

Die Erstellung von Shildimon Levels

1. Erstellung einer neuen Datei

Man gehe in das gewünschte Verzeichnis und erstelle dort eine reine Textdatei (Endung: .txt).

Dann ändere man die Dateiendung in **.shl** .

Dies kann bei Windowssystemen teilweise Probleme bereiten, wenn Dateiendungen ausblenden eingestellt ist, da dann die Endung nicht geändert wird, sondern nur der Dateiname durch .shl ergänzt. → Man erhält **fälschlicherweise name.shl.txt**
Name der Datei ist zwar nicht gleichzeitig der im Spiel angezeigte Name der Landschaft, sollte aber trotzdem sinnvoll gewählt sein, da er in der weiteren Gestaltung von Levels eine wichtige Rolle spielt.

2. Der Grundinhalt der Datei

2.1 Der Dateikopf

Der Dateikopf der Datei enthält wichtige Daten darüber, wie der Computer mit der Datei umgehen soll:

[Eigenschaften] size = (x,y) name = Landschaftsname music = bgmusic.mp3
--

Zuerst kopiere man diesen Kopf in seine neue Landschaftsname.shl Datei.

Dann ändere man x und y in die gewünschte Breite und Höhe der Landschaft.

In unserem Beispielfall in **size = (1,1)** , da wir erst einmal der Übersicht halber nur ein Feld erstellen.

Dann ändere man den Landschaftsnamen zB. in **name = Meine erste Landschaft.**

Und zuletzt füge man den Namen der Hintergrundmusik ein. Es ist dabei wichtig, dass diese in dem Spielordner in data/Musik liegt.

Eine Angabe von Unterordnern in diesem Ordner ist möglich, indem man zum Beispiel music = **Unterordner/musikname.zb** angibt.

Wichtig: Für das abtrennen von Unterordnern mus ein / verwendet werden.

Für unser Beispiel nehmen wir die Standartmusik

name = by_joram/sstandartm.mid.

Damit ist der Kopfteil unserer Datei schon einmal fertig.

2.2 Ein erstes Feld

Als zweites müssen wir natürlich das eine Feld unserer 1x1 Landschaft definieren. Dazu schreiben wir folgenden Text in unsere Datei:

[x,y] action=. background=boden.png file=gras.png
--

In diesen Angaben steht alles, was der Computer über das Feld wissen muss.

background definiert zB. die Unterste Bildebene des Feldes. Darüber wird dann die zweite Bildebene angezeigt. Diese besitzt meist durchsichtige Stellen und man kann so zB. einen Baum obwohl man nur ein Bild von einem Baum hat sowohl auf Gras, als auch auf einem einfache braunen Boden, oder sogar im Wasser anzeigen lassen. Alle Bilder die man bei diesen beiden Stellen benutzen kann sind unter data/Objekte/ Landschaftssegmente/ und den entsprechenden Unterordnern zu finden. Es ist dabei zu beachten, dass die in Unterordnern enthaltenen Bilder innerhalb dieses Ordners auch nur mit einfachem Namen und nicht mit Pfad angesprochen werden und daher doppelte Dateinamen nicht vergeben werden dürfen. (Dies gilt insbesondere dann, wenn man selber neue Bilder hinzufügt.)

Die Beschreibung **action** eines Feldes gibt an, was passiert, wenn der Spieler versucht das Feld zu betreten, d.h. wenn ein Spieler vor dem Feld steht und die entsprechende Pfeiltaste betätigt. Die möglichen Befehle für Felder werden im nächsten Abschnitt erläutert. Es ist zu beachten, dass der Spieler das Feld wirklich nur dann Betritt, wenn der Befehl das so angibt, in jedem anderen Fall bleibt der Spieler davor stehen.

Die Werte x und y in der Einleitung zum jeweiligen Feld geben an, wo das Feld angezeigt werden soll. Dabei wird von der oberen linken Ecke aus gezählt. Wichtig ist, dass dabei beim Zählen mit Null angefangen wird und nicht mit Eins.

In unserem 1x1 Landschaft Beispiel müsste die Zeile also [0,0] lauten.

2.3 Aktionsbefehle

Anmerkung:

! keine Leerzeichen in Befehlen verwenden (außer in text() ab Sh1.8)

! geschachtelte Befehlstrukturen funktionieren nicht

! Die vordefinierten Variablen nicht mit varset ändern

Befehl	Was passiert
x	dieses Feld ist gesperrt: Aufprallgeräusch (+ Spieler läuft nicht)
.	dieses Feld ist begehbar: Spieler geht ein Feld in Blickrichtung
xtbrl	teilweise gesperrtes Feld: t -> von oben nicht betretbar, b -> von unten nicht betretbar, l -> von links nicht betretbar, r -> von rechts nicht betretbar; <i>(es müssen nicht alle dieser Buchstaben hinter dem x stehen: zB. xtb)</i>
(befehl,befehl,...)	führt mehrere Befehle der Reihe nach durch
e / eis / ice	Feld vereist: Spieler geht ein Feld und versucht nächstes zu betreten
v	Spieler wird auf das unter diesem Feld befindliche Feld bewegt
^	Spieler wird auf das über diesem Feld befindliche Feld bewegt
<	Spieler wird auf das links neben diesem Feld befindliche Feld bewegt
>	Spieler wird auf das rechts neben diesem Feld befindliche Feld bewegt
drehelinks	Spieler wird nach links gedreht (Westen)
dreherechts	Spieler wird nach rechts gedreht (Osten)
dreheoben	Spieler wird nach oben gedreht (Norden)
dreheunten	Spieler wird nach unten gedreht (Süden)
links	Wie wenn der Spieler die linke Pfeiltaste drückt
rechts	Wie wenn der Spieler die rechte Pfeiltaste drückt
oben	Wie wenn der Spieler die obere Pfeiltaste drückt
unten	Wie wenn der Spieler die untere Pfeiltaste drückt
open(landschaft,(x,y))	Öffnet die angegebene Landschaft am Punkt (x,y) (nach Koordinatensystem mit (0,0) = Landschaftsmitte)

goto(x,y) / goto((x,y))	bewegt den Spieler in der aktuellen Landschaft an den Punkt (x,y)
var(varname)	wird vor dem Ausführen anderer Befehle durch den Wert der Variable varname ersetzt
varset(varname=wert)	setzt den Wert der Variable varname auf wert
wenn(wertovvariable1=wertodervariable2, dann,sonst)	vergleicht Werte und/oder Variablen und führt entweder dann oder sonst aus (zB.: wenn(var(name)=odie,text(Das ist aber ein komischer Name),nichts)
nichts	macht nichts (man kann die Stelle auch einfach freilassen)
text(text)	zeigt text an (bis Version1.8 <i>muss</i> ~ statt Leerzeichen verwendet werden)
timer(Sec,Befehl)	führt Befehl aus, wenn man die entsprechende Anzahl von Sekunden (Kommazahlen mit Punkt statt Komma) stehen bleibt
sound(soundfile)	Spielt einen Sound aus dem Ordner data/Sounds ab
getitem(item, [dann,sonst])	Spieler bekommt das Item (wenn es klappt führe dann aus, wenn nicht, dann sonst)
removeitem(item, [dann,sonst])	Spieler verliert das Item (wenn es klappt führe dann aus, wenn nicht, dann sonst)
Befehle für Objekte (Objekte siehe 3.1 Objekte)	
objekt.shape(img.png)	Das Objekt objekt nimmt die Form img.png an (s. Objektformen)
objekt.move(x,y)	Bewege das entsprechende Objekt um (x,y)

3. Erweiterte Möglichkeiten der Gestaltung

3.1 Objekte

3.2 Forperson