

Surroundings

Praktikum Interaktive Medien

WS 2005 | 2006

Barbara Vogelmann

13381 | Medieninformatik

Betreuer: Prof. Dr. Hahn | Beate Schlitter

Inhaltsverzeichnis:

1. EINLEITUNG	4
1.1. MOTIVATION	4
1.2. MITWIRKENDE	4
1.3. VERWENDETE PROGRAMME	4
1.4. SELBSTDEFINIERT ANFORDERUNGEN	4
2. STORY	5
2.1. INTRO	5
2.2. SEQUENZ 1	5
2.3. SEQUENZ 2	5
2.4. SEQUENZ 3	5
3. STRUKTUR	6
3.1. WORKFLOW - DIAGRAMM	6
3.2. GROBER AUFBAU	7
4. AUFBAU DES SPIELS UND DIE EINZELNEN FILME.....	8
4.1. INT.DIR	8
4.2. IN.DIR	8
4.3. INTRO.DIR	8
4.4. MENUE.DIR	9
4.5. SEQUENZ1.DIR	9
4.6. ALL.DIR – LEVEL 1	10
4.7. SEQUENZ2.DIR	10
4.8. ERDE.DIR – LEVEL 2	11
4.9. SEQUENZ3.DIR	11
4.10. WASSER.DIR – LEVEL 3.....	12
4.11. OUTRO.DIR.....	12
4.12. ENDE.DIR	12
4.13. SOUNDMENUE.DIR	12
4.14. HILFE / VOLUME.....	13
5. SPIELFUNKTIONEN	14
5.1. ALLGEMEIN.....	14
5.2. LEVEL 1	14
5.3. LEVEL 2	14
5.4. LEVEL 3	14
5.5. HIGHSCORE.....	15
6. PROGRAMMIERUNG IN LINGO	16
6.1. SKRIPTE DER EINZELNEN FILME.....	16
6.1.1. <i>int.dir</i>	16
6.1.2. <i>intro.dir</i>	17
6.1.3. <i>menue.dir</i>	17
6.1.4. <i>all.dir</i>	20
6.1.5. <i>erde.dir</i>	25
6.1.6. <i>wasser.dir</i>	27
6.1.7. <i>Hilfe / Volume</i>	29
6.1.8. <i>Soundmenue.dir</i>	29
6.2. EXTERNE CASTS (BESETZUNGEN).....	30
7. PROJEKTOR ERSTELLEN	32
8. PROBLEME.....	33

8.1.	SONDERZEICHEN	33
8.2.	HINTERGRUND.....	33
8.3.	PLATTFORMUNABHÄNGIGE PFADANGABEN.....	33
8.4.	SCHRÄGER LAUTSTÄRKEREGLER	33
8.5.	QUIT.....	34
8.6.	OBJEKTE IN OOP.....	34
8.7.	HIGHSCORE XTRA.....	34

Abbildungsverzeichnis:

Abbildung 1:	Fluss-Diagramm.....	6
Abbildung 2:	Screenshot Hauptmenü.....	9
Abbildung 3:	Screenshot von All.dir – Level 1.....	10
Abbildung 4:	Screenshot von Erde.dir – Level 2.....	11
Abbildung 5:	Screenshot von Wasser.dir – Level 3.....	12
Abbildung 6:	Screenshot von Soundmenue.dir.....	33

Skriptverzeichnis:

Skript 1:	Movieskript: Lautstärkeregelung in int.dir	16
Skript 2:	Movieskript in intro.dir	17
Skript 3:	Movieskript in menue.dir	20
Skript 4:	Initialisierungsskript in all.dir	21
Skript 5:	Movieskript in all.dir	23
Skript 6:	Enemyskript in all.dir.....	24
Skript 7:	Besatzungsskript in all.dir	25
Skript 8:	Auszüge aus dem Movieskript in erde.dir.....	25
Skript 9:	Auszüge aus dem Enemy2-Skript in erde.dir	26
Skript 10:	Freund-Skript in erde.dir.....	27
Skript 11:	Auszüge aus dem Initialisierungsskript in wasser.dir.....	28
Skript 12:	Auszüge aus dem Movieskript in wasser.dir	28
Skript 13:	Auszüge aus dem Enemy3-Skript in wasser.dir.....	28
Skript 14:	Auszüge aus dem Movieskript in all.dir / erde.dir / wasser.dir	29
Skript 15:	Movieskript in soundmenue.dir.....	30
Skript 16:	Auszüge aus dem Skript hilf wandern in hilfvol.cst	31
Skript 17:	Auszüge aus dem Movieskript in soundmenue.dir.....	34

1. Einleitung

1.1. Motivation

CD-ROM-Produktionen verschwinden zunehmend immer mehr vom Markt. Früher waren sie ein beliebtes Medium für Produktvorstellungen oder Spiele. Durch die zunehmende Verbreitung des Internet und vor allem der zunehmenden Übertragungsraten, werden die CD-ROM -Produktionen immer mehr durch Flash verdrängt.

Deswegen wollte ich mich noch einmal mit Director und Lingo beschäftigen – vor allem dem objektorientierten Lingo, um ein Gefühl zu bekommen, was damit möglich ist oder nicht und welche Funktionalitäten durchführbar sind.

1.2. Mitwirkende

- Barbara Vogelmann: Idee und Programmierung
- Herwarth Konnerth: Design, Sounds, 3D-Animationen

Für die Realisierung dieser CD-ROM-Produktion für die Windowsplattform wurden ältere 3D-Animationen verwendet, um dem Spiel einen Rahmen zu geben.

1.3. Verwendete Programme

- Macromedia Director 7.0
- Photoshop 6.0
- Cinema 4 D
- Premiere
- Etc.

1.4. Selbstdefinierte Anforderungen

- Erstellen eines einfachen Director Spiels
- Einbetten von Multimedia wie 3D-Animationen, Filme, Sound, Bilder
- Programmierung mit Lingo OOP
- Realisierung eines Highscore
- Soundsteuerung mit schrägen Lautstärkereglern
- Abmessungen der Bühne: 800 x 600 Pixel, wegen den verwendeten 3D-Sequenzen, die noch kleiner sind.

2. Story

Es sollen die drei Elemente Luft, Erde und Wasser in diesem Spiel vorkommen und somit die menschliche Welt und ihren Bewegungsspielraum aufzeigen. Außenherum soll eine Mission simuliert werden.

Am Ende entdeckt man eine neue Galaxie Atlantis, die quasi als Belohnung winkt.

2.1. Intro

Im Jahr 3065 wissen die Menschen, daß sie nicht die einzigen Lebewesen im Weltall sind. Die Technik ist auf ein unvorstellbares Maß angestiegen und ihre Neugier hatte sie dazu gebracht, immer tiefer ins All vorzustossen um neue Welten und Galaxien zu erkunden. Doch noch immer gibt es Galaxien, die von feindlichen Lebewesen besetzt werden und die die Menschen davon abhalten, die dahinterliegenden Galaxien zu erforschen. Überwindet die Herrscher der Galaxien und Ihr werdet entdecken, welche Welt dahinter liegt!

2.2. Sequenz 1

Um die neuen Galaxien zu erreichen, mußt Du erst an Xano, dem Herrscher der nächsten Galaxie vorbeikommen!

Besiege seine Flotte und Du wirst Deinem Ziel ein Stück näher kommen.

Das ist Deine erste Mission!

2.3. Sequenz 2

Du hast Xano besiegt !

Es wartet bereits die zweite Mission!
Leider hat aber Weltraummüll Dein Schiff beschädigt.
Du mußt nun auf einem Planeten notlanden.
Die Xaragos helfen Dir aber, Dein Schiff zu reparieren!

Beschütze sie vor ihren Feinden und Du wirst Deinem Ziel noch ein Stück näher kommen.

2.4. Sequenz 3

Du hast es wieder geschafft!

Doch die dritte Mission erwartet Dich bereits!
Du erkundest jetzt eine - der Menschheit völlig unbekannte -Unterwasserwelt!
Doch der Herrscher Xanto dieses Wasserreiches will besänftigt werden.

Bekämpfe seine Feinde und er wird Dir als Dank sein Reich zeigen!

3. Struktur

3.1. Workflow - Diagramm

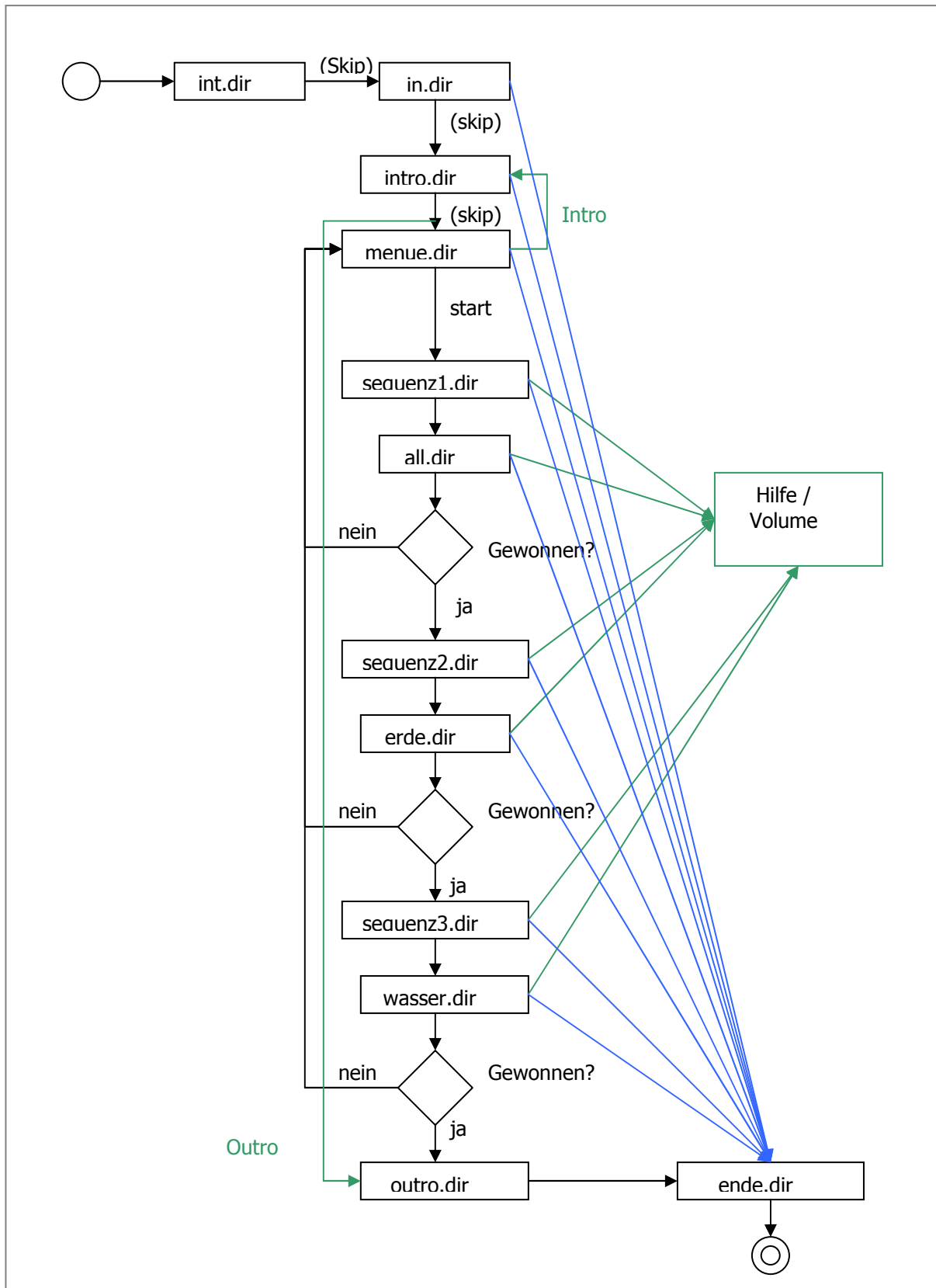


Abbildung 1: Fluss-Diagramm

3.2. Grober Aufbau

- Intro
- Hauptmenü
- Level 1- 3
- Highscore
- Outro

Dazwischen werden jeweils kleine Filme (3D-Sequenzen) abgespielt, um die Übergänge in die neuen Level zu „erzählen“.

4. Aufbau des Spiels und die einzelnen Filme

Im Folgenden werden alle Filme und deren Inhalte kurz beschrieben.

4.1. int.dir

Hier wird die Lautstärke des Rechners vor Start des Intros geregelt, damit die Lautstärke irgendwie in der Mitte eingepegelt wird. Leider funktioniert das ganze auf der Windowsplattform nicht 100%ig.

Dazu steht in der Director-Hilfe folgendes:

„In Windows wird die System-Soundeinstellung mit der Lautstärkeeinstellung der externen Lautsprecher verbunden. Aus diesem Grund kann die tatsächliche Lautstärke nach Einstellung der Sound-Stufe unterschiedlich ausfallen.“

4.2. in.dir

- beinhaltet den „Intro-Film“
- skip
- exit

4.3. intro.dir

Hier wird der Eingangstext im Schreibmaschinenstil gezeigt und dabei wird die Story ein bisschen erläutert.

Danach folgen wieder ein 3D-Film mit den üblichen Skip und Exit Buttons.

4.4. menu.dir

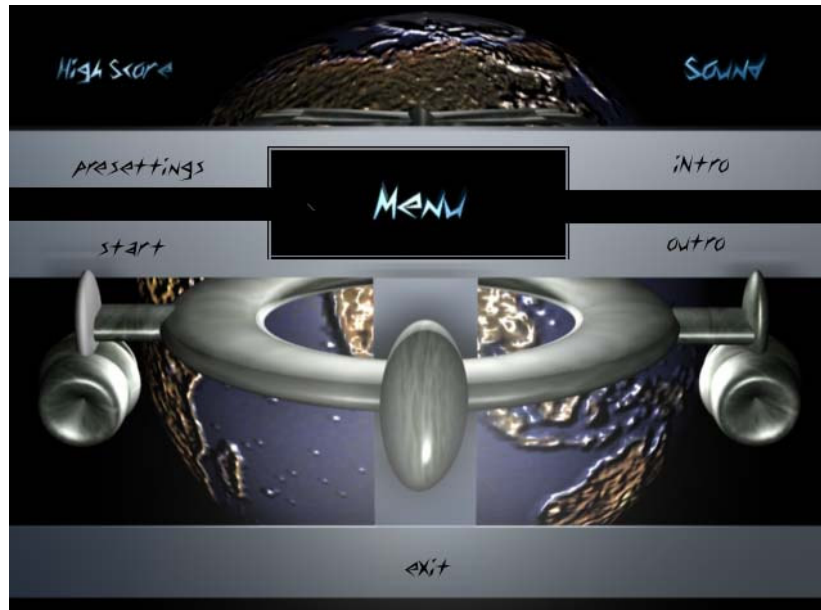


Abbildung 2: Screenshot Hauptmenü

Das Hauptmenü beinhaltet viele Funktionalitäten, um die Grundeinstellungen des Spiels zu überprüfen oder zu verändern:

- **Highscore:**
Hier kann man sich den aktuellen Punktestand anschauen. Bei Ende des Spiels gelangt man dorthin, um sich eintragen lassen zu können. Allerdings nur, wenn man alle 3 Level geschafft hat. Falls man verliert, gelangt man zurück zum Hauptmenü.
- **Sound:**
Hier kann man sich einen von drei Loops aussuchen, der einen dann als Hintergrundmusik begleitet
- **Presettings:**
Hier kann man die Spielschwierigkeit auswählen durch Auswahl eines Charakters. Zudem kann man aus 3 Waffen auswählen (mit dem Button Weapon), die den Sound beim Schuss im weiteren Spielverlauf darstellt. (verschiedene Schussgeräusche)
- **Intro:**
Hier gelangt man an den Intro-Film zurück
- **Outro:**
Hier kann man sich den Outro-Film gesondert anschauen, falls man die Level nie schafft
- **Start:**
Hiermit startet das Spiel mit einer Text-Zwischensequenz

4.5. sequenz1.dir

Hier wird zuerst wieder wie in Kapitel 2 beschrieben, der Text ausgegeben, der die Geschichte erzählt.

Daraufhin folgt wieder eine kurze 3D-Sequenz, die mit Skip übersprungen werden kann.

4.6. all.dir – Level 1

Im oberen Spielfeld wird der Mauszeiger als Cursor angezeigt. Dazu wurde ein transparenter Sprite auf die Bühne gelegt und eben im Skript festgelegt, dass innerhalb dieses Sprites der Cursor das Zielkreuz als Gestalt annimmt und außerhalb normal, um die Buttons bedienen zu können. Hier kann man zum Menü zurückkehren oder Volume und Hilfe aufrufen.

Zudem hat man im unteren Feld die Anzeige, wie viele Leben man noch hat, wie lange man noch spielen muss (Spielzeit), die erreichten Punkte und wie viel Schuss Munition man noch hat.

Dies ist für alle drei Level gleich. Es ändert sich in den Leveln nur die Spielschwierigkeit und andere Funktionen, die aber dann im entsprechenden Kapitel der Level beschrieben werden.

Danach kommt man zum ersten Level. Hier startet man das Spiel mit Klick auf die Bühne.



Abbildung 3: Screenshot von All.dir – Level 1

4.7. sequenz2.dir

Hier wird zuerst wieder wie in Kapitel 2 beschrieben, der Text ausgegeben, der die Geschichte erzählt.

Daraufhin folgt wieder eine kurze 3D-Sequenz, die mit Skip übersprungen werden kann.

4.8. erde.dir – Level 2



Abbildung 4: Screenshot von Erde.dir – Level 2

4.9. sequenz3.dir

Hier wird zuerst wieder wie in Kapitel 2 beschrieben, der Text ausgegeben, der die Geschichte erzählt.

Daraufhin folgt wieder eine kurze 3D-Sequenz, die mit Skip übersprungen werden kann.

4.10. wasser.dir – Level 3

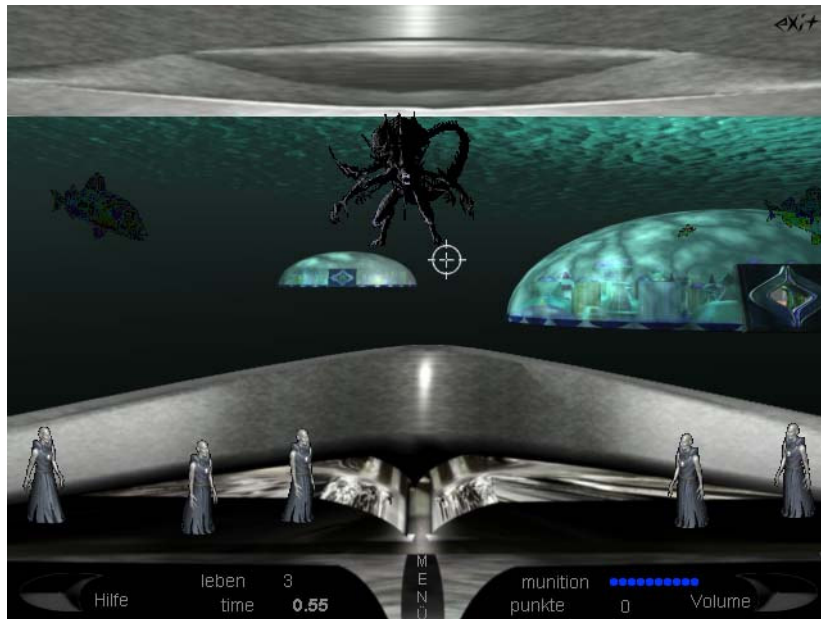


Abbildung 5: Screenshot von Wasser.dir – Level 3

4.11. outro.dir

- 3D film Atlantis
- skip
- ende

4.12. ende.dir

- Abfrage, ob man das Spiel wirklich beenden möchte: ja | nein
- Abspann mit Namen
- Quit am Ende des Movies, das den Projektor beenden soll

4.13. soundmenue.dir

Zuerst werden die Lautstärkeregler auf eine mittlere Lautstärke eingestellt und ebenso die zugehörige Position der Buttons.

Hier können dann während des Spiels die einzelnen Sounds geregelt werden.

Dazu kann man die Gesamtlautstärke, die Soundeffekte und die Musik getrennt regeln.

4.14. Hilfe / Volume

Diese Movies sind eigentlich keine „richtigen“ Movies sondern MIAW. MIAW bedeutet Movie in a Window.

Hier wird der Hilfetext bzw. das Soundmenü angezeigt, und zwar nicht auf der Bühne sondern außerhalb der Bühne im schwarzen Raum außerhalb der Bühne.

Das hat den Vorteil, dass auf der Bühne, die wegen der 3D-Sequenzen ohnehin etwas knapp bemessen ist, der gesamte Platz verwendet werden kann, und die Hilfe und das Soundmenü ausgelagert werden kann.

5. Spielfunktionen

5.1. Allgemein

Es müssen immer 1 Minute Spielzeit mit mindestens 1 Leben erreicht werden. Zusätzlich müssen mindestens 150 Punkte erreicht werden.

Man hat immer 10 Schuss Munition, die mit der Space-Taste aufgefüllt werden kann.

Wartet man zu lange, verliert man ein Leben, da man nur eine begrenzte Lebenszeit hat. (Der Feind greift sonst an und man verliert ein Leben).

5.2. Level 1

- Nur ein Feind Enemy
- Besatzung
- Lebenszeit (je nach Spielschwierigkeit leicht|mittel|schwer: random(120) + (600|480|360))

5.3. Level 2

- Cursor kleiner
- Man bekommt zusätzlich Freunde auf der Bühne, die den Feind nicht berühren dürfen
- Enemy 2
- Lebenszeit (je nach Spielschwierigkeit leicht|mittel|schwer: random(120) + (480|360|240))

5.4. Level 3

- Cursor wiederum kleiner
- Cursor Zittert
- Feind muss genau in der Mitte getroffen werden (transparenter Sprite)
- Lebenszeit (je nach Spielschwierigkeit leicht|mittel|schwer: : random(120) + (360|240|120))

5.5. Highscore

Wenn man alle drei Level erfolgreich absolviert hat, kann man sich in den Highscore eintragen.

Dazu kann man den Namen eintragen und dann werden die aktuellen Gesamtpunkte unter diesem Namen gespeichert.

Um die Punkte dauerhaft speichern zu können, muss man mit einem Xtra arbeiten, das es möglich macht, aus dem Directorspiel heraus auf der Festplatte eine Textdatei mit den Daten zu speichern.

Das wurde aber nicht implementiert aus folgenden Gründen:

Das Xtra FileIO, das man dafür benötigt, kostet extra Geld und stand mir leider nicht zur Verfügung. Des Weiteren muss man bedenken, dass auch Sicherheitsaspekten das Speichern von Daten vom Benutzer auf dessen Festplatte nicht erwünscht sein könnte.

Der Ausweg aus diesem Dilemma wäre eine Verbindung heraus aus dem Spiel ins Internet, um dort über eine kleine Datenbank die entsprechenden Daten abzuspeichern. Das könnte durchaus mit php realisiert werden.

Aus Zeitgründen konnte ich dies aber nicht realisieren und somit existiert nun nur die kleine Lösung, dass man eben pro laufende Session (d.h. solange der Projektor läuft) die Daten gespeichert werden kann. Damit hat man dann eine kleine Übersicht über den Spielestand, wenn man mehrmals hintereinander spielt.

6. Programmierung in Lingo

6.1. Skripte der einzelnen Filme

Im Folgenden werden Auszüge aus den Skripten der einzelnen Filme gezeigt. In den drei Levels werden dann im Folgenden nur noch die Änderungen angezeigt. Zudem werden die aufrufenden Skripte oder Steuerungsskripte wie Frameskripte nicht aufgeführt.

6.1.1. int.dir

- Movieskript:

```
--regelt die Lautstärke bevor auf der Bühne irgendetwas angezeigt wird

global gLaut1, gLaut2, gLaut3
on exitFrame
  gLaut1 = 3
  gLaut2 = 80
  gLaut3 = 80
  set the soundlevel = gLaut1
  set the volume of sound 1 = glaut2
  set the volume of sound 1 = glaut2
  set the volume of sound 3 = glaut3
  set the volume of sound 4 = glaut3
  set the volume of sound 5 = glaut3
  set the volume of sound 6 = glaut3
  set the volume of sound 7 = glaut3
  set the volume of sound 8 = glaut3

  go movie "@:data:movies:in.dir"
end
```

Skript 1: Movieskript: Lautstärkeregelung in int.dir

Unterschied von "the soundlevel" und "the volume of sound":

"the soundlevel":

Die Systemeigenschaft stellt die Lautstärke des Sounds zum Abspielen über die Lautsprecher Ihres Computers zwischen "kein Sound (0)" und der maximalen Lautstärke (7, Standard) ein.

"the volume of sound":

Diese Systemeigenschaft bestimmt die Lautstärke des Sound-Kanals, der in whichChannel angegeben ist. Die Sound-Kanäle sind als 1, 2, 3 usw. nummeriert. Kanäle 1 und 2 erscheinen im Drehbuch. Der Wert der Sound-Eigenschaft volume liegt zwischen 0 (Ton aus) und 255 (maximale Lautstärke). Ein Wert von 255 zeigt die Höchstlautstärke für den Computer an, die von der Eigenschaft soundLevel gesteuert wird, und niedrigere Werte werden im Verhältnis zu diesem Höchstwert skaliert. Mit dieser Eigenschaft können verschiedene Kanäle voneinander unabhängige Einstellungen innerhalb des zulässigen Wertebereichs haben.

Plattformübergreifende Pfadangabe:

```
go movie "@:data:movies:in.dir"
```


6.1.2. intro.dir

Eingangstext: Schreibmaschinenstil:

Man hat 2 fields „input“ und „output“. Nun wird nach und nach die einzelnen Buchstaben vom input field ins output field übertragen und dabei der Sound „pip“ abgespielt. Das ergibt den Eindruck, dass eine Schreibmaschine oder Computer den Text Buchstabe für Buchstabe eintippt.

Problem hier: durch die if-Schleife kann dieser Vorgang nicht geskript werden. Der Film reagiert auch nicht auf Escape oder sonstige Eingaben. Man muss daher warten, bis die Schleife abgearbeitet worden ist. Deswegen existiert auch während dieser Zeit kein Skip oder Exit Button auf der Bühne.

- Movieskript:

```
on startmovie
  put EMPTY into field "output"
end

on tapme
  put "" into field "output"
  repeat with x = 1 to the number of chars in field "input"
    set ch = char x of field "input"
    put ch after field "output"
    if ch = RETURN then
      nothing
    else
      if ch = " " then
        nothing
      else
        puppetsound 3, "pip"
      end if
    end if
    updatestage
    set waitin = 3
    starttimer
    repeat while the timer < waitin
    end repeat
  end repeat
end
```

Skript 2: Movieskript in intro.dir

6.1.3. menue.dir

- Movieskript:

```
global gWaffe, gCharakter, gWaffeSound ---Einstellungen-Var.

global gHighScore, Teil , gGesamt , gPunkte1 , gPunkte2 , gPunkte3 , ergebnisliste , eintrag
, gVar , Inhalt---highscore globale

on startmovie
  set gCharakter = 1 ---gruen = 1 , rot = 2 , blau = 3
  set gWaffe = 1
  set gWaffeSound = 1--Waffe 1 = 1 , Waffe 2 = 2 . Waffe 3 = 3

  if (gHighScore = void) then
    gHighScore = []
  end if

  gGesamt = gPunkte1 + gPunkte2 + gPunkte3 ---Für den Highscore
  the keyDownScript = "meinTastenSkript"
end
```

```

--#####HIGHSCORE #####-----

on Speichern
  LName = field "Name"-- Name und Punkte
  LPunkte = value (gGesamt)

  Eintrag = ["Name":LName,"Punkte":LPunkte]-- Liste erstellen

  if (gHighscore = Void) then
    gHighScore = []
  end if

  if gHighScore.count() < 10 then --wenn die Anzahl der Einträge in Liste gHighscore < 10
    gHighscore.add(Eintrag)
    put Empty into field "Name" -dann Liste mit neuem Eintrag füllen
  end if

  if gHighScore.count() = 10 then --bei 10 Einträgen prüfen, ob letzter größer als aktuelle
    gHighScore.deleteAt(gHighScore.count()) --wenn ja: streichen und den neuen einsetzen
    gHighscore.add(Eintrag)
    put Empty into field "Name"
  end if
end

on sortieren
  ergebnisliste = [] --leere liste erzeugen

  repeat while gHighScore > 0
    maxPunkte = -1 -- kleinster vorkommender Wert
    pos = 0
    repeat with x = 1 to gHighScore.count()
      if gHighScore[x]["Punkte"] > maxPunkte then
        maxPunkte = gHighScore[x]["Punkte"]
        pos = x
      end if
    end repeat
    ergebnisListe.add(gHighScore[pos])
    put ergebnisliste
    gHighScore.deleteAt(pos)
  end repeat

  gHighscore = ergebnisListe.duplicate()
end

on anzeigen
  put EMPTY into field "Ausgabe"
  repeat with i = 1 to gHighscore.count()
    put gHighscore[i]["Name"]&" " after field "Ausgabe"
    put gHighscore[i]["Punkte"]&RETURN after field "Ausgabe"
  end repeat
end

--nach Eintragen des Namens Return drücken, löst speichern, sortieren und anzeigen aus
on meinTastenSkript
  if the key = RETURN then ----checken, an welche Stelle im Highscore neuer Eintrag kommt
    speichern
    sortieren
    anzeigen
    stopEvent
  end if
end

-----#####Highscore End#####-----

---Charakter vorwärts durchwählen

on charvor
  if the member of sprite 35 = member "gruen" then
    set the member of sprite 35 = member "rot"
    set the member of sprite 36 = member "r2"
    set gCharakter = 2
    exit
  end if

  if the member of sprite 35 = member "rot" then
    set the member of sprite 35 = member "blau"
    set the member of sprite 36 = member "b2"
    set gCharakter = 3
    exit
  end if

  if the member of sprite 35 = member "blau" then
    set the member of sprite 35 = member "gruen"
    set the member of sprite 36 = member "g2"
    set gCharakter = 1
    exit
  end if
end if

```

```

end charvor
---Charakter rückwärts durchzählen
on charzruck
  if the member of sprite 35 = member "gruen" then
    set the member of sprite 35 = member "blau"
    set the member of sprite 36 = member "b2"
    set gCharakter = 3
    exit
  end if

  if the member of sprite 35 = member "blau" then
    set the member of sprite 35 = member "rot"
    set the member of sprite 36 = member "r2"
    set gCharakter = 2
    exit
  end if

  if the member of sprite 35 = member "rot" then
    set the member of sprite 35 = member "gruen"
    set the member of sprite 36 = member "g2"
    set gCharakter = 1
    exit
  end if
end charzruck

----Waffe auswählen vorwärts
on waffvor
  if the member of sprite 33 = member "waffel" then
    set the member of sprite 33 = member "waffe2"
    set the member of sprite 34 = member "waff2"
    set gWaffe = 2
    exit
  end if

  if the member of sprite 33 = member "waffe2" then
    set the member of sprite 33 = member "waffe3"
    set the member of sprite 34 = member "waff3"
    set gWaffe = 3
    exit
  end if

  if the member of sprite 33 = member "waffe3" then
    set the member of sprite 33 = member "waffel"
    set the member of sprite 34 = member "waff1"
    set gWaffe = 1
    exit
  end if
end waffvor

---Waffe auswählen rückwärts
on waffzruck
  if the member of sprite 33 = member "waffel" then
    set the member of sprite 33 = member "waffe3"
    set the member of sprite 34 = member "waff3"
    set gWaffe = 3
    exit
  end if

  if the member of sprite 33 = member "waffe3" then
    set the member of sprite 33 = member "waffe2"
    set the member of sprite 34 = member "waff2"
    set gWaffe = 2
    exit
  end if

  if the member of sprite 33 = member "waffe2" then
    set the member of sprite 33 = member "waffel"
    set the member of sprite 34 = member "waff1"
    set gWaffe = 1
    exit
  end if
end waffzruck

--Waffe anhören
on waffeanhoeren
  if the member of sprite 33 = member "waffel" then
    puppetSound 3 , "wav1"
    set gWaffeSound = 2
    exit
  end if

```

```

if the member of sprite 33 = member "waffe2" then
  puppetSound 3 , "wav2"
  set gWaffeSound = 3
  exit
end if

if the member of sprite 33 = member "waffe3" then
  puppetSound 3 , "wav3"
  set gWaffe = 1
  exit
end if
end waffeanhoeren

```

---für Hauptmenü -----

```

on soundHS
  puppetsound 3, "computer4"
end

on soundMe
  puppetsound 4, "computer3"
end

on soundSou
  puppetsound 5, "computer5"
end

on cursorAn
  cursor 280
end

on cursorAus
  cursor -1
end

```

Skript 3: Movieskript in menue.dir

6.1.4. all.dir

- Initialisierungsskript:

```

global gLeer , gMunition , gGedruckt , gTreffer , gzeit , gpunkt1 , gliste , gflag , gLeben
global gZitterFlag

on exitFrame
  if the mouseV > 246 or the mouseV < 83 then
    cursor -1
    sprite(112).locH = 320
    sprite(112).locV = 240
  else
    sprite(112).locH = the mouseH ---ZielkreuzH = MausH
    sprite(112).locV = the mouseV
  end if

  put 0 & "." & (59 - the timer/60) into field "zeit" -----Zeitanzeige über 10 Sekunden
  if (59 - the timer/60) < 10 then -----Zeitanzeige unter 10 Sekunden
    put 0 & "." & 0 & (59 - the timer/60) into field "zeit"
  end if

  if gTreffer = 0 then
    gGedruckt = 0
  end if

  if the timer > 3540 then ---Wenn Zeit abgelaufen und
    if gpunkt1 >= 150 then ---Punkte von Level1 über 150 Punkte
      go "gewonnen1" ---dann gewonnen
    else
      go "verloren1"
    end if
  end if
end if

```

```

if gLeer = 1 then
  if keypressed (49) then
    gLeer = 0
    repeat with z = 20 to 29
      add(gMunition , z)
    end repeat
  end if
end if

if gLeben = 0 then
  go "verloren1"
end if

the actorList = (gliste) -- ActorList der Liste gleichsetzen.

go the frame
end

on mouseDown
  gflag = 0 ----gewährleistet, daß man nur 1 Schuß mit einem mousedown abgeben kann
  if gTreffer = 1 then
    gGedruckt = 1
  end if

  if gLeer <> 1 and the mousedown then
    if count(gMunition) > 0 then ----sichtbare Munition
      deleteAt(gMunition , 1)
    end if
    if count(gMunition) = 9 then
      sprite(20).height = 0
    end if
    if count(gMunition) = 8 then
      sprite(21).height = 0
    end if
    if count(gMunition) = 7 then
      sprite(22).height = 0
    end if
    if count(gMunition) = 6 then
      sprite(23).height = 0
    end if
    if count(gMunition) = 5 then
      sprite(24).height = 0
    end if
    if count(gMunition) = 4 then
      sprite(25).height = 0
    end if
    if count(gMunition) = 3 then
      sprite(26).height = 0
    end if
    if count(gMunition) = 2 then
      sprite(27).height = 0
    end if
    if count(gMunition) = 1 then
      sprite(28).height = 0
    end if
    if count(gMunition) = 0 then
      sprite(29).height = 0
      gLeer = 1
    end if
  end if

end if
end

on mouseUp
  gflag = 1 -----richtet wieder die flag auf, damit man weiterschießen kann
  gTreffer = 0
end

on enterFrame
  if gZeit = 0 then
    the timer = 0
    gZeit = 1
  end if

  if gLeer = 0 then
    repeat with m = 20 to 29
      sprite(m).height = 4
      gLeer = 2
    end repeat
  end if
end

end

```

- Moviescript:

```

global gVol, x1, y1, x2, y2 , gHilf , gx1 , gy1 , gx2 , gy2

global gliste, gLeer, gMunition, gGedruckt, gpunkt1, gZeit, gWeg , gLeben , gLevel
global gAnzahl , gWaffe

on prepareMovie
  gWaffe = 2
  case gWaffe of
    1: castlib ("waffel").filename = "waffel"
    2: castlib ("waffel").filename = "waffe2"
    3: castlib ("waffel").filename = "waffe3"
  end case

  initialVol  ---miaw für volume initialisieren
  initialHilf  ---miaw für volume initialisieren

  gMunition = []  ----Munitionsliste initialisieren
  repeat with m = 20 to 29  ---10 Schuß in Liste Munition eintragen
    add (gMunition , m)
  end repeat

  gpunkt1 = 0
  gZeit = 0
  gLeben = 3
  gLeer = 0
  gGedruckt = 0
  gLevel = 0
  put gLeben into field "leben"
  put gpunkt1 into field "punkte"
  put 1 & ":" & 0 & 0 into field "zeit"
  the timer = 0
  cursor 200
end

on StartMovie
  moveToBack window "Volume"
  moveToBack window "Hilfe"
  gliste = [] -- Liste initialisieren
  glevel = 0
end

on Startgame
  glevel = 2
  repeat with i = 38 to 42
    gliste.add (new (script "besatzung" , i))
  end repeat

  anzahl = 101
  addat gliste, (gliste.count+1), new (script "enemy" , anzahl )
end

on neuerfeind
  anzahl = 102
  addat gliste, (gliste.count+1), new (script "enemy" , anzahl )
end

on stopMovie
  the actorList = [] -- ActorListe löschen !!!!!!! WICHTIG !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
  sound Fadeout 1 , 120
  sound FadeOut 2 , 120
end

-----

on initialVol -----MIAW - Volumeregler
  gVol = window "Volume"
  gVol.filename = "@:soundmenue.dir"
  gVol.Windowtype = 2

  x1 = (the stage).rect[3]
  y1 = (the stage).rect[2] + 250
  x2 = (the stage).rect[3] + 200
  y2 = (the stage).rect[4]
  gVol.rect = rect(x1,y1,x2,y2)
  gVol.open()

  set the visible of window "Volume" to false
end initialVol

-----MIAW - Hilfeseite-----
on initialHilf
  gHilf = window "Hilfe"

```

```

gHilf.filename = "@:hilfe.dir"
gHilf.Windowtype = 2

gx1 = (the stage).rect[1] - 200
gy1 = (the stage).rect[2]
gx2 = (the stage).rect[1]
gy2 = (the stage).rect[4]

gHilf.rect = rect (gx1,gy1,gx2,gy2)
gHilf.open()

set the visible of window "Hilfe" to false
end initialHilf
-----

```

Skript 5: Movieskript in all.dir

- Enemy-Script:

```

global gliste , gLeer , gLeben , gpunkt1 , gGedruckt, gTreffer , glebenszeit , gflag2 ,
gCharakter , gWaffe
property pKanal, pKanal2 , pPosX , pPosY , pSpeedX , pSpeedY , pSpeedZ , pLebenszeit , pHoehe
, pWeite
property pFam , pTicks

on new me , kanalnr , Familie
pKanal = kanalnr
pkanal2 = kanalnr + 2
familie = pFam
pFam = 1
sprite(pKanal).memberNum = pFam

set pPosX = random (440) + 50
set pPosY = random (50) + 150
set pWeite = 1
set pHoehe = 1
the locH of sprite pkanal = pPosX
the locV of sprite pkanal = pPosY
the locH of sprite pkanal2 = pPosX
the locV of sprite pkanal2 = pPosY

the width of sprite pkanal = pWeite
the height of sprite pkanal = pHoehe
pSpeedX = random (7) -4
repeat while pSpeedX = 0
pSpeedX = random (7) -4
end repeat

pSpeedY = random (7) - 4
repeat while pSpeedY = 0
pSpeedY = random (7) -4
end repeat
pTicks = the ticks

if gCharakter = void then
gCharakter = 1
end if

case gCharakter of
1: pLebenszeit = (random(120)+ 600) +the ticks
2: pLebenszeit = (random(120)+ 480) +the ticks
3: pLebenszeit = (random(120)+ 360) +the ticks
end case
gLebenszeit = pLebenszeit

return me
end

on fliegen me
if pPosX > 510 or pPosX < 130 then pSpeedX = -pSpeedX
if pPosY > 260 or pPosY < 100 then pSpeedY = -pSpeedY
pPosX = pPosX + pSpeedX
pPosY = pPosY + pSpeedY
the locH of sprite pkanal = pPosX
the locV of sprite pkanal = pPosY
the locH of sprite pkanal2 = pPosX
the locV of sprite pkanal2 = pPosY

set pHoehe = 5
set pWeite = 13

if the width of sprite pkanal < 260 then
set the width of sprite pkanal = the width of sprite pkanal + pWeite
else
set the width of sprite pkanal = 260
end if

```

```

if the height of sprite pKanal < 100 then
  set the height of sprite pKanal = the height of sprite pKanal + pHoehe
else
  set the height of sprite pKanal = 100
end if

end fliegen me

on kill me , i
  repeat with i = 1 to (gliste).count()
    if (gliste)[i].pFam <> 1 then i = gliste.count()

    if intersects(sprite(112), ((gliste) [i].pKanal2)) then----wenn zielkreuz objekt berührt

      sprite(112).member = member "zielkreuzakt"
      gTreffer = 1
      if gLeer <> 1 then
        if gGedruckt = 1 and the mousedown then

          (gliste) [i].pPosY = 1000

          sprite ((gliste)[i].pKanal).locV = pPosY
          sprite ((gliste)[i].pKanal2).locV = pPosY
          (gliste) . deleteAt (i)
          gpunkt1 = gpunkt1 + 20
          put gpunkt1 into field "punkte"

          puppetSound 3 , "waffe"
          gGedruckt = 0

          neuerfeind()
        end if
      end if
    else
      sprite(112).member = member "zielkreuz"
    end if
  end repeat
end

on toeten me
  if the ticks > pLebenszeit then ----wenn die Anzahl der Ticks größer als Lebenszeit
    repeat with i = 1 to (gliste).count() ----alle Feinde überprüfen
      if (gliste)[i].pKanal = pKanal then ----Feind mit überschrittener Lebenszeit finden
        (gliste)[i].pPosX = 1000 ----Feind wegschießen
        (gliste)[i].pPosY = 1000
        sprite ((gliste)[i].pKanal).locH = pPosX
        sprite ((gliste)[i].pKanal).locV = pPosY
        (gliste).deleteAt(i) ----Feind aus Liste löschen
        puppetSound 4 , "bäng"
        if gLeben > 0 then ----Leben -1
          gLeben = gLeben - 1
          neuerFeind() ----neuen Feind erschaffen
        end if
        put gLeben into field "leben"
      end if
    end repeat
  end if

end

on stepframe me
  fliegen
  toeten
  kill
end

```

Skript 6: Enemyskript in all.dir

- Besatzungsskript:

```

global gliste , gflag2
property pKanal , pBesX , pBesY , pDirect , pFlug
property pFam

on new me , kanalnr , Familie
  pKanal = kanalnr
  familie = pFam
  pFam = 6
  sprite(pKanal).memberNum = pFam

```



```

pBesY = random (30) + 360 ----random höhe
sprite(pkanal).locV = pBesY

pDirect = random (2)
pFlug = random (4) + 1

if pDirect = 1 then
  sprite(pkanal).locH = random (50) - 100
  sprite(pkanal).flipH = 1
else
  sprite(pkanal).locH = random (50) + 690
  sprite(pkanal).flipH = 0
end if
return me
end

on renne me
  if pDirect = 1 then
    sprite(pkanal).locH = sprite(pkanal).locH + pFlug
  else
    sprite(pkanal).locH = sprite(pkanal).locH - pFlug
  end if

  if sprite(pKanal).locH > 740 or sprite(pkanal).locH < -100 then
    if pFlug < 0 then
      pFlug = random(4)+1
    else
      pflug = - (random (4) + 1)
    end if
    sprite(pKanal).flipH = not (sprite(pKanal).flipH)
  end if
end

on stepframe me
  renne
end

```

Skript 7: Besatzungsskript in all.dir

6.1.5. erde.dir

- Initialisierungsskript:
Analog all.dir (andere Spritenummern, gPunkte2 etc., aber sonst vom Prinzip her gleich)
- Movieskript: ein Freund kommt noch hinzu

```

on Startgame
  glevel = 2
  -- repeat with anzahl =60 to 61
  repeat with i = 40 to 44
    gListe2.add (new (script "besatzung" , i))
  end repeat

  neuerfreund()
  anzahl = 73
  addat gListe2, (gListe2.count+1), new (script "enemy2" , anzahl )
end

on neuerfreund()
  repeat with i = 66 to 70
    addat gliste2, 1, (new (script "freund" , i ))
    --(gListe2).addAT (new (script "freund" , i ))
  end repeat
end

```

Skript 8: Auszüge aus dem Movieskript in erde.dir

- Enemy2-Skript:

```

global gliste2 , gLeer , gLeben , gpunkte2 , gGedruckt , gTreffer , glebenszeit , gflag2 ,
gCharakter , gWaffe
property pKanal, pKanal2 , pPosX , pPosY , pSpeedX , pSpeedY , pSpeedZ , pLebenszeit , pHoehe
, pWeite
property pFam , pTicks

on new me , kanalnr , Familie

```

```

.....

case gCharakter of
  1: pLebenszeit = (random(120)+ 480) +the ticks
  2: pLebenszeit = (random(120)+ 360) +the ticks
  3: pLebenszeit = (random(120)+ 240) +the ticks
end case
gLebenszeit = pLebenszeit
put gLebenszeit - the ticks
return me
end

on freundberuehren me
  repeat with i = 1 to (gListe2).count()
    if sprite(pKanal).intersects ((gListe2)[i].pKanal) then ----wenn feind anderen berührt
      if (gListe2)[i].pFam <> 1 then ----und es ist kein feind
        puppetSound 5 , "aua"
        gPunkte2 = gPunkte2 - 5
        put gPunkte2 into field "punkte"
        (gListe2)[i].pY = 1000 ----freund abschiessen
        sprite((gListe2)[i].pKanal).locV = (gListe2)[i].pY
        (gListe2).deleteAt(i) ---freund aus Liste löschen
        neuerfreund---Punkte zur Strafe abziehen
        exit
      end if
    end if
  end repeat
end

on stepframe me

  fliegen
  toeten
  kill
  freundberuehren
end

```

Skript 9: Auszüge aus dem Enemy2-Skript in erde.dir

- Freund-Skript:

```

global gListe2 , gLeer , gLeben , gPunkte2 , gGedruckt , gTreffer , gflag2
property pKanal , pX , pY , pZ , pFlugX , pFlugY , pFlugZ , pRichtung , pFam , pkanal2

on new me , kanalnr , Familie
  pKanal = kanalnr
  pkanal2 = kanalnr - 5
  familie = pFam
  pFam = random (3) + 2
  sprite(pKanal).memberNum = pFam
  pRichtung = random (2)

  -- set pX = random (440) + 50
  set pY = random (50) + 150
  set pZ = random (70) ---70 Schritte Unterschied zwischen dem hintersten und vordersten Punkt
  -- the locH of sprite pkanal = pX
  the locV of sprite pKanal = pY
  the locZ of sprite pKanal = pZ
  the locV of sprite pkanal2 = pY
  the locZ of sprite pkanal2 = pZ
  the width of sprite pKanal = integer (pZ / 7 * 9.5)
  the height of sprite pkanal = integer (pZ / 7 * 10.5)
  pFlugX = random (7) -4
  repeat while pFlugX = 0
    pFlugX = random (7) -4
  end repeat

  pFlugY = random (7) - 4
  repeat while pFlugY = 0
    pFlugY = random (7) -4
  end repeat

  pFlugZ = random (7) - 4
  repeat while pFlugZ = 0
    pFlugZ = random (7) -4
  end repeat

  if pRichtung = 1 then
    sprite(pKanal).locH = random (50) - 100
    sprite(pKanal).flipH = 1
  else

```

```

        sprite(pKanal).locH = random (50) + 640
        sprite(pKanal).flipH = 0
    end if
    return me
end

on laufen me
    if pRichtung = 1 then
        sprite(pKanal).locH = sprite(pKanal).locH + pFlugX
        sprite(pKanal2).locH = sprite(pKanal2).locH + pFlugX
    else
        sprite(pKanal).locH = sprite(pKanal).locH - pFlugX
        sprite(pKanal2).locH = sprite(pKanal2).locH - pFlugX
    end if

    if sprite(pKanal).locH > 690 or sprite(pKanal).locH < -100 then
        if pFlugX < 0 then
            pFlugX = random (7) - 4
        else
            pFlugX = -(random (7) - 4)
        end if
        sprite(pKanal).flipH = not (sprite(pKanal).flipH)
    end if

    if pY > 250 or pY < 180 then
        pFlugY = -pFlugY
    end if

    pY = pY + pFlugY

    the locV of sprite pKanal = pY
    the locV of sprite pKanal2 = pY

    if pZ > 70 or pZ < 1 then
        pFlugZ = -pFlugZ
    end if

    pZ = pZ + pFlugZ
    sprite(pKanal).width = integer(pZ/7*9.5)
    sprite(pKanal).height = integer(pZ/7*10.5)
    sprite(pKanal).locZ = pZ
    sprite(pKanal2).locZ = pZ
end laufen me

on stepframe me
    gFlag2 = 1
    laufen
end

```

Skript 10: Freund-Skript in erde.dir

- Besatzungsskript:
Analog all.dir

6.1.6. wasser.dir

- Initialisierungsskript:

```

global gLeer , gMunition , gGedruckt , gTreffer , gzeit , gPunkte3 , gListe3 , gflag , gLeben
, gzitterflag

on exitFrame
    if gzitterFlag = 1 then
        sprite(112).locH = sprite(112).locH + 10
        gzitterFlag = 2
    end if

    if the mouseV > 246 or the mouseV < 83 then
        cursor -1
        sprite(112).locH = 320
        sprite(112).locV = 240
    else
        cursor 200
        if gzitterFlag = void then
            sprite(112).locH = the mouseH ---ZielkreuzH = MausH
            sprite(112).locV = the mouseV
            gzitterFlag = 1
        end if
    end if

```

```

end if

if gZitterFlag = 2 then
  gZitterFlag = void
end if

...

go the frame
end

```

Skript 11: Auszüge aus dem Initialisierungsskript in wasser.dir

- Moviescript:

```

on bewegen M
  sprite(M).loch = sprite(M).loch + 2
end

on abfrage
  if sprite(26).loch > 640 then
    sprite(26).loch = sprite(26).loch - 300
    if sprite(27).loch > 640 then
      sprite(27).loch = sprite(27).loch - 150
    end if
  end if
end

on bewegen2 M
  sprite(M).loch = sprite(M).loch - 2
end

on bewegen3 A
  sprite(A).loch = sprite(A).loch + 2
end

-----

on neuerfreund()
  repeat with i = 67 to 70
    addat gliste3, 1, (new (script "freund" , i ))
  end repeat
end

```

Skript 12: Auszüge aus dem Movieskript in wasser.dir

- Enemy3-Script:

```

global gliste3 , gLeer , gLeben , gpunkte3 , gGedruckt , gTreffer , glebenszeit , gflag2 ,
gCharakter , gWaffe
property pKanal, pKanal2 , pPosX , pPosY , pSpeedX , pSpeedY , pSpeedZ , pLebenszeit , pHoehe
, pWeite
property pFam , pTicks

on new me , kanalnr , Familie
....

case gCharakter of
  1: pLebenszeit = (random(120)+ 360) +the ticks
  2: pLebenszeit = (random(120)+ 240) +the ticks
  3: pLebenszeit = (random(120)+ 120) +the ticks
end case
gLebenszeit = pLebenszeit
put gLebenszeit - the ticks
return me
end

```

Skript 13: Auszüge aus dem Enemy3-Skript in wasser.dir

- Freund-Script:
analog erde.dir
- Besatzung:
analog all.dir und erde.dir

6.1.7. Hilfe / Volume

MIAW: Hier wird der Hilfetext bzw. das Soundmenü in einem MIAW angezeigt.

Muss aus erde.dir, all.dir und wasser.dir initialisiert werden mit folgendem Code im jeweiligen Moviescript:

```
on initialVol -----MIAW - Volumeregler
  gVol = window "Volume"
  gVol.filename = ":@:soundmenue.dir"
  gVol.Windowtype = 2

  x1 = (the stage).rect[3]
  y1 = (the stage).rect[2] + 250
  x2 = (the stage).rect[3] + 200
  y2 = (the stage).rect[4]
  gVol.rect = rect(x1,y1,x2,y2)
  gVol.open()
  set the visible of window "Volume" to false
end initialVol

-----MIAW - Hilfeseite-----
on initialHilf
  gHilf = window "Hilfe"
  gHilf.filename = ":@:hilfe.dir"
  gHilf.Windowtype = 2

  gx1 = (the stage).rect[1] - 200
  gy1 = (the stage).rect[2]
  gx2 = (the stage).rect[1]
  gy2 = (the stage).rect[4]

  gHilf.rect = rect(gx1,gy1,gx2,gy2)
  gHilf.open()
  set the visible of window "Hilfe" to false
end initialHilf
```

Skript 14: Auszüge aus dem Movieskript in all.dir / erde.dir / wasser.dir

6.1.8. Soundmenue.dir

- Moviescript:

```
global gknopf , glaut1 , glaut2 , glaut3

--memberaustausch
on knopf kanal , austausch
  set the member of sprite kanal = member austausch
end

--lautstaerkeregler schief
on startMovie
  gknopf = 0
  festsetzen (13 , 10)
  festsetzen (14 , 11)
  festsetzen (12 , 9)
end

on festsetzen regler , schiene--festsetzen des Reglers auf die Schiene
  set the constraint of sprite regler = sprite schiene
end fest

on bewegung wer--bewegung
  if gknopf = 1 then
    the locV of sprite wer = the mouseV
  end if
end bewegung

on lautstaerke schiene , regler --lautstaerkenregler - unterteilung
  abH = the bottom of sprite schiene - the top of sprite schiene
  reH = the bottom of sprite schiene - the locV of sprite regler
  case regler of
```

```

12: glaut1 = integer (reH/float (abH) *7)
13: glaut2 = integer (reH/float (abH) *256)
14: glaut3 = integer (reH/float (abH) *256)
end case
end

on reglerstellen1 schiene, regler --von rechts unten nach links oben
abH = the bottom of sprite schiene - the top of sprite schiene
the locV of sprite regler = integer(((glaut1*float((abH))/7) - the bottom of sprite schiene)
* -1)

breit = (the right of sprite schiene - the left of sprite schiene)
absHö = (float (the bottom of sprite schiene) - the top of sprite schiene)
faktor = breit / absHö --verhältnis horizontaler zu vertikaler bewegung
startH = the right of sprite schiene --Startpunkt horizontal
startV = the bottom of sprite schiene--startpunkt vertikal
the locH of sprite regler = startH + float ((faktor * (startV - the locV of sprite regler))*
-1) --umkehrung der bewegung durch vorzeichenwechsel
end

on reglerstellen2 schiene, regler ---von links unten nach rechts oben

abH = the bottom of sprite schiene - the top of sprite schiene
the locV of sprite regler = integer(((glaut2*float((abH))/256) - the bottom of sprite
schiene) * -1)

breite = (the right of sprite schiene - the left of sprite schiene)
absH = (float (the bottom of sprite schiene) - the top of sprite schiene)
faktor = breite / absH --verhältnis horizontaler zu vertikaler bewegung
startH = the left of sprite schiene --Startpunkt horizontal
startV = the bottom of sprite schiene--startpunkt vertikal
the locH of sprite regler = startH + float (faktor * (startV - the locV of sprite regler))
end

on schieben schiene , regler --schräge Bewegung von links unten nach rechts oben
breite = (the right of sprite schiene - the left of sprite schiene)
absH = (float (the bottom of sprite schiene) - the top of sprite schiene)
faktor = breite / absH --verhältnis horizontaler zu vertikaler bewegung
startH = the left of sprite schiene --Startpunkt horizontal
startV = the bottom of sprite schiene--startpunkt vertikal
the locH of sprite regler = startH + float (faktor * (startV - the locV of sprite regler))
end schieben

on schieb schiene , regler --schräge Bewegung von rechts unten nach links oben
breit = (the right of sprite schiene - the left of sprite schiene)
absHö = (float (the bottom of sprite schiene) - the top of sprite schiene)
faktor = breit / absHö --verhältnis horizontaler zu vertikaler bewegung
startH = the right of sprite schiene --Startpunkt horizontal
startV = the bottom of sprite schiene--startpunkt vertikal
the locH of sprite regler = startH + float ((faktor * (startV - the locV of sprite regler))*
-1) --umkehrung der bewegung durch vorzeichenwechsel
end schieben

```

Skript 15: Movieskript in soundmenu.dir

6.2. Externe Casts (Besetzungen)

Externe Besetzungen sind nützlich zum Speichern von Elementen, die in verschiedenen Filmen verwendet werden, und zur Erstellung von Bibliotheken von häufig verwendeten Darstellern, wie z. B. Schaltflächen und Verhalten. Sie können externe Besetzungen auch zum Austausch ganzer Mediagruppen während der Laufzeit verwenden, ähnlich wie beim Wechsel zwischen Sprachen. Externe Besetzungen sind häufig sinnvoll, wenn die Größe der Filmdatei für das Herunterladen möglichst gering sein soll, da eine externe Besetzungsdatei nur bei Bedarf separat von der Filmdatei geladen werden kann.

Man kann eine externe Besetzung mit dem aktuellen Film verknüpfen:

Dazu öffnet man eine externe Besetzung und zieht dann einen Darsteller aus der externen Besetzung auf die Bühne oder ins Drehbuch. In einem Dialogfeld kann man auswählen, ob man die Besetzung mit dem Film verknüpfen oder den Darsteller in die interne Besetzung kopieren möchten.

Man kann aber auch eine externe Besetzung mit einem Film verknüpfen, ohne einen Darsteller hinzuzufügen:

- Auswählen von "Modifizieren" > "Film" > "Besetzungen".
- Anklicken im Dialogfeld "Filmbesetzungen" auf "Verknüpfen".
- gewünschten externen Besetzung aussuchen und auswählen, dann klickt man auf "Öffnen".

Man kann Verknüpfungen mit Besetzungen auf dem lokalen Datenträger oder mit Besetzungen, die an einer beliebigen URL gespeichert sind, herstellen.

Hier in Surroundings werden vier externe Besetzungen verwendet: waffe 1.cst, waffe 2.cst, waffe3.cst und hilfvol.cst

Auszüge aus dem Inhalt der externen Besetzung hilfvol.cst:

Skript hilf wandern:

```
global gx1 , gInihilf
on mouseWithin
  if ginihilf = 1 then
    initialhilf
    set ginihilf = 0
  end if

  set the visible of window "Hilfe" to true
end
```

Skript 16: Auszüge aus dem Skript hilf wandern in hilfvol.cst

Ansonsten beinhaltet es den Button und die Beschriftungen und diese Funktionalitäten „wandern“, um das MIAW sichtbar zu machen.

Waffe1-3.cst beinhaltet nur den jeweiligen Sound der Waffen 1-3, die ja im Spiel verwendet werden können.

7. Projektor erstellen

Zum Erstellen von Projektoren für Windows muss man mit der Windows-Version von Director arbeiten. Genauso können Macintosh-Projektoren nur mit der Macintosh-Version von Director erstellt werden.

Bestimmte Xtras müssen in einem Projektor eingebettet sein, damit dieser Funktionen wie Text, Flash-Filme, Internet-Verbindung usw. ausführen kann. Die am häufigsten verwendeten Xtras werden standardmäßig eingeschlossen. Sie können Xtras einzeln für jeden Film integrieren bzw. entfernen, und zwar mit der Option "In Projektor einschließen" im Dialogfeld "Film-Xtras". Sie können Xtras auch manuell einem Projektor hinzufügen, auf die gleiche Weise, wie Sie Filmdateien auswählen. Siehe Xtras für vertriebene Filme verwalten.

Man muss auch darauf achten, dass der Anfangsfilm in der Dateiliste im Dialogfeld "Projektor" an oberster Stelle steht.

Da der Film Lingo enthält, das zwischen Filmen hin- und herschaltet, wird nur vom allerersten Film ein Projektor erstellt. Die anderen Filme und Besetzungen und 3D-Filme liegen in den Ordnern „data/movies“ bzw. „data/qtmovies“, die mit auf die CD gebrannt werden müssen.

Vorteil von dieser Vorgehensweise ist, dass der Projektor selbst nicht so riesig ist und nicht komplett am Anfang geladen werden muss, sondern nur Stück für Stück, wenn der jeweilige Film aufgerufen wird.

Director behält die definierten Optionseinstellungen bei, so dass sie nicht jedes Mal neu eingestellt werden müssen.

Für diesen Projektor habe ich folgende Optionseinstellungen verwendet:

- Alle Filme abspielen
- Vollbild
- Filmeinstellungen verwenden
- Zentrieren
- Standardplayer
- Keine Komprimierung

Um Probleme mit verknüpften Media zu vermeiden, sollte der neue Projektor am besten in seinem endgültigen Zielordner erstellt und anschließend nicht mehr in einen anderen Ordner verlagert werden.

Director konvertiert alle Filme, Besetzungen und eingeschlossenen Xtras in einen einzigen Projektor.

8. Probleme

8.1. Sonderzeichen

Sonderzeichen wie „ä“ in „Lautstärke“ führte am Ende manchmal zu Problemen wie „Handler not defined“ .

Umbenennen in „lautstaerke“ löst das Problem sehr schnell.

8.2. Hintergrund

MIAW (Movie in a Window) lässt sich durch nichts in den Hintergrund rücken. Wenn der Film im Projektor abläuft, wird das MIAW vor der Bühne angezeigt.

Abhilfe: MIAW ausblenden und bei Mouseover einfach einblenden lassen.

8.3. Plattformunabhängige Pfadangaben

Pfadangaben plattformunabhängig mit @ machen, ansonsten kann es sein, dass auch auf der eigenen Plattform die Pfade nicht erkannt werden.

8.4. Schräger Lautstärkereglер

Schräger lautstärkereglер

- Constraint beschreibt Rechteck
- Schieberegler somit im Rechteck verschiebbar

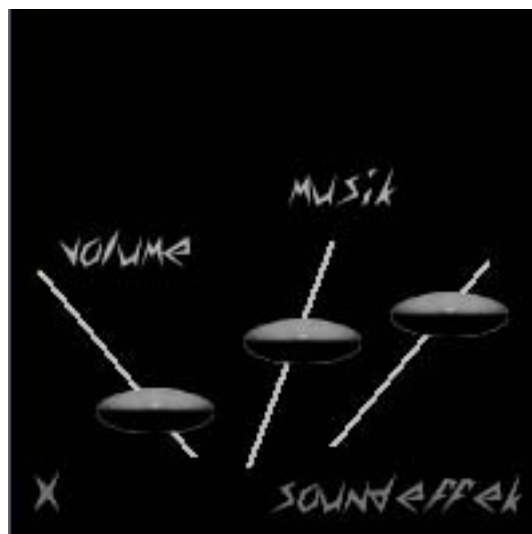


Abbildung 6: Screenshot von Soundmenue.dir

Bei senkrecht oder waagrechten Reglern ist die Lösung sehr einfach: Höhe oder Breite : 1 Pixel

Wie bekommt man die Schräge hin?

- Mathematische Formel finden: über die Diagonale findet man den rechten unteren bzw. linken unteren Punkt
- Linke obere Ecke des Sprites von Director über Attribute rechts unten errechenbar

```
on schieb schiene , regler --schräge Bewegung von rechts unten nach links oben
  breit = (the right of sprite schiene - the left of sprite schiene)
  absHö = (float (the bottom of sprite schiene) - the top of sprite schiene)
  faktor = breit / absHö --verhältnis horizontaler zu vertikaler bewegung
  startH = the right of sprite schiene --Startpunkt horizontal
  startV = the bottom of sprite schiene--startpunkt vertikal
  the locH of sprite regler = startH + float ((faktor * (startV - the locV of sprite regler))*
-1) --umkehrung der bewegung durch vorzeichenwechsel
end schieben
```

Skript 17: Auszüge aus dem Movieskript in soundmenue.dir

8.5. Quit

Quit beendet den Projektor nicht. Auch exit beendet den Projektor nicht. Dazu habe ich leider keine Lösung gefunden, da er nach dem Befehl immer zum aufrufenden Film zurückkehrt.

8.6. Objekte in OOP

Das erste erzeugte Objekt muss getötet werden, ansonsten kann kein normaler Spieleablauf gewährleistet werden. Woran das liegt ist bis heute unklar.

8.7. Highscore Xtra

Um die Punkte dauerhaft speichern zu können, muss man mit einem Xtra arbeiten, das es möglich macht, aus dem Directorspiel heraus auf der Festplatte eine Textdatei mit den Daten zu speichern.

Das wurde aber nicht implementiert aus folgenden Gründen:

Das Xtra FileIO, das man dafür benötigt, kostet extra Geld und stand mir leider nicht zur Verfügung. Des Weiteren muss man bedenken, dass auch Sicherheitsaspekten das Speichern von Daten vom Benutzer auf dessen Festplatte nicht erwünscht sein könnte.

Der Ausweg aus diesem Dilemma wäre eine Verbindung heraus aus dem Spiel ins Internet, um dort über eine kleine Datenbank die entsprechenden Daten abzuspeichern. Das könnte durchaus mit php realisiert werden.

Aus Zeitgründen konnte ich dies aber nicht realisieren und somit existiert nun nur die kleine Lösung, dass man eben pro laufende Session (d.h. solange der Projektor läuft) die Daten gespeichert werden kann. Damit hat man dann eine kleine Übersicht über den Spielstand, wenn man mehrmals hintereinander spielt.