67
add group to force.....	72	set AI.group group1.....	68
add number of units in group to cell.....	73	set AI.group group2.....	68
add patrol locations for group.....	68	set AI.group location1.....	67
add planes to player.....	70	set AI.group location2.....	67
add route point over object.....	71	set attributes for units in group.....	67
add strike point over object.....	71	set attributes for units in zone.....	67
clear force.....	72	set attributes for units of player in group...67	
clear patrol locations for group.....	68	set attributes for units of player in zone....67	
deliver group to player.....	68	set new group for units in location.....	68
exclude this condition.....	62	set next mission.....	71
fire a volley by Howitzer of group into object.....	68	set timer for all mines in zone.....	74
fire a volley by Howitzer of group into zone	68	set timer period.....	65
fire a volley by Katyusha of group into object.....	68	shift patrol locations for group.....	68
fire a volley by Katyusha of group into zone	68	show modal dialog.....	63
fire a volley by V2 of group into zone.....	68	show modal dialog with pointer to object..63	
form reserve from force.....	72	show object for player.....	64
form reserve from force and set group.....	72	start avia track.....	71
kill units in location through flag.....	69	start countdown.....	65
kill units of group through flag.....	69	start planes of player on avia track and land to airfield.....	71
let cell contain value.....	73	start timer.....	65
mathematical operation.....	73	stop timer.....	65
move screen to location.....	64	terminate mission.....	71
put units from force into location.....	72	Aktionen.....	35
put units from force into location and set group to them.....	72	Alpha-Wert.....	81
resurrect units.....	69	and (Operator).....	60
save mission.....	72	Animation.....	
		abbrechen.....	85

abspielen.....	85	Bildbearbeitung.....	78
erzeugen.....	85	Blockierung.....	19, 22, 26
laden.....	85	Boolesche Logik.....	59
Attribute.....	67	Boolesche Operatoren.....	59, 60
Auswahlleiste.....	10	Briefing.....	9
Automatische Generierung.....	18	Missionsziele.....	30
Bahnlinien.....	18, 19	Briefing-Animation.....	83
Bedingung.....		Briefing-Editor.....	
AI.group behaviour.....	67	Formen.....	79
AI.group group1.....	67	Missionsziele.....	31
AI.group group2.....	67	Teilungspunkte.....	30
AI.group location1.....	67	Briefing-Editor (anied.exe).....	76
AI.group location2.....	67	Brightness (Schieberegler).....	15
countdown.....	65	Brush size (Schieberegler).....	14
group ammo/fuel.....	67	Bugs.....	89
group was attacked some time ago.....	68	Build (Button).....	18
mission start.....	64	Cliff type (Auswahlfeld).....	18
object is dead.....	74	Container.....	72
object is fully repaired.....	74	Curv priority (Schieberegler).....	19
player kept flights.....	70	Damage (Schieberegler).....	17
player kept planes.....	70	Delete last node (Button).....	18
time from start of the mission is.....	64	Deselect (Button).....	24
timer elapsed.....	65	Dialogbox.....	63
units in group.....	66	Editor (edit3.exe).....	8
units in group in per cents of units in group	66	Editorfenster.....	
units of force.....	72	Briefing-Editor.....	76
units of group in location.....	66	Karteneditor.....	10
units of group in location per cents.....	66	Einheiten.....	
units of player.....	66	aktiv.....	66
units of player in group.....	66	auswählen.....	40
units of player in location.....	66	bearbeiten.....	40
units of player in location per cents.....	66	Gruppen.....	40, 42
units of player in per cents.....	66	Guns.....	66
value in cell.....	73	in Folgemission übertragen.....	72
Bedingungen.....		in hell.....	66
Klammern.....	61	löschen.....	42
Beziérkurven.....	81	plazieren.....	39
		Transfer.....	68

unterwegs.....	66	Gruppen.....	39
Verhalten.....	40, 48, 67	Gruppenübersicht.....	39, 42
Einheiteneditor.....	39	Handbuch.....	
Benutzeroberfläche.....	32	Editor-Fibel (Irrläufer).....	2, 29, 69
Besatzung.....	33	Kampagnentutorial (Slider).....	2, 36
Bug.....	33	Originaldokumentation zu SuSt II.....	2, 29
Flugzeugladungen festlegen.....	30	Tutorial zum Briefing-Editor (Lindbergh)...	2
Passagiere.....	33	Hangar.....	20
Verstärkungen.....	42	Häuser besetzen.....	51
Einheitenklasse.....	66	Helligkeit (Terrain).....	15
Einzelspielermission.....	36	variieren.....	16
Exact end (Schieberegler).....	19	Hintergrundbild.....	78
F1 (Map Info).....	17	Icons.....	
F3 (Flags einblenden).....	21, 22, 23	Briefing-Editor.....	77
F4 (Objekte ausblenden).....	17	Karteneditor.....	11
F5 (Hangarmarkierungen einblenden).....	20	Infanterie.....	
Fahne.....	44	transportieren.....	54
Fehlermeldungen.....	87	info.txt (Kampagneninfo).....	36
Felder.....	26	Kampagne.....	36
Figuren.....	81	erzeugen.....	37
Flags.....	21	Infodatei.....	36
Sichtbarkeit.....	22	Skript.....	36
Flugplatz.....	19	Kampagnenmission.....	36, 71
Flugplatzgelände.....	20	Karte.....	8
Flugroute (Skriptbefehle).....	71	anlegen.....	13
Flugzeuge.....	30	Bitmap erzeugen.....	14
als Verstärkung.....	43	laden.....	14
Flugzeuge (Skriptbefehle).....	70	Raster.....	21
Folgemission.....	71	speichern.....	13
Force.....	72	taktische.....	31
Form.....		Kartenausschnitt.....	10
bearbeiten.....	80	Kartengröße.....	13
einfügen.....	79	Kartenschema.....	13
Linienstärke.....	80	Keyframe.....	76, 79, 83
Frame-Liste (Schieberegler).....	85	bearbeiten.....	84
Fullstop.....	24	erzeugen.....	84
Gelände.....	14	löschen.....	85
Geschütze transportieren.....	53	verschieben.....	85

KI-Gruppe.....	40, 67	löschen.....	18
KI-Zone.....	40, 67	plazieren.....	16
Klippen.....	18, 19, 22	Random.....	16
Kontrollpunkte.....	80	Schadenzustand.....	17
Kriegsnebel aufheben.....	64	Objektblock.....	24
Landmarkierung.....	20	Beispiele.....	25
Langstreckenfeuer (Skriptbefehle).....	68	laden.....	25
Load at center (Button).....	25	markieren.....	24
Location (Auswahlfeld).....	45	speichern.....	24
Map.....	8	Objekte.....	
Marker.....	45	Anzahl.....	17
Menüstruktur.....		ausblenden.....	17
Karteneditor.....	87	statische.....	16
Minen (Skriptbefehl).....	74	or (Operator).....	60
Minenfelder.....	45	Patrouille.....	68
Minikarte.....	10	Pfeil.....	81
Minimap.....	10	Pic height (Schieberegler).....	17
Anzeige umschalten.....	12	Pic width (Schieberegler).....	17
Mission.....	8, 28	Pinselgröße.....	14, 23
anlegen.....	28	Pontonbrücken.....	23
Bitmap erzeugen.....	29	bauen.....	56
Briefing-Animation einbinden.....	86	verbieten.....	23
laden.....	29	Quadrat.....	82
löschen.....	29	Radiergummi.....	18
Name.....	32	Reverse (Button).....	19
speichern.....	29	Sel all (Button).....	24
Missionsbeginn (Skriptbefehl).....	64	Sel mis (Button).....	24
Missionseinstellungen.....	31	Shallows.....	23
Missionsende (Skriptbefehl).....	71	Sichtlücken.....	22
Missionsgröße.....	28	Skript.....	8, 59
Missionstexte.....	46, 62	Skriptausführung (Reihenfolge).....	34
Missionsziele.....	30	Skripteditor.....	34
Missionszone.....	28	Verwaltung.....	34
Missionszone.....		Skriptprogrammierung.....	9, 34, 59
Rand.....	17	Aktionen.....	59
No waves	23	Bedingungen.....	35, 59
not (Operator).....	59	Operatoren.....	35
Objekt.....		Soundeffekte.....	46, 47

Soundeffekte.....	Objekte reparieren.....	55
Bug.....	Verstärkungen.....	42
Bug.....	bearbeiten.....	42
Spielereinstellungen.....	Einschränkungen.....	43
Spielstand speichern.....	in Mehrspielermissionen (Skriptbefehle)...	69
Startmarkierung.....	Tore.....	44
Straßen.....	vorgefertigte.....	43
Sue-Tool (Sue.exe).....	Verstärkungen (Skriptbefehl).....	69
Teameinstellungen.....	Verteidigungsstellungen.....	82
Teilungspunkte.....	Werkzeugbaum.....	10
Testen einer Mission.....	Airfields.....	19
Textausgabe.....	Automatic generation.....	18
Textur.....	Bridges.....	16
Timer.....	Cliffs.....	16
Countdown.....	Craters.....	16
einfacher.....	Eraser.....	18
periodischer.....	Fences.....	16
Tore.....	Free objects.....	16
Trigger.....	Gates.....	44
Verbündete.....	Horizontal objects.....	16
Verhaltensmodell.....	Houses.....	16
Infantry - Defend Location.....	Lighten/darken.....	15
Infantry - Gun crew.....	Locations.....	45
Infantry - Pursue.....	Markers.....	45
Katyusha - Move to location.....	Mines.....	45
Maintenance - Repair.....	Other flags.....	22
None.....	Roads.....	16
Planes - Move to location.....	Save/load objects.....	24
Reconnaissance.....	Set brightness.....	15
Tanks - Defend location.....	Sounds.....	46
Tanks - Pursue.....	Terrain.....	14
Trains - Defend location.....	Trees.....	16
Trucks - Help.....	Units.....	39
Trucks - Move to location.....	Vertical objects.....	16
Trucks - Transport from location to location	Visibility.....	22
.....	Wetter.....	32
Verhaltensoptionen.....	Wiederbelebung von Einheiten.....	
Versorgungsfahrzeuge.....	Leben.....	33, 41
Bau von Pontonbrücken.....		

Optionen.....	41	Zone verteidigen.....	
Skriptbefehle.....	69	Infanterie.....	49
Zelle (Skript).....	72	Panzer.....	52
als Flag.....	73	Zug.....	57
als Zähler.....	74	Zonen.....	44
Zelle (Teilfeld).....	21	□-Wert.....	81

Zonenübersicht (Locations)

Kampagne: Mission:

0	1	2	3	4	5	6	7
Gesamte Missionszone							
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55
56	57	58	59	60	61	62	63

[] **Timer** [] **Zellen** [] **Marker** **Kampagne:** **Mission:**

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29
30	31	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	48	49
50	51	52	53	54	55	56	57	58	59
60	61	62	63	64	65	66	76	86	96
70	71	72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87	88	89
90	91	92	93	94	95	96	97	98	99

