




Dies ist eine Gebrauchsanleitung für den Visual Pinball Editor, erstellt von Clold1
 Übersetzung von Gobi mit Ergänzungen und Überarbeitung für VP 0.62 von Steampump*

Visual Pinball Visual Guide /Deutsch

Editor	- Grundlagen des Visual Pinball Editors. Menüleisten Funktionen - importieren von Images & Sounds
*Table Optionen - BallFarbe	- Erklärt die Table Optionen, importieren eines Ball Images & verändern der Ballfarbe
*Optionen der Rechten Taste	- Verwendung der rechten Maustaste mit den Objekten (neu von Steampump)
*Erzeugen eines Sling Shot	- Zeigt wie man schnell ein Sling Shot macht
*Decals	- Erklärt die Decal Optionen, & zeigt wie Bilder dafür importiert werden. Pinball DemoDecal
*Plungers & Kickers	- Wie man Plunger & Kicker verwendet & ein Beispiel
*Walls & Targets	- Mauern und Zielscheiben bauen (neu von Steampump)
*Lights & Bumpers	- Alles über Light & Bumper States & Lights- & Bumpers- Optionen Leiste. Pinball Demo2
*Collection manager	- Wie Objektgruppen verwaltet werden (neu von Steampump)
*Gate & Sound Ereignisse	- Erklärt die Gate Optionen & wie man Sounds über das Script startet. Pinball Demo2
*TextBox & Making Score count	- Zeigt wie man TextBoxen verwendet & AddScore & Dim Score im Script. Pinball Demo3
*Ramps	- Zeigt die Ramp Optionen & und den Bau einer einfachen Rampe. Pinball Demo4
*Spinner	- Erläut Spinner Optionen & Script für einen Spinner. Pinball Demo4
*Triggers	- Erklärt Trigger Optionen & Script für einen Trigger. Pinball Demo5
*Timers	- Erklärt Timer Optionen & Script für einen Timer. Pinball Demo6
*Flippers	- Zeigt die Flipper Optionen & das entsprechende Script. Pinball Demo7
*EmReels	- Wie man EmReels Zähler verwendet (neu von Steampump)
*Kontrolpunkte und Seiten Bilder:	- Wie man Seitenbilder kontrolliert auf Wände klebt. (neu von Steampump)

 [English Index](#) - Back to English Index

 [Französisches Index](#)

Der Visual PinBall Editor

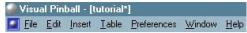
Der **Editor** sieht wahrscheinlich anfangs etwas einschüchternd aus, aber wir werden Stück für Stück die einzelnen Grundelemente durchgehen. Der Editor ist das grundlegende Teil um einen Tisch zu bauen. Hier werden die Wände, Lichter, Bumper, Targets und alle anderen Bestandteile des Flippers gesetzt.

Im Bild rechts ist **Options** ausgewählt.

Wenn **Options** angeklickt ist, können die Eigenschaften des ausgewählten Objektes verändert werden. Zu sehen ganz rechts im Bild, **Table** ist ausgewählt.



Zuerst betrachten wir die Menüleiste, dort wird geladen/gespeichert und vieles mehr.



Ganz oben (dunkelblau) steht der Name des Tisches, der in den VP Editor geladen ist. Zu erst sehen wir uns die File Optionen an

Commande **File**

File	Edit	Insert	Table	Preferences	Window	Help
New						Ctrl+N
Open...						Ctrl+O
Close						
Save						Ctrl+S
Save As...						
Save As Protected...						
Export						
Unlock						

File/
New - Erzeugt einen neuen Tisch (enthält nur Grundelemente)
Open - Einen Tisch zum öffnen auswählen
Close - Den geöffneten Tisch schließen
Save - Speichert den geöffneten Tisch
Save As... - Speichert den geöffneten Tisch z.B. unter anderem Namen
Save As Protected - Schützt den Tisch mit einem Passwort
Export - Exportiert eine Blaupause des Tisches als Bearbeitungsgrundlage in einem Malprogramm
Unlock - Hebt die Passwortschutz auf
Exit - Visual Pinball verlassen

Save As Protected. Options

Table Protection Options (At least one must be ticked)

- Disable Save, Save As..., Save As Protected... and Export Blueprint
- Disable only Save As Protected.
- Disable Script Editor from being opened
- Disable File Managers (Image, Sound, Font and Collection) from being opened
- Disable copying of table elements into the clipboard
- Disable viewing of table elements
- Disable everything, Load and Play only. (This also includes any future locks)

Please enter your Password: [] (Maximum of 16 characters)
 Please reenter your Password: []

Show Password

Buttons: Save Table As..., Cancel

Unlock Protected Table

Please enter Password: []

Buttons: Unlock Table, Cancel

Unlock Das Dialogfenster erlaubt ihnen ein Passwort einzugeben um den Tisch frei zu geben.

Commande **Edit**

Edit	Insert	Table	Preferences
Undo			Ctrl+Z
Copy			Ctrl+C
Paste			Ctrl+V
Delete			Del
Options			
Script			
Backglass View			

Edit/
Undo - Macht den letzten Arbeitsschritt rückgängig. (Strg+Z erfüllt den gleichen Zweck)
Copy - Kopiert ein ausgewähltes Teil. (Strg+C)
Paste - Fügt das kopierte Teil ein, und zwar genau über dem Original (Strg+V)
Delete - Löscht das ausgewählte Teil, oder eine Ausgewählte Gruppe von Teilen (Entf)
Options - Öffnet die Options, es werden jeweils die Optionen des angeklickten Teiles angezeigt.

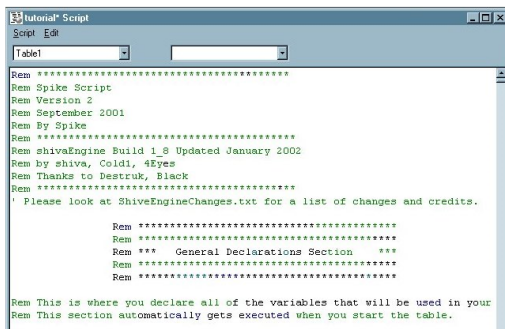
Options Options entsprechende Schaltfläche im **Toolbar** (Werkzeugleiste)

Script - Öffnet den Script-Editor. Hier werden die Steuerbefehle hineingeschrieben. Geräusch an, Licht aus usw.

Script Sriptentsprechende Schaltfläche im **Toolbar**

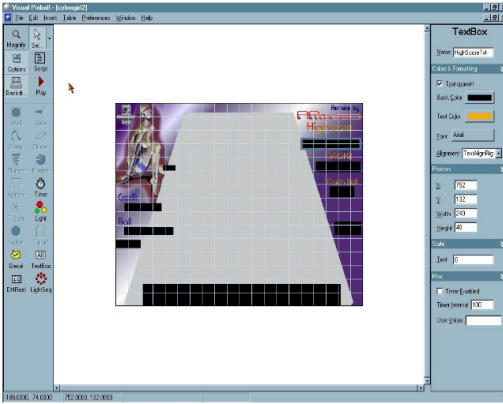
Backglass View - zeigt die Backglass Ansicht, dort sieht man wo sich z.B. die Score Anzeige befindet und eine Übersicht in welcher Perspektive der Tisch dargestellt wird. Backglass Entsprechende Schaltfläche im Toolbar bezeichnet als **Backdrop**

Scriptfenster



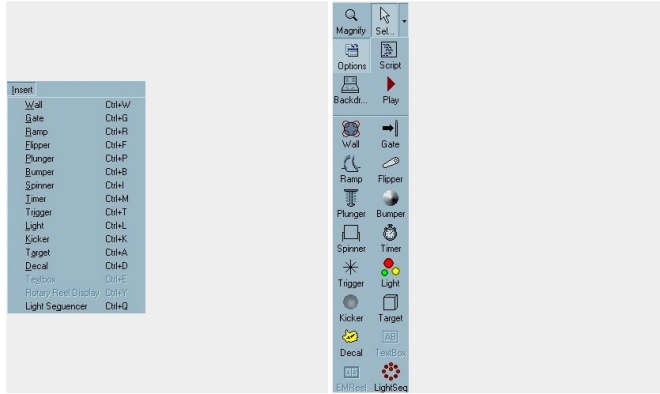
```
Force declaration of variables to prevent errors and typos
Option Explicit
```

Rückwand und Anzeigetafel (Backglass o Backdrop)



Die **Insert Options**

Das Selbe als Schaltfläche im **ToolBar**



Das sind die Objekte die in einen Flipper eingefügt werden können. Sie schauen wir uns im nächsten Teil genauer an, besonders mit ihren Options Eigenschaften.

Comande **Table**

Table/

Play—Wenn ein Tisch in den Editor geladen ist, kann er hier gespielt werden.

Playpossède un bouton sur la **barre d'outil** et un raccourcis **F5**.

Sound Manager—Hiermit importiert man "Sounds" in den Tisch, sie werden später im script durch den Befehl *PlaySound* "**BeispielSound**" aufgerufen.

Image Manager—Hiermit importiert man Bilder in den Tisch. Sie können als Oberfläche für diverse Objekte benutzt werden.

Font Manager - **nicht verwendet**

Collection Manager - Eine Gruppe von Bestandteilen auswählen um eine "collection" anzulegen

Magnify - Zoomt in einen ausgewählten Bereich des Tisches im Editor (Strg+ linke Maustaste zum hineinzoomen, bzw. rechte zum herauszoomen).

Magnify - Entsprechende Schaltfläche im Toolbar, wenn ausgewählt herauszoomen durch rechtsklicken

Der Zoom kann sehr langsam werde wenn der Tisch viele Objekte hat und die Anzeige der Bilder auf Oberflächen aktiv ist.

Comande **Sound Manager**

Import - Öffnet ein neues Fenster zum importieren von .wav Dateien

Der Sound wird beim Speichern des Tisches mitgespeichert, also Bestandteil des Flippers. Wiedergegeben wird er im Script z.B. durch *PlaySound* "Bumper"

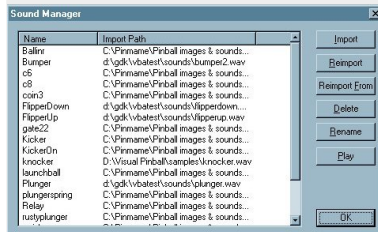
Reimport - Importiert den ausgewählten Sound nochmals von der gleichen Quelle.

Reimport From - Importiert den ausgewählten Sound nochmals von einer anderen Quelle.

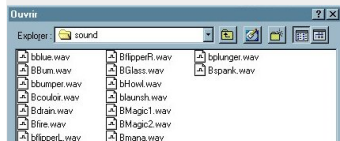
Delete - Löscht den ausgewählten Sound aus dem Tisch.

Rename - Ausgewählten Sound umbenennen.

Play - Ausgewählten Sound abspielen.



Nach der Auswahl von **Import**, öffnet sich dieses Fenster (unten), hier wird die Herkunft der Dateien angezeigt. Nach dem öffnen, werden sie im Sound Manager angezeigt.



Achtung: zur Zeit können nur .wav Dateien für Sounds im Spiel benutzt werden. Mp3's werden als "Music" durch das Script gestartet

Siehe auch [Faq](#) für mehr Informationen zum Thema "Music"

Nom :
 Type :
 Lecture seule

Commande Image Manager

Import- Ouvre un autre fenêtre pour importer des images. Les images sont sauvegardées dans le dossier de travail du Flipper.

Reimport- Réimporte les images sélectionnées de la même source.

Reimport From- Réimporte les images sélectionnées d'une autre source.

Delete- Supprime les images sélectionnées.

Rename- Renomme les images sélectionnées.

Achtung - À l'heure actuelle, seule la utilisation de .bmp et .jpg images est possible. Plus d'informations sont dans le [FAQ](#).

Transparency - Choisit une couleur qui sera transparente. Dans l'exemple, le blanc est sélectionné, ce qui signifie que les pixels blancs dans l'image seront transparents. Utilisez noir ou blanc RGB (255,255,255) pour la masque transparent, sinon cela ne fonctionnera pas.

Option Preferences

Preferences/
Einstellungen

Audio - Hier kann die Sound/Music Lautstärke geregelt werden. Voreinstellung ist dargestellt.

Editor Option

Permet de modifier certaines propriétés de l'éditeur.

Always Draw Control Points Affiche tous les points de contrôle même s'ils ne sont pas sélectionnés.

Draw Light Center Cross Affiche le centre de la lumière même si elle n'est pas sélectionnée.

Keys - Hier kann die Tastaturbelegung zugewiesen werden, mit der der Flipper gespielt wird.

Achtung - Alle Keycodes sind schon im Script eines neu angelegten Tisches enthalten, mit Ausnahme von "add credit" & "start game".

Z.B. Die Plunger Taste ist der Enter Taste in den keys **Preferences** zugewiesen:

Im Script:

```
If keycode = PlungerKey Then  
Plunger.PullBack  
End If
```

Wird eine andere Taste zugewiesen, muss entsprechend das Script geändert werden.

Für "add a coin" & "start game", müssten folgende Zeilen in das Script eingefügt werden:

```
If keycode = AddCreditKey Then 'assigned Key is 5
```

```
If keycode = StartGameKey Then 'assigned Key is 1
```

Sie können die Codes im Script eingeben. Wie das funktionieren können Sie aus dem Script [Shivaengine.vps](#) mit einem Texteditor entnehmen. Es ist das Script der Shivaengine.

Info: *PlungerKey AddCreditKey StartGameKey* sind Konstanten mit numerischen Werten. Man kann auch direkt den Wert eingeben, sofern man ihn kennt.

```
If keycode = 6 Then 'Keyboard taste 5 wird verwendet
```

[Liste des Keycodes im Textformat.txt](#)

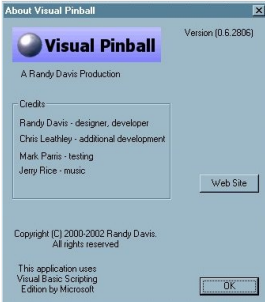
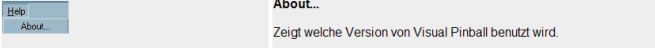
Video - Hier kann die Auflösung des Spiels eingestellt werden. Voreinstellung ist dargestellt.



Die **W**indow Optionen

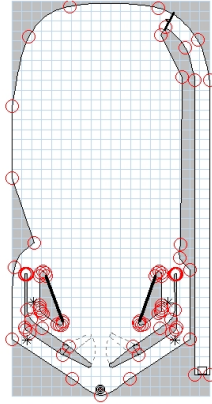


Die **H**elp Optionen



Editor Table Optionen

Das ist der Tisch den man durch **File/New** erhält.


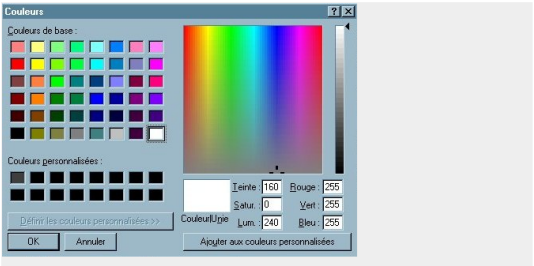


Auf den Tisch klicken und....



Options anwählen, um **Table Options** zu betrachten

Options Table	nach Anklicken des Tisches & und auswählen von Options in der Werkzeugeiste.				
<p>Table</p> <p>Name [Table]</p> <p>Editor</p> <p><input checked="" type="checkbox"/> Show Grid</p> <p>Grid Size [50]</p> <p><input checked="" type="checkbox"/> Display Image in Editor</p> <p>Colors & Formatting</p> <p>Color []</p> <p>Image []</p> <p><input checked="" type="checkbox"/> Render Shadows</p> <p>Physics</p> <p>Width [1000]</p> <p>Height [2000]</p> <p>Glass Height [210]</p> <p>Table Slope [6]</p> <p>Ball</p> <p>Ball Image []</p> <p>Front Decal []</p> <p>Back Decal []</p>	<p>Table - Die Art des ausgewählten Objektes.</p> <p>Name - <i>Table1</i> Der Name des Tisches, kann geändert werden, bringt aber eigentlich nichts.....</p> <p>----- Editor</p> <p>Show Grid - Voreingestellt ist dieses Gitternetz sichtbar. Die Horizontalen und Vertikalen Linien sind zur Orientierung nützlich, um Teile in eine Flucht zu stellen oder eine gerade Linie zu ziehen.</p> <p>Grid Size - Voreingestellt auf 50, das Gitter kann vergrößert oder verkleinert werden, durch die Eingabe größerer oder kleinerer Zahlen.</p> <p>Display Image in Editor - Wenn der Tisch nach einem Bild aufgebaut wird, muß man in der Lage sein das Bild schon im Editor zu sehen. Achtung: Der Editor wird dadurch, vor allem wenn "Display Image in Editor" auch noch bei anderen Walls angeschaltet ist, wahrscheinlich langsam! Es sollte also nur angeschaltet sein, bei dem Teil an dem man gerade arbeitet.</p> <p>----- Colors & Formatting</p> <p>Color - Farbe des Tisch ändern.</p> <p>Image - Hier das Bild des Tisches auswählen (es muß natürlich vorher mit dem Image Manager importiert worden sein) Hinweis: Table Images sind normalerweise 512 x 1024 Pixels groß (oder als Faustregel: halb so groß wie Width & Height) Notiz: Maximale Größe 1024x1024 pixel</p> <p>Render Shadow Schatte des Tisch anzeigen.Voreinstellung aktiv</p> <p>-----Physics</p> <p>Width - Ändert die Breite des Tisches. 1000 ist voreingestellt.</p> <p>Height - Ändert die Höhe des Tisches. 2000 ist voreingestellt.</p> <p>Glass Height - Die Höhe des Glases, entsprechend der Glasplatte eines echten Flippers. Achtung: Wenn Objekte, besonders Walls, Ramps, ect. zu hoch im Verhältnis der Glass Height sind, kann der Ball stecken bleiben! 210 ist voreingestellt. Hinweis: Der Ball in VP ist 50 Einheiten, also sollte die Maximalhöhe der Objekte Glass Height - 50 sein.</p> <p>Table Slope - Stellt den Neigungswinkel des Tisches ein. Voreingestellt ist 6.</p> <p>-----Ball</p> <p>Ball Image - Hier kann ein eigenes Bild für den Ball eingestellt werden. Das Bild muss im Image Manager importiert sein.</p> <p>Note: Welches Bild gezeigt wird, kann auch über das Script geregelt werden z.B. <i>Plunger.CreateBall.Image = "ballimage8"</i> (Kicker können übrigens auf die gleiche Weise einen Ball erzeugen.)</p> <p>Front Decal Erlaubte es ein Bild auf die Vorderseite des Ball zu kleben.Das Bild muss im Image Manager importiert sein. Befehl im Script für den Plunger:<i>Plunger.FrontDecal = "image1"</i></p> <p>Back Decal Erlaubte es ein Bild auf die Rückseite des Ball zu kleben.Das Bild muss im Image Manager importiert sein. Befehl im Script für den Plunger:<i>Plunger.BackDecal = "image1"</i></p> <p>Notiz: Im Editor kann man alle 3 Option gleichzeitig einstellen. (Ball Image Front Decal Back Decal). Im Script nur ein aus den 3 Optionen (Ball Image Front Decal Back Decal)</p> <p>Notiz: Das Bild für den Ball ist keine Textur im Sinne eines 3D Programm. Es ist nur ein Sprit (2 Bild) das man auf den Tisch bewegt. Die Decal auf dem Ball bewegen sich mit ihm. Leider sind die Bilde sehr klein und entsprechen nur 1/4 der Sichtbaren Oberfläche des Ball.</p>				
<p> 64x64 Pixels</p> <p>Tip 64x64 Pixel ist die Standardgröße für einen Ball. Das Bild muss im .bmp .jpg Format sein, weiß wäre im Beispiel auf transparent zu setzen, wie unten gezeigt.</p> <p>ballimage8 Download</p>					
<p>Table Preferences Window</p> <p>Play F5</p> <p>Table Info...</p> <p>Sound Manager...</p> <p>Image Manager...</p> <p>Font Manager...</p> <p>Collection Manager...</p> <p>Magnify</p>	<p>1) Table/Image Manager auswählen....</p>				
<p>Image Manager</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Import Path</th> </tr> </thead> <tbody> <tr> <td>ballimage8</td> <td>D:\Vp\ball\F0\oc01_demo_tier1b...</td> </tr> </tbody> </table> <p>Import</p> <p>Reimport</p> <p>Reimport From</p> <p>Delete</p> <p>Rename</p> <p>Transparency</p>	Name	Import Path	ballimage8	D:\Vp\ball\F0\oc01_demo_tier1b...	<p>2) Transparency anklicken.</p> <p>Hinweis: Ein neu angelegter Tisch hätte hier natürlich noch keine Bilder legen, es geht hier nur um die Transparenz</p>
Name	Import Path				
ballimage8	D:\Vp\ball\F0\oc01_demo_tier1b...				

	Schaltfläche
	<p>3) Eine Farbe in den Basic colors anklicken (ich würde immer schwarz oder weiß empfehlen, das ist noch mal ein eigenes Thema)</p> <p>Im Beispiel: Red:255 Green:255 Blue:255 (entspricht rein weiß)</p>
<p><i>CreateBall.color=rgb(64,0,128)</i> Correspond au violet au-dessus du carré blanc.</p> <p>Das sind die rgb Farben.</p> <p>Anklicken einer andere Farbe ergibt andere rgb Werte. Diese Werte kann man auch verwenden, um die Farbe eines Balles zu definieren. <i>Plunger.CreateBall.color=rgb(64,0,128)</i> würde z.B. einen dunkelblauen Ball ergeben.</p> <p><i>Plunger.CreateBall.color=rgb(255,255,255)</i> Weiss</p>	

Verändern der Ballfarben (RGB)

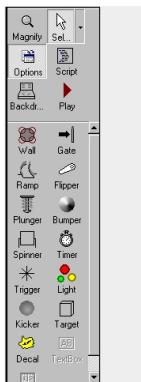
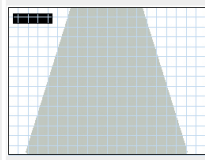
Die Farbe des Balles (*ohne Image !!*) kann über das Script verändert werden. Beispiel:

`Plunger.CreateBall.color=rgb(255,255,255)` (Weiß)

Die einzelnen Werte für RGB (red, green, blue) kann man jedem Malprogramm im Farbwähler entnehmen.

Option Backdrop (Rückwand)

Notiz: Das Aussehen des Tisch wird hier bestimmt

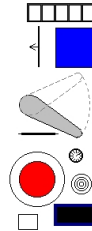
	<p>Backdrop in der Arbeitsleist auswählen.</p>  <p>Auf Options um die optionen der Rückwand anzuzeigen.</p>
<p>-----Editor</p> <p>Show Grid - Zeigt das Gitter an</p> <p>Grid size - größe des Gitterabstand ändern Voreinstellung 50</p> <p>Display Image in Editor - Zeigt den Hintergrund der Rückwand an.</p> <p>-----Colors & Formatting</p> <p>Color - Farbe der Rückwand ändern.</p> <p>Image - Hier das Bild der rückwnd auswählen, (es muß natürlich vorher mit dem Image Manager importiert worden sein) Hinweis: Ich verwend oft 800x600 als standard Grösse Notiz: Maximale Grösse 1024x1024</p> <p>Inclination - Sichbare Schiefele des Tisch</p> <p>Field of View (FOV) - Sichbarer Winkel einstellen . Von Weitwinkel bis Teleobjektiv Sicht</p>	

[zum Index](#)

[Nächste Seite](#)

Rechte Taste Optionen

Gate, Decal, Flipper, Spinner, Timer, Bumper, Kicker, Plunger, TextBox, EMReel



Alle diese Objekte haben 3 Optionen wenn man mit der rechten Maustaste auf sie klickt



Draw in Back

Rückt das Objekt nach hinten

Sie haben eine Wand vor ein Licht gesetzt.	Um das Licht zu bearbeiten müssen sie auf die Wand mit der rechten Taste drücken und Draw in Back wählen Das Licht schiebt sich jetzt nach vorn.	Nachdem sie das Licht bearbeitet haben, möchten sie vielleicht wieder die Wand bearbeiten. Wählen sie die Wand mit der Rechten Taste und die Option Draw in Front um die Wand nach vorne zu schieben.

Achtung die Objekte ändern keine Höhenposition durch den Eingriff.

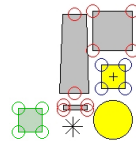
Lock

Lock ermöglicht es Objekte von irrtümlichen Aenderungen zu schützen.

Sie können mehrere Objekte gleichzeitig wählen. Ziehen sie den Mauszeiger mit der Rechts gedruckte Maus über die gewünschten Objekte.	Während die Objekte aktif sind ,	drücken sie die rechte Taste und wählen Lock um die Objekte zu schützen.	Die hellgrauen Objekte sind geschützte Objekte..

Um die Objekte wieder freigeben müssen sie sie wieder Wählen und auf die Option **Lock** drücken.
Die Option Lock ist markier wenn das Objekt geschützt ist.

Ramp, Wall, Target, TriggerCustom, TriggerCircle, LightCustom, LightCircle



Alle diese Objekte haben 9 Optionen wenn man mit der rechten Maustaste auf sie klickt

3 haben wir schon oben gesehen.

Die Objekte **Trigger ShapeCircle** & **Light ShapeCircle** haben keine Option **Add Point**.



Flip X

Verwenden sie ein Dreieck und wählen es. Drücken sie mit der rechten Taste und wählen Option Flip X	Resultat: das Bild hat sich um die X Achse gespiegelt.

Flip Y

Verwenden sie ein Dreieck und wählen es. Drücken sie mit der rechten Taste und wählen Option Flip Y	Resultat: das Bild hat sich um die Y Achse gespiegelt.


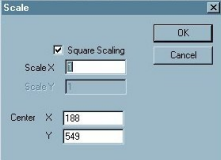

Rotate - Rotation eines Objekts

Verwenden sie wieder ein Dreieck Drücken sie mit der rechten Taste und wählen Option Rotate	Ein Dialogfenster öffnet sich. Geben sie 180 bei 0 ein.	Resultat: das Bild hat sich

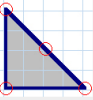
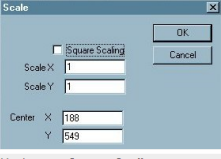

Center X & Y sind das Zentrum für die Rotation

um 180
Grad
gedreht.

Scale - skaliert ein Objekt




		
Verwenden sie wieder ein Dreieck Drücken sie mit der rechten Taste und wählen Option Scale .	Ein Dialogfenster öffnet sich. Geben sie den Wert 2 in Scale X und drücken OK Center X & Y sind das Zentrum der Scalerung (stimmt aber leider nicht richtig)	Das Dreieck hat sich verdoppelt.

Square Scaling unmarkiert




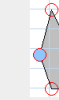
		
Verwenden sie wieder das Dreieck von vorher und Drücken sie mit der rechten Taste und wählen Option Scale .	Entfernen sie den Haken von Square Scaling . Geben sie für Scale X 5 ein und 1 für Scale Y Center X & Y sind das Zentrum der Scalerung (stimmt aber leider nicht richtig)	5 verkleinert um die Hälfte und 1 erhält die Länge.

Translate - verschiebt ein Objekt

Center X & Y sont le centre de redimensionnement de l'objet

		
Verwenden sie wieder ein Dreieck Drücken sie mit der rechten Taste und wählen Option Translate .	Ein Dialogfenster öffnet sich. 50 in Offset X Das Objekt bewegt sich um 50 Einheiten nach rechts X - nach Links und Rechts Y - nach Oben und Unten	Wenn die Option Grid Size in Options Table auf 50 gesetzt ist wird sich das Dreieck um ein ganzes Quadrat bewegen.

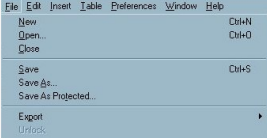
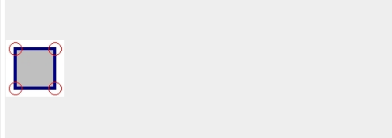
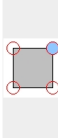


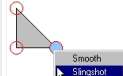


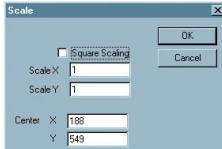
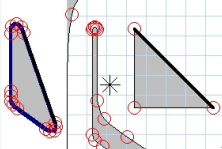


Add Point - Mit dieser Option erstellen sie einen neuen Kontrollpunkt um die Form eines Objekts zu ändern.

			
Verwenden sie wieder ein Dreieck. Bewegen den Zeiger auf ein Segment zu wo sie einen neuen Kontrollpunkt haben wollen und Drücken sie mit der rechten Taste und wählen sie Option Add Point	Ein neuer Kontrollpunkt wurde erstellt.	Wählen sie ein Punkt mit der linken Taste und bewegen sie sich an die gewünschte Position.	Lassen sie die Taste los wenn sie an der richtigen Posditiön sind.

[Zum Index](#)

[Nächste Seite](#)

Einen Slingshot schnell gestalten.

			
Erstmal mit File / New einen neuen Tisch öffnen.	1) Erzeuge eine Wall, benannt wall1	2) Den Punkt oben rechts anklicken	
			
3) Den Punkt löschen (Entf)	4) Den Punkt unten rechts rechts-klicken	5) Slingshot auswählen	
			
6) wird zum Slingshot Steuerband	7) Nochmal auf die dreieckige Wall rechts-klicken und Scale auswählen	8) 2 in die ScaleX Box eingeben & ok klicken Das macht das Dreieck 2x so groß.	
			
9) Tausche den LeftSlingshot gegen das Dreieck aus , den LeftSlingshot einfach neben den Tisch ziehen.	10) Das Script für den neuen Slingshot <pre>Sub Wall1_SlingShot() PlaySound "Bumper" End Sub</pre> <p>Achtung: Evt. muß du den Sub selbst ins Script schreiben, kopieren und einfügen scheint bei Manchen nicht zu funktionieren. Das Sub kann überall im Scrip stehen, vielleicht ganz ans Ende.</p>	11) und jetzt mal ausprobieren...	

Notiz: Frühere Version von VP hatte eine Compile Option, diese ist jetzt nicht mehr sichtbar, sie wird automatisch bei Play ausgeführt

Achtung:

```
Sub Wall1_Hit() <<<<<< !!!
  PlaySound "Bumper"
End Sub
```

ist nicht das Script für den Slingshot sondern für die dazugehörige Wall1.

Fortgeschrittene Technik: Un realistischere Schleuderbander (Slingshot) zu erstellen

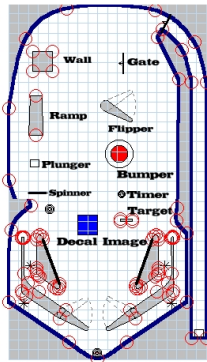
- 1) Das Band in kleinere Teile zerschneide (10 Teile) . Ein band ist in der Mitte sensibler als am Rand . Das verhindert das der Ball auf den seiten zu stark abbrems.
 - 2) Das zweite Beispiel zeigt wie man eine Zielscheibe mit slingshot benutzt . Die Zielscheibe geht nur runter wenn man das slingshot tift . Die anderen Seit reagieren nicht wie bei einer gewonlicher Zielscheibe. Das Slingshot steuert die Zielscheibe.
- Alternative:
- Um eine Zielscheibe nicht auf der falschen Seite zu treffen kann man auch eine zusetzliche dünne Zielscheibe vor der eigentlichen zielscheibe setzen . Die dünne Scheibe wird dann verwendet um beide Zielscheiben zusammen herunter zusetzen. In diesen Fall verwendet man kein Slingshot.

[zum Index](#)

[Nächste Seite](#)

Editor Objekte

Decal



Decal

Decals braucht man um Bilder oder Text auf dem Tisch oder das Backdrop zu bringen.



Decal Schaltfläche

<p>File Edit Insert Table Preferences Window Help</p> <p>New Ctrl+N Open... Ctrl+O Close Save Ctrl+S Save As... Save As Protected... Export Undo</p>			
<p>Ein Decal erzeugen: Zuerst mit File / New einen neuen Tisch öffnen.</p>	<p>Auf die Decal Schaltfläche klicken</p>	<p>er sollte jetzt so aussehen, klicken wo das Decal platziert werden soll.</p>	<p>So sieht es im Editor aus.</p>

Jetzt sehen wir uns die Decal Optionen an.

Options im ToolBar (Werkzeugleiste)	anklicken und das Decal durch anklicken auswählen.

Jetzt sehen wir die Decal Eigenschaften.

<p>Decal</p> <p>Colors & Formatting</p> <p>Type: DecalImage</p> <p>Text: <input type="text"/></p> <p><input type="checkbox"/> Vertical Text</p> <p>Font Color: <input type="color"/></p> <p>Font: Arial Black</p> <p>Image: <input type="text"/></p> <p>Sizing: ManualSize</p> <p>Position</p> <p>X: 165 Y: 478 Width: 100 Height: 100 Rotation: 0 Surface: <input type="text"/></p>	<p>Decal - Art des Objektes, das ausgewählt ist.</p> <p>Type - DecalImage oder DecalText stehen zur Auswahl</p> <p>Text - Hier den Text eingeben für DecalText</p> <p>Vertical Text - Voreingestellt ist "aus". Aktivieren für vertikalen DecalText</p> <p>Font Color - Wählt die Farbe für den DecalText</p> <p>Font - Wählt den Font (Schriftart) für den DecalText</p> <p>Image - Hier ein Bild auswählen für DecalImage</p> <p>Sizing - Zur Auswahl: ManualSize, AutoSize, AutoWidth.</p> <p>X - Zeigt die X Position dese Decals auf dem Tisch</p> <p>Y - Zeigt die Y Position dese Decals auf dem Tisch</p> <p>Width - Die Breite des Decals.</p> <p>Height - Die Höhe des Decals.</p> <p>Rotation - Stellt das Decal schräg.</p> <p>Surface - Wählt Oberfläche des Objektes wo das Decal liegen soll.</p>
<p>Police</p> <p>Police: <input type="text"/> Style: Normal Taille: 8</p> <p>Font list: Arial, Arial Black, Arial Narrow, Autocrat, Arial Unicode MS, AvantGarde Bk BT, AvantGarde Md BT</p> <p>Font list: Normal, Itaque, Gras, Gras Itaque</p> <p>Size list: 8, 9, 10, 11, 12, 14, 16</p> <p>Preview: AaBbVvZz</p> <p>Script: Latin</p>	<p>Font öffnet das Fenster für Textattribute</p> <p>Notiz: Die Fonts Ttf für spezielle Tische müsse unter Windows zuerst installiert werden bevor man sie auswählen kann.</p> <p>Sample: Textattribut Vorschau</p> <p>Script: Eigenschaft des Ländercode, so lass wie es steht.</p> <p>Notiz: Hat nicht mit dem Scripteditor zu tun.</p>


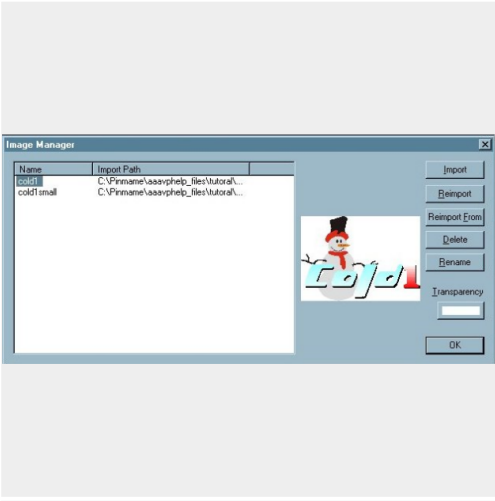
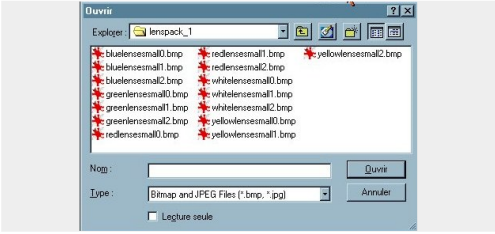

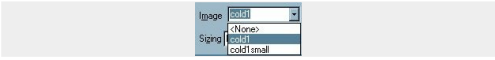
Das Folgende Beispiel zeigt, wie man ein Bild für ein Decal importiert und das es mit zu kleiner Pixelanzahl nicht gut aussieht.

[DemoDecal.zip](#)

In diesem Beispiel sind zwei Bmp.Bilder enthalten. cold1small.bmp ist 50x30 pixel & cold1.bmp ist 300x215 pixel groß.

Man kann erkennen, das das kleinere Bild zu niveln aussieht

Ein Bild importieren:

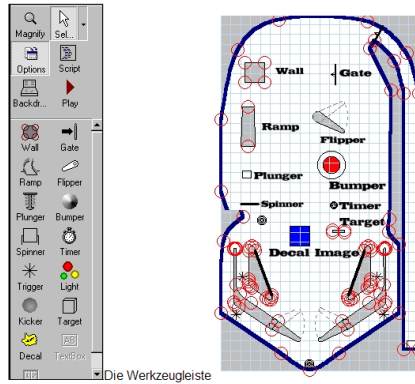
	<p>Table/Image Manager auswählen</p>
	<p>Import öffnet das folgende Fenster:</p> <p>Tip: Es gibt oft Probleme mit "Transparency", also die Farbe die unsichtbar erscheinen soll. Obwohl der ganze Hintergrund weiß sein sollte, bleiben oft eine Art von "Schmutzflecken" also weiße Stellen, die nicht unsichtbar geworden sind. Da hilft folgende Methode: Das Bild in "Paint" laden, die Hintergrundfarbe per Farbeimer mit einer möglichst scheußlichen Farbe auffüllen. Jetzt werden alle Bereiche sichtbar, die offensichtlich doch eine Nuance anders sind (ist das eine Microsoft-Schikane?), diese manuell mit der scheußlichen Farbe übermalen. Wenn alles erfasst ist, wieder mit dem Farbeimer in die Hintergrundfarbe (ich empfehle der Einfachheit halber schwarz oder weiß) zurückverwandeln, und Du hast garantiert alle Pixel erwischt.</p>
	<p>Hier das Bild zum importieren auswählen. Achtung: Das kann auf deinem PC etwas anders aussehen.</p> <p>Wenn ein Bild ausgewählt ist, wird es im Image Manager angezeigt. OK klicken</p>
	<p>Nochmal: das Decal muß ausgewählt sein</p>
	<p>In image das Bild auswählen.</p>

[zum Index](#)

[Nächste Seite](#)

Editor Objekte

Plunger & Kicker



Die Werkzeugleiste

Plunger

Der Plunger schießt den Ball in das Spiel, gewöhnlich ausgelöst durch die Enter Taste.



Plunger Schaltfläche

File	Edit	Insert	Table	Preferences	Window	Help
New						Ctrl+N
Open...						Ctrl+O
Close						
Save						Ctrl+S
Save As...						
Save As Protected...						
Export						U1955...
<p>Zuerst erzeugen wir einen Plunger: Falls noch nicht geschehen, einen neuen Tisch öffnen mit File / New</p>						
<p>Klicke auf die Plunger-Schaltfläche</p>		<p>Bewege den Mauszeiger über den Tisch, er sollte jetzt diese Form annehmen, klicke wo der Plunger platziert werden soll</p>		<p>so sieht er dann aus.</p>		

Hinweis: Der nächste Schritt ist nur um zu zeigen wie ein Plunger erzeugt wird, das ist aber eigentlich sehr selten nötig, denn der neue Tisch enthält ja schon einen, inklusive Script. (Der Plunger braucht auf beiden Seiten eine Wall, sonst fällt der Ball herunter)

Jetzt können wir uns die Plunger Optionen genauer ansehen.

Klicke Options im ToolBar (Werkzeugleiste) an	und achte darauf das der Plunger ausgewählt ist

Jetzt werden die Plunger Eigenschaften sichtbar.

Plunger - Art des Objektes, das ausgewählt ist.

Name - *Plunger* Der Name des Plungers, könnte auch verändert werden. **Tip** Wie gesagt, das macht keinen Sinn wenn man nur eine Plunger auf dem Tisch hat..... (ein neuer Plunger braucht dann auch sein eigens Script)

-----Position

X - Zeigt die X Position Des Plungers auf dem Tisch.

Y - Zeigt die Y Position Des Plungers auf dem Tisch.

Surface - Surface meint die Höhe eines anderen Objektes, die der Plunger annehmen soll. Voreingestellt ist <None> das heißt auf Höhe des Tisches. Zur Erinnerung: Der Plunger braucht Walls, die den Ball am Platz halten.

-----State

Pull Speed - Die Geschwindigkeit des Zurückziehens.

Release Speed - Die Geschwindigkeit des Loslassens (Ball wird abgeschossen)

-----Misc

Timer Enabled - Standart nicht aktiv. Der Time ist in Ruhsellung. Er wird normalerweise im Script eingeschaltet, und wird mit *PlungerName_Timer()* angesprochen.

Timer Interval - Zeitintervall bis der Timer anspringt.

UserValue - Frei zu verfung stehende Variable. Braucht keine Dim Deklaration. Muss jedoch vor Gebrauch initialisiert werden. Die Variable kann den Type Zeichenkette, Numerisch, Boolean haben.
Notiz: Ich brauche diese Variable ab und zu als Markierer.

Ein Beispiel-Script für einen **Plunger**. Hinweis: Das ist nur ein Basic-Script für einen Plunger. Man könnte z.B. noch ein Playsound "squeekylunger" für das Zurückziehen & Playsound "ballaunch" für das Abschießen hinzufügen.....

Das Beispiel ist ein Ausschnitt aus dem "Standart" Keycodes Sub, Deshalb fehlt hier das End Sub.

```
Sub Table1_KeyDown(ByVal keycode)
```

```
    If keycode = PlungerKey Then
        Plunger.PullBack
    End If
```

```
Sub Table1_KeyUp(ByVal keycode)
```

```
    If keycode = PlungerKey Then
        Plunger.Fire
    End If
```

Kicker

Der Kickers kann benutzt werden um einen Ball festzuhalten oder wegzuschießen.



Kicker Schaltfläche

File Edit Insert Table Preferences Window Help			
New Open... Close		Kicker	
Save Save As... Save As Projected...			
Export Unison			
Zuerst erzeugen wir einen Kicker: Falls noch nicht geschehen, einen neuen Tisch öffnen mit File / New	Klicke die Kicker-Schaltfläche an	Bewege den Mauszeiger über den Tisch, er sollte jetzt so aussehen. Klicke wo der Kicker platziert werden soll.	so wird er im Editor angezeigt

Jetzt können wir uns die Kicker Optionen ansehen.

Options	
Klicke auf Options im ToolBar	und wähle den Kicker durch anklicken. Das Objekt hat kein blauer Rand.....

Jetzt werden die Kicker Eigenschaften angezeigt.

Kicker - Art des Objektes das ausgewählt ist.

Name - *kicker1* Name des Kickers, kann umbenannt werden. **Tip** Zusammenhängende Kicker sollten auch zusammenhängend benannt werden, der Übersichtlichkeit wegen.

-----Colors & Formatting

Color - Öffnet das Farbwahl-Fenster, die Farbe kann verändert werden.

Display - 3 Möglichkeiten zur Auswahl, KickerHole (in Form eines Loches -Voreinstellung) KickerCup (in Form dieser kleinen Topfchen) & KickerHidden (unsichtbar)

-----Position

X - Zeigt die X Position des Kickers auf dem Tisch.

Y - Zeigt die Y Position des Kickers auf dem Tisch.

Surface - Surface meint die Höhe eines anderen Objektes, die der Kicker annehmen soll. Voreingestellt ist <None> das heißt auf Höhe des Tisches. Beispiel: Wenn der Ball nicht halb in den Kicker versinken soll, ist das hier eine gute Gelegenheit Surface auf eine Wall die nur ca 20 hoch ist zu stellen.

-----State

Enabled - Wenn aktiviert arbeitet der kicker immer, das heißt in dem Fall allerdings er hält den Ball fest und gibt ihn nicht mehr her. Also: Wie wo und wann der Kicker arbeitet muß über das Script geregelt werden.

-----Misc

Timer Enabled - Standart nicht aktiv. Der Time ist in Ruhsellung. Er wird normalerweise im Script eingeschaltet, und wird mit *LichtName_Timer()* angesprochen.

Timer Interval - Zeitintervall bis der Timer anspringt.

UserValue -Frei zu verflungung stehende Variable. Braucht keine Dim Deklaration. Muss jedoch vor Gebrauch initialisiert werden. Die Variable kann den Type Zeichenkette, Numerisch, Boolean haben.
Notiz: Ich brauche diese Variable ab und zu als Markierer.

Hier ein Beispiel: [demo0kickertimer.zip](#)

Anzeigemodus:
KickerHole, KickerCup
KickerHidden- erstellt ein unsibarer Kicker

Script ist grün geschrieben zur Unterscheidung:

Kicker Funktionen:

Ein Kicker kann:

Kicker1.DestroyBall - lässt den Ball verschwinden

Kicker1.CreateBall - erzeugt einen Ball

Kicker1.Kick - den Kicker raus schiessen

```
Sub kicker1_hit()
Kicker1.Kick 0,10
end sub
```

Der erste Wert setzt die Richtung, 0 (Grad) schießt nach oben. (90 würde also nach rechts schießen)
Der zweite Wert setzt die Stärke des Schusses.

Winkel von 0 bis 359 sind also möglich, winkel über 180 Grad können auch als Negativzahlen geschrieben werden. (210° = -90°)
Man kann sich die Richtungen wie ein Zifferblatt einer Uhr vorstellen:

```
..... 0
.....305.....45
.....270.....90
.....225.....135
.....180
```

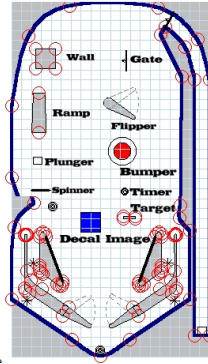
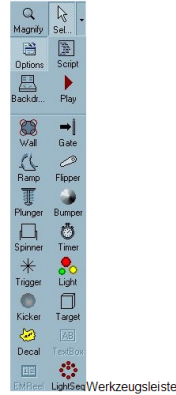
Siehe auch: [Faq](#) für mehr Hinweise zum Thema Kicker

[zum Index](#)

[Nächste Seite](#)

Editor Objekte

Walls & Targets



The Walls



Wall Schaltfläche

Einen Mauer (Wall) erzeugen: Einen neuen Tisch öffnen mit File / New .	Auf die Wall Schaltfläche klicken.	Der Mauszeiger verwandelt sich in Wall Mauszeiger. Bewegen sie den Mauszeiger über den Tisch, wo sie die Mauer erzeugen wollen

Die roten Kreisen sind Kontrollpunkte um die Form zu ändern.

[Plus sur les points de controles](#). Siehe Kontrollpunkte

Wir betrachten die Mauer (Wall) Optionen:

Options in ToolBar aktivieren	Mauer mit der Maus auswählen

Jetzt sehen wir die Wall Eigenschaften.

Wall	Wall - Art des Objektes das ausgewählt ist.
Name: Wall	Name - Wall1 Name der Mauer, kann geändert werden. Die Mauer wird mit dem Name im Script adressiert.
Colors & Formatting	-----Colors & Formatting
<input checked="" type="checkbox"/> Top Visible	Top Visible Oberseite sichtbar. Standardoption: Aktiv
Top Image: [Dropdown]	Top Image - Bild für die Oberfläche auswählen. Notize: Das Bild überdeckt die ganze Fläche des Tisch und nicht ist nicht nur für die ausgewählte Mauer. Für ein Standard Tisch ist das Bild 512 X 1024 gross. Hat der andere Masse sollte das unbedingt die richtigen Proportionen habe.
<input type="checkbox"/> Display Image in Editor	Display Image in Editor - Zeigt das oben geladenen Bild an. Rat: Diese Option ausschalten wenn die Manipulation des Tisches zu langsam wird. Der Zoom zum beispiel extrem langsam werden wenn viele Objekte auf dem Tisch sind.
Top Color: [Color Picker]	Top Color - Oeffnet das Farbwahlfenster für die Obere flächenfarbe.
<input checked="" type="checkbox"/> Side Visible	Side Image - Seitenbild für die Wand. Wird wie das Oberflächen gewählt jedoch ist das Bild nur für das gewählte Bild gültig, und sollte der höhe und breite der Wand entsprechen. (Siehe Bilder für Seitenwände)
Side Image: [Dropdown]	Side Color - Oeffnet das Farbwahlfenster für Farbe der Seitenwände.
Side Color: [Color Picker]	Cast Shadow Zeit den Schatten der Wand an. Standardoption: aktiv
<input checked="" type="checkbox"/> Casts Shadow	-----Position
Position	Top Height - Obere Mauerhöhe. 50 entspricht der standart Höhe und gleich hoch wie der Ball. Jede neue Mauer ist mit diesem Wert eingestellt. hauteur.
Top Height: 50	Bottom Height - Untere Mauerhöhe. 0 entspricht dem Tischboden. Will man die Mauer versenken (mit dropped) muss man den Wert auf -1 setzen. Man braucht jedoch ein Script um die Mauer zu versenken.
Bottom Height: 0	-----State
State	Has Hit Event - Will man die Mauer mit einen Ereigniss im Script ausstatten, muss man diese Option markieren
<input type="checkbox"/> Has Hit Event	Hit Threshold - Reaktionskraft, mit dieser Kraft muss das Objekt getroffen werden um ein Ereigniss auszulösen. Zum Beispiel die Mauer zu versenken.
Hit Threshold: 1	Slingshot Force - Ruckstosskraft. Der Ball kann stärker abgestossen werden als bei sein Aufschlag.
Slingshot Force: 0	Elasticity - Die Elastizität ist dem Slingshot Force ähnlich. Steuert de Aufprall
Elasticity: 0.3	Can Drop - Die Mauer kann versenken werden wenn die Option aktiv ist.
<input type="checkbox"/> Can Drop	-----Misc
Mac	Timer Enabled - Standart nicht aktiv. Der Time ist in Ruhrsellung. Er wird normalerweise im

Script eingeschaltet, und wird mit `WallName_Timer()` angesprochen.

Timer Interval - Zeitintervall bis der Timer anspringt.

UserValue - Frei zu Verfügung stehende Variable. Braucht keine Dim Deklaration. Muss jedoch vor Gebrauch initialisiert werden. Die Variable kann den Type Zeichenkette, Numerisch, Boolean haben.

Notiz: Ich brauche diese Variable ab und zu als Markierer.

Die Mauern und Zielscheiben (Targets) haben den gleichen Aufbau. Der Unterschied liegt nur in der Einstellung der Optionen. In der Zielscheibe ist die Option **Has Hit Event** markiert und müssen im Script verbunden werden. Das Element sollte ein Aussagekräftiger Namen bekommen um leichter im Script identifiziert werden zu können.

Editor Objekt

Targets



Target Schaltfläche

Eine Zielscheibe (Target) erzeugen: Einen neuen Tisch öffnen mit File / New .	Auf die Target! Schaltfläche klicken.	Der Mauszeiger verwandelt sich in Target Mauszeiger. Bewegen sie den Mauszeiger über den Tisch, von sie die Zielscheibe erzeugen wollen
		Sie haben eine Zielscheibe erstellt.

Wir betrachten die Zielscheiben (Target) Optionen:

Options in ToolBar aktivieren	Target mit der Maus auswählen

Jetzt sehen wir die Target Eigenschaften. Einige sind anders eingestellt.

Wall - Art des Objektes das ausgewählt ist.

Name - Wall1 Name der Zielscheibe sollte Aussagekräftig umbenennet werden. Ziel1,Ziel3 usw. Die Zielscheiber wird mit dem Namen im Script adressiert.

-----Colors & Formatting

Top Visible Oberseite sichtbar. Standardoption: Aktiv

Top Image - Bild für die Oberfläche auswählen.
Notize: Das Bild überdeckt die ganze Fläche des Tisch und nicht ist nicht nur für die ausgewählte Mauer. Für ein Standard Tisch ist das Bild 512 X 1024 gross. Hat der andere Masse sollte das unbedingt die richtigen Proportionen habe.

Display Image in Editor - Zeigt das oben geladenen Bild an.
Rat: Diese Option ausschalten wenn die Manipulation des Tisches zu langsam wird. Der Zoom zum beispiel extrem langsam werden wenn viele Objekte auf dem Tisch sind.

Top Color - Öffnet das Farbwahlfenster für die Obere flächenfarbe.

Side Image - Seitenbild für die Wand. Wird wie das Oberflächen gewählt jedoch ist das Bild nur für das gewählte Bild gültig, und sollte der höhe und breite der Wand entsprechen. Anders eingestellt als bei einer Mauer. (Siehe Bilder für Seitenwände)

Side Color - Öffnet das Farbwahlfenster für Farbe der Seitenwände.

Cast Shadow Zeit den Schatten der Wand an. Standardoption: aktiv

-----Position

Top Height - Obere Mauerhöhe. 50 entspricht der standart Höhe und gleich hoch wie der Ball. Jede neue Mauer ist mit diesem Wert eingestellt. hauteur.

Bottom Height - Untere Mauerhöhe. 0 entspricht dem Tischboden. Will man die Mauer versenken (mit dropped) muss man den Wert auf -1 setzen. Man braucht jedoch ein Script um die Mauer zu versenken.

-----State

Has Hit Event - Ist bei einem Target standart gemäss aktiv. Eine Zielscheibe braucht ein Ereigniss im Script.

Hit Threshold - Reaktionskraft, mit dieser Kraft muss das Objekt getroffen werden um ein Ereigniss auszulösen. Zum Beispiel die Mauer zu versenken.

Slingshot Force - Rückstosskraft. Der Ball kann stärker abgestossen werden als bei sein Aufschlag.

Elasticity - Die Elastizität ist dem **Slingshot Force** ähnlich. Steuert de Aufprall

Can Drop - Die Mauer kann versenkt werden wenn die Option aktiv ist.

-----Misc

Timer Enabled - Standart nicht aktiv. Der Time ist in Ruhrsellung. Er wird normalerweise im Script eingeschaltet, und wird mit `WallName_Timer()` angesprochen.

Timer Interval - Zeitintervall bis der Timer anspringt.

UserValue - Frei zu Verfügung stehende Variable. Braucht keine Dim Deklaration. Muss jedoch vor Gebrauch initialisiert werden. Die Variable kann den Type Zeichenkette, Numensch, Boolean haben.
Notiz: Ich brauche diese Variable ab und zu als Markierer.

Dies ist das Basis Script für eine Zielscheibe [demo1targetdrop.zip](#)

'Kommetar mit vorgehendem Zeichen'

Sub Target1_Hit() Wenn getroffen Aufgerufene Routine für Target1

Playsound "targetdropping" Spielt ein Sound

AddScore (50) Fügt 50 Punkte zum Resultat zu. Da diese Routine noch nicht geschrieben ist, lässt man die Linie als Kommentar.

Target1 IsDropped=True 'Die Zielscheibe neht runter

```
CheckTarget() 'Routine überprüft ob alle Zielscheiben getroffen sindbevor man sie wieder rauf zieht.  
End Sub 'Ende der Routine  
Sub Target2_Hit() 'Wenn getroffen Aufgerufene Routine für Target12  
Playsound "targetdropping" 'Spielt ein Sound  
'AddScore (50) 'Fügt 50 Punkte zum Resultat zu . Da diese Routine noch nicht geschrieben ist , lässt man die Linie als  
Kommentar.  
Target1.IsDropped=True 'Die Zielscheibe geht runter  
CheckTarget() 'Routine überprüft ob alle Zielscheiben getroffen sindbevor man sie wieder rauf zieht.  
End Sub 'Ende der Routine  
Routine überprüft ob beide Zielscheiben getroffen sind  
Sub CheckTarget()  
'Wenn Target1 unten und Target2 unten dann werden wieder beide raufgesetzt  
If Target1.IsDropped = True And Target2.IsDropped = True Then  
Target1.IsDropped=False 'remonte la cible 1 Target1  
Target2.IsDropped = False 'remonte la cible 2 Target2  
End If  
End Sub
```

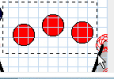

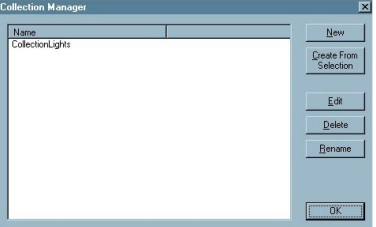
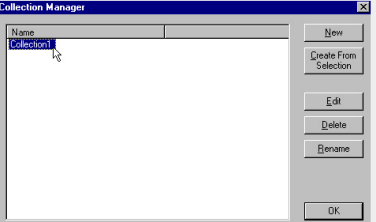
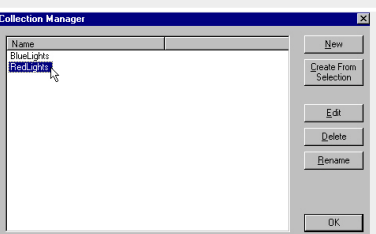
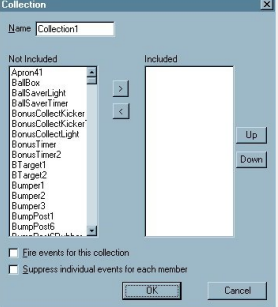
Siehe auch [Targets-BasicTutorial.txt](#) für mehr Infos zu Zielscheiben in [ShivaEngine1_7.zip](#)

[Zum Index](#)

[Nächste Seite](#)

Editor Objekte

Wie man Lichter Gruppen mit dem Collection Manager erstellt

	Lichter auswählen womit sie eine Gruppe erstellen wollen.
	Table/Collection Manager auswählen
	Klicke auf Create From Selection
	Erstellen sie eine Gruppe und nennen sie Collection1 N ew - erstellt ein neue Gruppe (Collection) C reate From Selection - Erstellt eine Gruppe aus den ausgewählten Objekten vom Tischeditor. E dit - Öffnet den Gruppeneeditor D elele - Löscht eine Gruppe R ename - Eine Gruppe umbenennen
	Es befinden sich schon 2 Gruppen in der Datei. Wählen sie RedLights und klicken auf E dit
<p>Das folgende Fenster öffnet sie wenn sie auf Edit drücken .</p> 	N ame - Collection1 Der Name der Gruppen, kann umbenannt werden, und ist somit im Script verwendbar. Not Included - Liste der Objekte die nicht in der Gruppe eingeschlossen sind. Included - Liste der Objekte die sich der Gruppe befinden. > - neues Objekt in die Gruppe einfügen <- Objekt aus der Gruppe entfernen. Up - Das ausgewählte Objekt in der Liste höher stellen. Down - Das ausgewählte Objekt in der Liste tiefer stellen. Fire Events for this collection - Fire Ereignisse?? Suppress individual events for each member - Zugriff auf einzelne Objekte in der Liste verwalten.

Hier ein Demo für ein Script dass den Zustand von Lichter und der Status *Light.State* verwendet.

Verwenden sie [DemoLights.zip](#)

Verwenden des Befehls "For Each" in einem Script.

Beispiel:

Dim light

```
For each light in RedLights  
light.State = LightStateOn  
Next
```

Die Variable 'light' wird von allen Licht Elemente der Gruppe verwendet unabhängig von der Größe. Diese Methode erlaubt Änderungen für eine komplette Gruppe.

In der Demo, löst die Taste *keycode* eine Routine aus die den Zustand der Lichter ändert.

```
If keycode = PlungerKey Then  
Plunger.Fire
```

In diesem Teil löst das nachlassen der Plungertaste ein Löschen der Lichter aus.

```
If keycode = LeftFlipperKey Then  
LeftFlipper.RotateToStart  
PlaySound "FlipperDown"
```

In diesem Teil löst die linke Flippertaste das Erlöschen der Lichter aus.

```
If keycode = RightFlipperKey Then
```

```
RightFlipper.RotateToStart
PlaySound "FlipperDown"
```

In diesem Teil löst die rechte Flippertaste das blinken der Lichter aus.

```
'Lichter Zustand 0 = LightStateOff (gelöscht)
' 1 = LightStateOn (brennt)
' 2 = LightStateBlinking (blinkt)
```

```
***** Array
```

Verwendung einer Array Variable um den Lichter Zustand zu ändern

```
LightsArray = Array(Light1,Light2,Light3,Light4,Light5,Light6,Light7,Light8)
For a = 0 to 7 'a ist ein Wert zwischen 0 und 7 = 8 lights
LightsArray(a).State=2 'Licht blinkt
Next
```

```
***** Collections
```

' Green Lights & Red Lights & Blue Lights sind Lichtgruppen

' Um die einzelnen Elemente zu sehen, wählen sie

' Table / Collection Manager (Select a collection) / Edit

' Die Gruppe verwendet den Befehl "For Each"

For Each Light in GreenLights

Light.state = LightStateBlinking 'or 2

Next

For Each Light in Redlights

Light.State=2 'ou LightStateBlinking 'les 3 lumières rouges au-dessus des the flippers

Next

'Man kann auch Tabellenvariablen (Arrayvariablen) verwenden. "For Each" ist jedoch oft einfacher.

For a = 0 to 11 '0 bis 11 = 12 Lichter, 0 zählt als erste Zahl

BlueLights(a).State=2 ' 12 blaue Lichter blinken in Uhrzeigerichtung (Siehe Pattern Option für den Effekt)

Next

```
***** Setting Lights
```

'Diese Option braucht länger zu schreiben :)

Dim Red12Lights(11)

Set Red12Lights(0) = r1

Set Red12Lights(1) = r2

Set Red12Lights(2) = r3

Set Red12Lights(3) = r4

Set Red12Lights(4) = r5

Set Red12Lights(5) = r6

Set Red12Lights(6) = r7

Set Red12Lights(7) = r8

Set Red12Lights(8) = r9

Set Red12Lights(9) = r10

Set Red12Lights(10) = r11

Set Red12Lights(11) = r12

For a = 0 to 11 '0 bis 11 = 12 Lichter

Red12Lights(a).State=2 '12 rote Lichter blinken in Gegenuhrzeigerichtung (Siehe Pattern Option für den Effekt)

Next

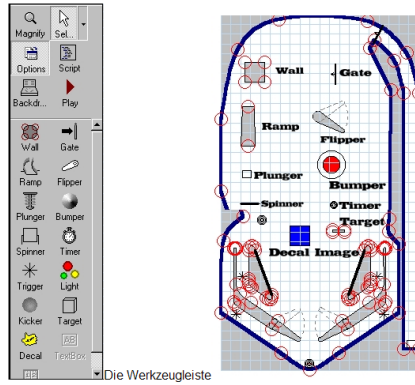
Achtung die Beispieldateien sind nicht übersetzt worden.

Notiz: Jede Methode hat vor und Nachteile es kommt auf die verwend drauf an. Die Letzt trotz mehr Schreibaufwand kann durch bessere Lesbarkeit des Scripts in aufwendigen und schwierigen Scripts die Mehrarbeit ausgleichen. Soll ein Licht einzel sowohl als Gruppe gesteuert werde ist sie die bessere Wahl.

[zum Index](#)

Editor Objekte

Lights&Bumpers



Lights

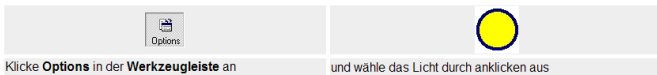


Light Schaltfläche

File	Edit	Insert	Table	Preferences	Window	Help
New	Ctrl+N					
Open...	Ctrl+O					
Close						
Save	Ctrl+S					
Save As...						
Save As Projected...						
Export						
Undo						

Zuerst setzen wir mal ein Licht: Falls noch kein Tisch geöffnet ist klicke File / New	Klicke auf die Lights Schaltfläche,	bewege den Mauszeiger über den Tisch an dem du arbeiten willst, es sollte jetzt so aussehen, klicke wo das Licht gesetzt werden soll,	und hier ist das Licht.

Nachdem wir ein Licht erzeugt haben, können wir uns die Light Eigenschaften Optionen ansehen.



Jetzt kannst Du die Eigenschaften des Lichtes betrachten und verändern.

Light - Art des Objektes, das ausgewählt ist.

Name - Light1 der Name des Lichtes, kann umbenannt werden. **Tip:** Es macht Sinn zusammenhängende Lichter auch zusammenhängend zu benennen z.B. HeartLight1, HeartLight2.

-----Colors & Formating

Shape - 2 Optionen - ShapeCircle & ShapeCustom (frei veränderbare Form)

Radius - Stellt die Größe des Lichtes ein. Voreingestellt ist 50

Color - Öffnet die Farbeinstellungs-Box für die Lichtfarbe.

BorderWidth - Dicke der "Lichtumrandungslinie".

BorderColor - Farbe der "Lichtumrandungslinie".

-----Position

X - Zeigt die X Position des Lichtes auf dem Tisch an.

Y - Zeigt die Y Position des Lichtes auf dem Tisch an.

Surface - In dieser Scroll Down Box weist man das Licht der Höhe einer bestimmten Oberfläche zu z.B. <None> (ist voreingestellt), das bedeutet auf die Ebene des Tisches. Oder <Wall1>, das bedeutet das Licht "schwebt" auf der Höhe von Wall1.

-----State

State - Es gibt 3 Optionen - LightStateOff - LightStateOn - LightStateBlinking (oder 0 - 1 - 2)

Notiz: Wenn **LightStates** (Zustände) im Script geändert werden- können sie in unten gezeigter Weise geändert werden.

Light1.State=LightStateOff	oder	Light1.State=0
Light1.State=LightStateOn		Light1.State=1
Light1.State=LightStateBlinking		Light1.State=2

Blink Pattern - 10 ist voreingestellt. Das bedeutet an-aus (wiederholt) falls LightStateBlinking oder 2 (Script) gesetzt ist

Notiz: Ein Pattern ist eine wiederholte Sequenz. Beispiel: 100100010 1=an 0=aus

Blink Interval - 125 ist die benötigte Zeit, pro Schritt im Muster. Hinweis: 1000 = 1 Sekunde

-----Misc

Timer Enabled - Standart nicht aktiv. Der Time ist in Ruhrsellung. Er wird normalerweise im

Script eingeschaltet, und wird mit `LichtName_Timer()` angesprochen.

Timer Interval - Zeitintervall bis der Timer anspringt.

UserValue - Frei zu verwendende Variable. Braucht keine Dim Deklaration. Muss jedoch vor Gebrauch initialisiert werden. Die Variable kann den Typ Zeichenkette, Numerisch, Boolean haben.

Notiz: Ich brauche diese Variable ab und zu als Markierer.



Hinweis: **InShapeCustom** können die kleinen Kreise (Points) angeklickt und verschoben werden. Dadurch ändert sich die Form des Lichtes. Näheres unter [Erzeugen eines einfachen "Sling Shot"](#)

Bumper

Bumper können 3 (Light)States haben: on, off & blinking.

Sie können überall auf dem Tisch plaziert werden.



Bumper Schaltfläche

<p>Zuerst mal einen neuen Bumper erzeugen: Mit File / New einen neuen Tisch öffnen.</p>	<p>Klicke auf die Bumper Schaltfläche,</p>	<p>bewege den Mauszeiger über den Tisch an dem du arbeiten willst, es sollte jetzt so aussehen, klicke wo der Bumper gesetzt werden soll,</p>	<p>So sieht er im Editor aus.</p>

Sehen wir uns mal die Bumper Optionen an.

<p>Auf Options der Werkzeugleiste klicken</p>	<p>und dann den Bumper anklicken um ihn auszuwählen,</p>

Die Bumper Eigenschaften werden angezeigt

	<p>Bumper - Art des Objektes, das ausgewählt ist.</p> <p>Name - <i>Bumper1</i> Der Name des Bumpers, kann umbenannt werden. Wird mit diesem Namen im Script angesprochen um seine die Routinen auszuführen.</p> <p>-----Colors & Formatting</p> <p>Color - Öffnet die Farbeinstellungs-Box für die Bumperfarbe.</p> <p>Side Color - Die Seitenfarbe auswählen.</p> <p>Image - Ein Image für die Oberfläche auswählen. Hinweis: Das Bild sollte etwa 300 x 300 pixel groß und im .bmp Format sein. Achtung: das Bild muss transparente Bereiche enthalten, sonst sieht man den Buper nicht mehr leuchten!</p> <p>Radius - Stellt den Durchmesser ein. 45 ist voreingestellt.</p> <p>Overhang - Stellt die Größe des "Überhanges" ein.</p> <p>Cast Shadow - Zeigt Bumpershatten an wenn aktiv</p> <p>-----Position</p> <p>X - Zeigt die X Position des Bumpers auf dem Tisch.</p> <p>Y - Zeigt die Y Position des Bumpers auf dem Tisch.</p> <p>Surface - In dieser Scroll Down Box weist man das Licht der Höhe einer bestimmten Oberfläche zu z.B. <None> (ist voreingestellt), das bedeutet auf die Ebene des Tisches. Oder <Wall1>, das bedeutet das Licht "schwebt" auf der Höhe von Wall1.</p> <p>-----State</p> <p>State - 3 Optionen - LightStateOff - LightStateOn - LightStateBlinking oder (0 - 1 - 2 im Script)</p> <p>Beachten: Wenn die Lightstates eines Bupers per script geändert werden, kann es auf die zwei unten gezeigten Arten gemacht werden. (Das wird selten gemacht, denn Flash When Hit ist voreingestellt, das heißt er leuchtet kurz auf wenn er getroffen wird.)</p> <pre>Bumper1.State=LightStateOff Bumper1.State=0 Bumper1.State=LightStateOn oder Bumper1.State=1 Bumper1.State=LightStateBlinking Bumper1.State=2</pre> <p>Blink Pattern - 1 0 ist voreingestellt, das bedeutet, falls LightStateBlinking eingestellt ist, wiederholtes an (1) - aus (2)</p> <p>Blink Interval - 125 ist die Zeit für einen Durchlauf des Blink Patter. Hinweis: 1000 = 1 Sekunde</p> <p>Flash When Hit - Wenn aktiviert, blitzt der Bumper kurz auf, wenn er getroffen wird.</p> <p>----- Physics</p> <p>Force - Stellt ein wie stark der Ball abprallt.</p> <p>Hit Threshold - Stellt ein wie stark der Ball den Bumper treffen muß um ein abprallen auszulösen. (voreingestellt ist 1)</p>
--	--

-----misc (ecran coupe)

Timer Enabled - Standart nicht aktiv. Der Timer ist in Ruhezustand. Er wird normalerweise im Script eingeschaltet, und wird mit `BumperName_Timer()` angesprochen.

z.B. so:

```
Sub Bumper1_Timer()  
Light1.state = 2  
End Sub
```

Timer Interval - Zeitintervall bis der Timer anspringt.

UserValue - Frei zu Verfügung stehende Variable. Braucht keine Dim Deklaration. Muss jedoch vor Gebrauch initialisiert werden. Die Variable kann den Typ Zeichenkette, Numerisch, Boolean haben.

Notiz: Ich brauche diese Variable ab und zu als Markierer.

Hier einige Erklärungen aus einem Visual Pinball Script zu Thema Light.State & Bumper.State

Script ist zur Unterscheidung grün gefärbt.

[demo2light&bumpersstates.zip](#)

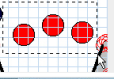

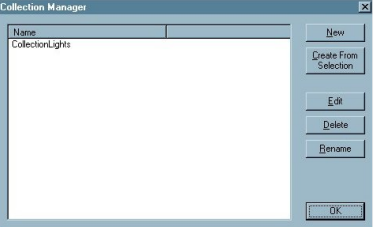
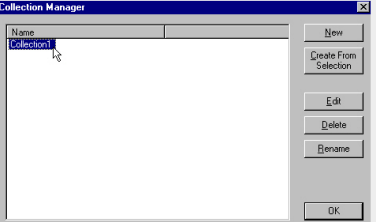
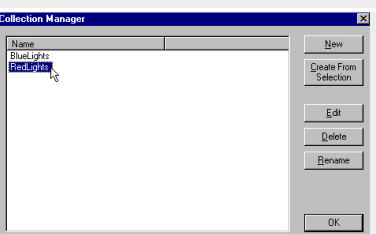
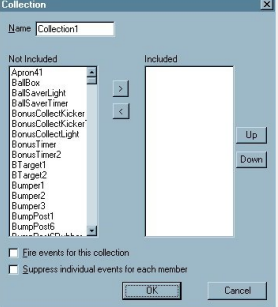
```
*****Light Stuff Starts Here  
  
'Gate_Hit sub will turn on the gatelight when hit  
  
Sub Gate_Hit() 'what happens when the gate is hit inside the sub  
GateLight.State = 2 '2 = LightStateBlinking This Starts Light Blinking when gate is hit.  
End Sub  
  
'Bumper1_Hit sub starts Light1. blinking with a blink pattern  
  
Sub Bumper1_Hit() 'what happens when the bumper is hit inside the sub  
PlaySound "Bumper"  
Light1.State=2 '2 is same as LightStateBlinking this starts Light1 blinking state when bumper is hit.  
Light1.BlinkPattern =1011110 'pattern for blinking lights on,off,on,on,on,on,off  
Light1.BlinkInterval=250 'Timer for lights on & off 250 = 1/4 of a second.  
End Sub  
  
'Bumper2_Hit sub starts Bumper2.blinking with a blink pattern  
  
Sub Bumper2_Hit() 'what happens when the bumper is hit inside the sub  
PlaySound "Bumper"  
Bumper2.State = LightStateBlinking 'or 2  
End Sub  
*****Light Stuff Ends Here
```

[zum Index](#)

[Nächste Seite](#)

Editor Objekte

Wie man Lichter Gruppen mit dem Collection Manager erstellt

	Lichter auswählen womit sie eine Gruppe erstellen wollen.
	Table/Collection Manager auswählen
	Klicke auf Create From Selection
	Erstellen sie eine Gruppe und nennen sie Collection1 N ew - erstellt ein neue Gruppe (Collection) C reate From Selection - Erstellt eine Gruppe aus den ausgewählten Objekten vom Tischeditor. E dit - Öffnet den Gruppeneeditor D elele - Löscht eine Gruppe R ename - Eine Gruppe umbenennen
	Es befinden sich schon 2 Gruppen in der Datei. Wählen sie RedLights und klicken auf E dit
<p>Das folgende Fenster öffnet sie wenn sie auf Edit drücken .</p> 	N ame - Collection1 Der Name der Gruppen, kann umbenannt werden, und ist somit im Script verwendbar. Not Included - Liste der Objekte die nicht in der Gruppe eingeschlossen sind. Included - Liste der Objekte die sich der Gruppe befinden. > - neues Objekt in die Gruppe einfügen <- Objekt aus der Gruppe entfernen. Up - Das ausgewählte Objekt in der Liste höher stellen. Down - Das ausgewählte Objekt in der Liste tiefer stellen. Fire Events for this collection - Fire Ereignisse?? Suppress individual events for each member - Zugriff auf einzelne Objekte in der Liste verwalten.

Hier ein Demo für ein Script dass den Zustand von Lichter und der Status *Light.State* verwendet.

Verwenden sie [DemoLights.zip](#)

Verwenden des Befehls "For Each" in einem Script.

Beispiel:

Dim light

```
For each light in RedLights  
light.State = LightStateOn  
Next
```

Die Variable 'light' wird von allen Licht Elemente der Gruppe verwendet unabhängig von der Größe. Diese Methode erlaubt Änderungen für eine komplette Gruppe.

In der Demo, löst die Taste *keycode* eine Routine aus die den Zustand der Lichter ändert.

```
If keycode = PlungerKey Then  
Plunger.Fire
```

In diesem Teil löst das nachlassen der Plungertaste ein Löschen der Lichter aus.

```
If keycode = LeftFlipperKey Then  
LeftFlipper.RotateToStart  
PlaySound "FlipperDown"
```

In diesem Teil löst die linke Flippertaste das Erlöschen der Lichter aus.

```
If keycode = RightFlipperKey Then
```

```
RightFlipper.RotateToStart
PlaySound "FlipperDown"
```

In diesem Teil löst die rechte Flippertaste das blinken der Lichter aus.

```
'Lichter Zustand 0 = LightStateOff (gelöscht)
' 1 = LightStateon (brennt)
' 2 = LightStateBlinking (blinkt)
```

```
***** Array
```

Verwendung einer Array Variable um den Lichter Zustand zu ändern

```
LightsArray = Array(Light1,Light2,Light3,Light4,Light5,Light6,Light7,Light8)
For a = 0 to 7 'a ist ein Wert zwischen 0 und 7 = 8 lights
LightsArray(a).State=2 'Licht blinkt
Next
```

```
***** Collections
```

' Green Lights & Red Lights & Blue Lights sind Lichtgruppen

' Um die einzelnen Elemente zu sehen, wählen sie

' Table / Collection Manager (Select a collection) / Edit

' Die Gruppe verwendet den Befehl "For Each"

For Each Light in GreenLights

Light.state = LightStateBlinking 'or 2

Next

For Each Light in Redlights

Light.State=2 'ou LightStateBlinking 'les 3 lumières rouges au-dessus des the flippers

Next

'Man kann auch Tabellenvariablen (Arrayvariablen) verwenden. "For Each" ist jedoch oft einfacher.

For a = 0 to 11 '0 bis 11 = 12 Lichter, 0 zählt als erste Zahl

BlueLights(a).State=2 ' 12 blaue Lichter blinken in Uhrzeigerichtung (Siehe Pattern Option für den Effekt)

Next

```
***** Setting Lights
```

'Diese Option braucht länger zu schreiben :)

Dim Red12Lights(11)

Set Red12Lights(0) = r1

Set Red12Lights(1) = r2

Set Red12Lights(2) = r3

Set Red12Lights(3) = r4

Set Red12Lights(4) = r5

Set Red12Lights(5) = r6

Set Red12Lights(6) = r7

Set Red12Lights(7) = r8

Set Red12Lights(8) = r9

Set Red12Lights(9) = r10

Set Red12Lights(10) = r11

Set Red12Lights(11) = r12

For a = 0 to 11 '0 bis 11 = 12 Lichter

Red12Lights(a).State=2 '12 rote Lichter blinken in Gegenuhrzeigerichtung (Siehe Pattern Option für den Effekt)

Next

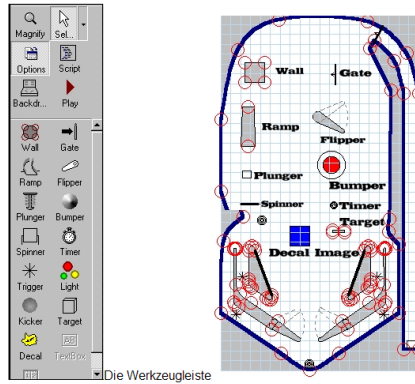
Achtung die Beispieldateien sind nicht übersetzt worden.

Notiz: Jede Methode hat vor und Nachteile es kommt auf die verwend drauf an. Die Letzt trotz mehr Schreibaufwand kann durch bessere Lesbarkeit des Scripts in aufwendigen und schwierigen Scripts die Mehrarbeit ausgleichen. Soll ein Licht einzel sowohl als Gruppe gesteuert werde ist sie die bessere Wahl.

[zum Index](#)

Editor Objekte

Gate & "Sound Events"



Die Werkzeugleiste

Gate

Gates sind eine Art Schleuse beim Flipper, sie lassen den Ball nur in eine Richtung durch.



Gate Schaltfläche

<p>File Edit Insert Table Preferences Window Help</p> <p>New Ctrl+N Open... Ctrl+O Close Save Save As... Ctrl+S Save As Protected... Export Unlock</p>			
<p>Zuerst setzen wir ein Gate: Falls noch kein Tisch geöffnet ist..... File / New</p>	<p>Die Gate-Schaltfläche anklicken</p>	<p>Den Mauszeiger über den Tisch bewegen. Er sollte jetzt so aussehen, klicken wo das Gate platziert werden soll.</p>	<p>So sieht das Gate im Editor aus.</p>

Jetzt können wir uns die Gate Optionen genauer ansehen.



Options imToolBar

aktivieren und das Gate durch anklicken auswählen.

Jetzt werden die Gate Eigenschaften sichtbar.

Gate - Art des ausgewählten Objektes.

Name - Gate1 Kann umbenannt werden. Wird mit diesem Namen im Script angesprochen um seine die Routinen auszuführen.

---- Color & Formatting

Color - Farbe des Gates auswählen.

---- Position

X - Zeigt die X Position des Gate auf dem Tisch.

Y - Zeigt die Y Position des Gate auf dem Tisch.

Length - 100 ist voreingestellt. Breite kann verändert werden.

Rotation - -90 ist voreingestellt. Der wert gibt den Winkel des Gate auf dem Tisch an (der Pfeil zeigt in die Richtung in die es sich öffnen kann)

Surface - Sélectionne la surface de l'objet sur laquelle est placer la grille. La surface par défaut est <None> ce qui correspond à la surface de la table.

Timer Enabled - non coché par défaut le timer est en position repos. On l'active en général dans le script. Permet l'appel à la procédure `GateName_Timer()` dans le script

Timer Interval - Définit le temp d'attente avant execution du contenu de Timer

UserValue - Variable utilisable librement dans le script. Il m'arrive de l'utiliser comme marqueur(flag)

Download [demo2light&bumperstates.zip](#)

Es folgt ein Beispiel Script um zu zeigen wie Sounds mit Events (Ereignissen) verknüpft werden.

Script ist zur Unterscheidung grün markiert

`*Gate_Hit sub will turn on the gatelight when hit`

```
Sub Gate_Hit() 'what happens when the gate is hit is inside the sub
PlaySound "gate22" 'plays this sound when Gate_Hit
GateLight.State = 2 '2 = LightStateBlinking This Starts Light Blinking when gate is hit
End Sub
```

Beachte das PlaySound in the PlungerKey_PullBack & Fire Hinweis: Alles hinter dem >' <Apostroph (oder "Rem" ist ein Kommentar des Script-Autors und wird vom Programm nicht beachtet.

`Sub Table1_KeyDown(ByVal keycode)`

`If keycode = PlungerKey Then`

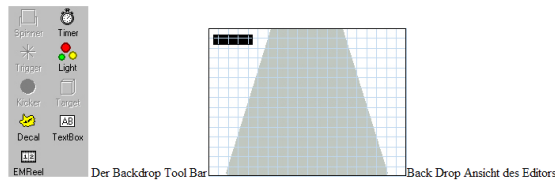
```
Plunger.PullBack  
PlaySound "rustyplunger"  
End If  
  
Sub Table1_KeyUp(ByVal keycode)  
  
If keycode = PlungerKey Then  
Plunger.Fire  
PlaySound "plungerSpring"  
End If
```

[zum Index](#)

[Nächste Seite](#)

Editor BackDrop Objekte

TextBox & Score Anzeige



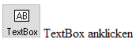
TextBox

TextBoxes benutzt man um Text anzuzeigen.



TextBox Schaltfläche

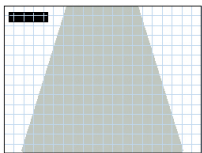
Eine TextBox erzeugen: Mit **File / New** einen neuen Tisch öffnen.



den Mauszeiger über den Tisch bewegen,

er sollte jetzt so aussehen, klicken wo eine TextBox platziert werden soll.

So sieht sie im Editor aus.



kann überall platziert werden, der graue Bereich stellt den Tisch dar.

Ein neuer Tisch enthält schon eine TextBox, die mit "ScoreText" benannt ist, dafür gibt es aber noch kein Script.

Zu den TextBox Optionen:

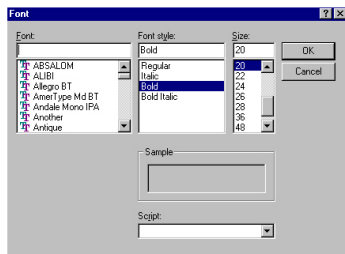


Options im **ToolBar** anklicken und die TextBox durch anklicken auswählen.

Jetzt können wir die TextBox Eigenschaften betrachten:

	<p>TextBox - Art des Objektes das ausgewählt ist</p> <p>Name - <i>ScoreText</i> Der Name der TextBox, die der neue Tisch schon enthält Tipp: Es ist sinnvoll der TextBox einen Namen zu geben, der etwas mit Ihrem Text zu tun hat, das spart später Verwirrung</p> <p>Text - Hier kann man Text eingeben. Dieser Text wird (erstmal) gezeigt, wenn der Tisch gespielt wird.</p> <p>Transparent - Macht die Back Color unsichtbar.</p> <p>Back Color - Stellt die Hintergrundfarbe ein</p> <p>Text Color - Stellt die Textfarbe ein</p> <p>Font - Auswählen der Schriftart, Schriftschnitt & Größe.</p> <p>Alignment - Richtet den Text mittig, nach links oder rechts aus.</p> <p>X - Zeigt die X Position der TextBox auf dem Tisch.</p> <p>Y - Zeigt die Y Position der TextBox auf dem Tisch.</p> <p>Width - Ändert die Breite der TextBox.</p> <p>Height - Ändert die Höhe der TextBox.</p> <p>Timer Interval - Eigener Timer, über <code>TextBoxName_Timer()</code> aktivieren .Ausgeschaltet ist voreingestellt.</p>
--	---

Anklicken von **Font** öffnet das Schriftart Menü. Hier kann man Schriftart, Schriftschnitt und Größe auswählen.



Im **Sample** Feld sieht man eine Vorschau.

Wenn der Flipper veröffentlicht werden soll, muß man bedenken, es können beim Anwender nur Schriften angezeigt werden, die dieser auch installiert hat. Wenn du also exotische Schriften verwendest, solltest du sie in "mit liefern", sie sind also **nicht** im Flipper enthalten!

Beispiel [demo3scoretextbox.zip](#)

Hier noch etwas Script für die Demo , um Score zu zählen und die Anzeige in der **TextBox**. Script ist zur Unterscheidung grün geschrieben.

Am Anfang des Skriptes steht folgende Zeile:

Dim Score 'Beachte das Sub AddScore, Diese Variable speichert den Wert der dort gegeben wird.

Alles hinter einem ' oder Rem ist eine Bemerkung des Autors und wird vom Script nicht mitgelesen.

Dim deklariert 'Score' als eine Variable. Das bedeutet sie ist eine Art Speicher geworden, sie kann z.B. einen Zahlenwert enthalten. Ein neuerliches AddScore ändert diesen Wert. (Ein Dim kann auch einen Zustand speichern, gewöhnlich 'true' oder 'false')

Hier noch mal das Sub um den Score hinzuzufügen.

```
***** Adding Score
```

Sub AddScore(points) 'dafür brauchen wir das Dim Score am Anfang (Variablen stehen üblicherweise am Anfang des Skriptes)

Score = Score + points ' Hier werden zum Score die Punkte addiert die vorher durch AddScore (#) gegeben wurden (in den Klammern stehen die Points)

ScoreText.Text = FormatNumber(Score, 0, -1, 0, -1) 'etwas für Fortgeschrittene, gibt das Format an, in dem die Ziffern dargestellt werden)

End Sub

```
***** Adding Score Ends here
```

Durchsuche mal das Script nach AddScore(50) oder jedes andere AddScore (#) Bumper, Targets & SlingShots usw. rufen jeweils das Sub AddScore auf und geben gleichzeitig die Punktzahl.

[Editor](#) - Grundlagen des Visual Pinball Editors. Menüleisten Funktionen - importieren von Images & Sounds

[Table Optionen - Ballfarbe](#) - Erklärt die Table Optionen, importieren eines Ball Images & verändern der Ballfarbe

[Rechts Klick Optionen](#) - Erklärt alle Rechts-Klick-Optionen für die versch. Objekte

[Erzeugen eines Sling Shot](#) - Zeigt wie man in einer Wall aus einer Linie einen Sling Shot macht

[Decals](#) - Erklärt die Decal Optionen, & zeigt wie ein Image importiert wird. Pinball DemoDecal

[Phunger & Kicker](#) - Wie man Phunger & Kickers verwendet & ein Beispiel

[Walls & Targets](#) - Alles über Walls & Targets

[Lights & Bumpers](#) - Light & Bumper States & Lights, Bumper Option Leiste

[Gate & Sound Events](#) - Erklärt die Gate Optionen & wie Sounds über das Script gestartet werden. Pinball Demo2

TextBox & Making Score count - Zeigt wie man TextBoxen verwendet & AddScore & Dim Score in Script. Pinball Demo3

[Ramps](#) - Erläut Ramp Optionen & und wie man eine einfache Rampe baut. Pinball Demo4

[Spinners](#) - Erläut Spinner Optionen & Script für einen Spinner. Pinball Demo4

[Triggers](#) - Erläut Trigger Optionen & Script für einen Trigger. Pinball Demo5

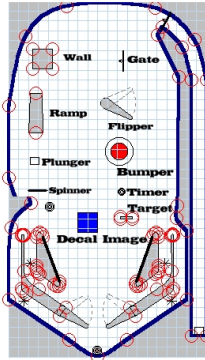
[Timers](#) - Erläut Timer Optionen & Script für einen Timer. Pinball Demo6

[Flippers](#) - Erläut Flipper Optionen & Script für einen Flipper. Pinball Demo7

[Zurück zur Hauptseite](#)

Editor Objekte

Ramp



Ramp

Ramps -Rampen, um den Ball auf ein höheres Level zu bringen.



Ramp Schaltfläche

File Edit Insert Table Preferences Window Help New Ctrl+N Open... Ctrl+O Close Save Save As... Ctrl+S Save As Protected... Export Undo			
Eine Rampe erzeugen: Mit File / New einen neuen Tisch öffnen.	Die Ramp Schaltfläche anklicken	den Mauszeiger über den Tisch bewegen, er sollte jetzt so aussehen, klicken wo die Rampe platziert werden soll.	So sieht sie im Editor aus.

Betrachten wir uns nun die Optionen der Rampe.

Die Optionen im ToolBar anschalten	und die Rampe durch anklicken auswählen

Das zeigt und die Eigenschaften der Rampe.

Ramp - Art des Objektes, das ausgewählt ist.

Name - *Ramp1* Der Name der Rampe, kann umbenannt werden, macht aber eigentlich keinen Sinn, weil sie eh nicht im Script auftaucht.

-----Colors & Formatting

Type - Rampe Type zur Auswahl RampTypeFlat, RampType4Wire, RampType2Wire, RampType3WireLeft, RampType3WireRight.Left. Erklärt sich durch das Bild unten.

Image - Hier ein Bild für die Rampe auswählen je nach Modus

Mode - Im Modus **ImageModeWorld** ist das Tischbild die Vorlage und nicht die Rampengröße. Standard des Tischbild ist 512 X 1024 Pixel. Im Modus **ImageModeWrap** - Wird das Bild auf die Rampe aufgezogen. Das Bild sollte der Rampengröße entsprechen.

Apply Image to Walls - Zeigt das Rampenbild im Editor an (im Type RampTypeFlat)

Color - Die Farbe der Rampe verändern.

Cast Shadow Zeigt den Schatten der Rampe an.

-----position

Top Height - Stellt die maximal Höhe der Rampe ein.Voreinstellung ist 100.

Bottom Height - Stellt die Basis der Rampe ein.Voreinstellung ist 0.

Top Width - Stellt die obere Weite der Rampe ein.Voreinstellung ist 60.

Bottom Width - Stellt die untere Weite der Rampe ein.Voreinstellung ist 75.

Visible Wall -Zeigt die Angegeben Seitenhöhe der Rampe an.

Left Wall - Sichtbare Höhe der Linken wand

Right Wall - Sichtbare Höhe der Rechten Wand

-----Physics

Physical Wall -Höhe der realen Wand.
Notiz: Manchmal muss die reale Wand höher stelle damit de Ball nicht runterfährt aber Optisch setzt man Visible Wall tiefer damit es besser aussieht.

Left Wall - Reale Höhe der Linken Wand

Right Wall - Reale Höhe der Rechten Wand

-----misc **wird normalerweise nicht benützt. Könnte wie ein gewöhnlicher Timer benützt werde bringt aber kein Sinn.**

Timer Enabled - Standart nicht aktiv. Der Time ist in Ruhrsellung. Er wird normalerweise im Script eingeschaltet, und wird mit *RampName_Timer()* angesprochen.

Timer Interval - Zeitintervall bis der Timer anspringt.

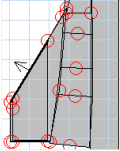
Flat, 4Wire, 2Wire.

UserValue -Frei zu Verfügung stehende Variable. Braucht keine Dim Deklaration. Muss jedoch vor Gebrauch initialisiert werden. Die Variable kann den Type Zeichenkette, Numerisch, Boolean haben.
 Notiz: Ich brauche diese Variable ab und zu als Markierer.

... Mit **File/Export/Blueprint...** exportiert man Eine Blaupause des Tisches. Diese eignet sich hervorragend als Grundlage zur Gestaltung des Tisch -Bildes (Als Faustregel kann man sagen, das Bild sollte etwa halb so groß (in Pixel) wie die Einheiten der Tisch Höhe und Breite sein). Per **Table/Image Manager...** importieren Maximale Bildgröße 1024X1024.

Rampen zu Bauen ist manchmal nicht ganz einfach. Es passiert oft, das der Ball seitlich herausfällt, deshalb ist es nützlich dort Walls hinzustellen. Der Ball kann auch von hinten unter die Rampe rollen und dort steckenbleiben, auch da hilft eine kleine Wall (es reicht schon eine Höhe von 25).


Beachte [demo4rampspinner.zip](#)



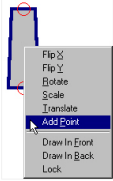
Im Beispiel sieht man solche "Hilfswände".

Direkt neben der Rampe ist ein Tunnel gebaut. Beachte einmal die verschiedenen Höhen in den Optionen für die Walls. Die Rampe selbst, besteht in Wirklichkeit aus Zweien, auch hier lohnt sich ein Blick in die Optionen!


Der Bau einer einfachen Rampe




Eine Rampe erzeugen, den Mauszeiger zu einem Punkt bewegen, wo ein neuer Punkt gesetzt werden soll.




Rechts klicken & und Add a Point auswählen.




Der neue Punkt ist hinzugefügt.




Noch ein paar hinzugefügt....




Mit Hilfe der Punkte, kann man eine Kurve bauen.



Jetzt jeweils einen Punkt auswählen,



Rechts klicken und smooth auswählen. Ist auch sichtbar im rechten Optionen Fenster wenn es aktiv ist.



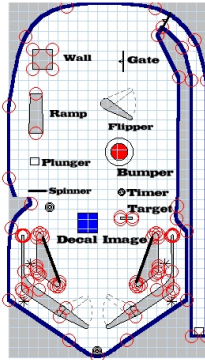
Das erst macht die Kurve richtig rund. Option **Slingshot** bringt wieder die Kanten zum vorschein.

[zum Index](#)

[Nächste Seite](#)

Editor Objekte

Spinner



Spinner

Spinnersdrehen sich wenn der Ball durch sie hindurchrollt.



Spinner Schaltfläche

File Edit Insert Table Preferences Window Help			
New	Ctrl+N		
Open...	Ctrl+O		
Close			
Save	Ctrl+S		
Save As...			
Save As Projected...			
Export			
Undo			

Einen Spinner erzeugen: Mit **File / New** einen neuen Tisch öffnen

Auf die Spinner Schaltfläche klicken

Den Mauszeiger über den Tisch bewegen.

Der Spinner im Editor.

Betrachten wir die Spinner Optionen.

Die Optionen im ToolBar	anklicken und den Spinner durch anklicken auswählen

Jetzt können wir die Spinner Eigenschaften betrachten.

Spinner - Art des Objektes, das ausgewählt ist.

Name - *Spinner1* Name des Spinners, kann umbenannt werden. **Tipp:** Wenn der **Spinner** umbenannt wird, darauf achten, daß auch in Script alles stimmt.

-----Color & Formatting

Front Image - Bild für Vorderseite des Spinners wählen.

Back Image - Bild für Hinterseite des Spinners wählen.

Color - Farbe des Spinners ändern.

Cast Shadow Schatten des Spinners anzeigen.

-----Position

X - Zeigt die X Position des Spinners auf dem Tisch.

Y - Zeigt die Y Position des Spinners auf dem Tisch.

Length - Verändert die Breite des **Spinners**. Voreinstellung 80

Height - Verändert die Höhe des **Spinners**. Voreinstellung 60

Rotation - Stellt den **Spinner** schräg.

Overhang - Verlängert den Draht der den **Spinner** hält.

Surface - Surface meint die Höhe eines anderen Objektes, die der Plunger annehmen soll. Voreingestellt ist <None> das heißt auf Höhe des Tisches. Zur Erinnerung: Der Plunger braucht Walls, die den Ball am Platz halten.

-----Physics

Friction - Stellt ein, wie leicht der Spinner rotiert (höhere Zahl, schwächeres Rotieren)

-----Misc

Timer Enabled - Standart nicht aktiv. Der Time ist in Ruhrsellung. Er wird normalerweise im Script eingeschaltet, und wird mit *SpinnerName_Timer()* angesprochen.

Timer Interval - Zeitintervall bis der Timer anspringt.

UserValue - Frei zu Verfügung stehende Variable. Braucht keine Dim Deklaration. Muss jedoch vor Gebrauch initialisiert werden. Die Variable kann den Type Zeichenkette, Numerisch, Boolean haben.

Notiz: Ich brauche diese Variable ab und zu als Markierer.

Note: La grandeur usuel pour une image applicable au spinner est de 300x300 pixels.

Spinner
Name <input type="text" value="Spinner1"/>
Color & Formatting
Front Image <input type="text"/>
Back Image <input type="text"/>
Color <input type="color" value="#00FF00"/>
<input checked="" type="checkbox"/> Casts Shadow
Position
X <input type="text" value="353"/>
Y <input type="text" value="581"/>
Length <input type="text" value="80"/>
Height <input type="text" value="60"/>
Rotation <input type="text" value="0"/>
Overhang <input type="text" value="10"/>
Surface <input type="text"/>
Physics
Friction <input type="text" value="0.999999"/>
Misc
<input type="checkbox"/> Timer Enabled
Timer Interval <input type="text" value="100"/>
User Value <input type="text"/>

Spinner Script

Beispiel [demo4rampspinner.zip](#)

Spinner sind einfach zu scripten. Der entsprechende Teil findet sich im Script ganz unten.

Achtung: benutze Spinner1_Spin() *nicht* Spinner1_Hit()

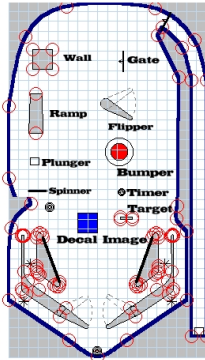
```
***** Spinner1
Sub Spinner1_Spin() Inside this Sub is what the spinner1 will do
PlaySound "swish"
AddScore (5)
End Sub
***** Spinner1 Ends here
```

[zum Index](#)

[Nächste Seite](#)

Editor Objekte

Trigger



Trigger

Trigger sind eigentlich nichts anderes als Schalter, sie lösen irgend ein Hit Event aus.



Trigger Schaltfläche

File	Edit	Insert	Table	Preferences	Window	Help
New						Ctrl+N
Open...						Ctrl+O
Close						
Save						Ctrl+S
Save As...						
Save As Protected...						
Export						
Unlink						

		
Einen Trigger erzeugen: Einen neuen Tisch öffnen mit File / New .	Auf die Trigger Schaltfläche klicken.	Bewege den Mauszeiger über den Tisch, er sollte jetzt so aussehen, klicken, wo der Trigger platziert werden soll.
		So sieht er im Editor aus.

Wir betrachten uns die Trigger Optionen:

	
Options in der ToolBar	aktivieren und den Trigger auswählen

Jetzt sehen wir die Trigger Eigenschaften,

Hinweis: Jeder neue Tisch enthält bereits die 4 klassischen Trigger: LeftOutlane, RightOutlane, LeftInlane, RightInlane -es gibt für sie aber noch kein Skript.

Trigger

Name:

Colors & Formatting

Shape:

Visible

Radius:

Position

X:

Y:

Surface:

State

Enabled

Misc

Timer Enabled



Timer Interval:

User Value:

Trigger- Art des Objektes das ausgewählt ist.

Name - *Trigger1* Name des Triggers, kann geändert werden.

-----Colors & Formatting

Shape - ShapeCircle  & ShapeCustom  - Custom läßt sich ähnlich wie eine Wall verändern.

Visible - Sichtbar oder unsichtbar einstellen.

Radius - Stellt den Radius des Triggers ein. 25 ist voreingestellt.

-----Position

X - Zeigt die X Position des Triggers auf dem Tisch an.

Y - Zeigt die Y Position des Triggers auf dem Tisch an.

Surface - Surface meint die Höhe eines anderen Objektes, die der Plunger annehmen soll. Voreingestellt ist <None> das heißt auf Höhe des Tisches. Zur Erinnerung: Der Plunger braucht Walls, die den Ball am Platz halten.

-----State

Enabled - Wenn aktiviert, funktioniert der Trigger immer, oder er wird erst über das Script aktiviert.

-----Misc

Timer Enabled - Standart nicht aktiv. Der Time ist in Ruhrsellung. Er wird normalerweise im Script eingeschaltet, und wird mit *SpinnerName_Timer()* angesprochen.

Timer Interval - Zeitintervall bis der Timer anspringt.

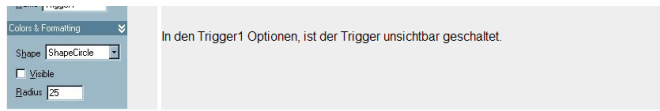
User Value - Frei zu verfügung stehende Variable. Braucht keine Dim Deklaration. Muss jedoch vor Gebrauch initialisiert werden. Die Variable kann den Type Zeichenkette, Numerisch, Boolean haben.
Notiz: Ich brauche diese Variable ab und zu als Markierer.

Trigger Skript

Beispiel [demo5triggersinlaneoutlane.zip](#)

Trigger sind auch einfach zu skripten, am Ende des Beispiel Scriptes findest Du den Teil für die Trigger.

Trigger
Name: <input type="text" value="Trigger1"/>



Beachte: LeftOutline, RightOutline, LeftInlane, RightInlane haben jetzt auch "Events"

```
***** Triggers

Sub Trigger1_Hit() "Trigger1 wird von der Kugel überrollt, dann:
Playsound "LaunchBall"
End Sub

Sub RightInlane_Hit
AddScore (10)
Playsound "swoosh"
End Sub

Sub RightOutline_Hit
AddScore (10)
Playsound "swoosh"
End Sub

Sub LeftInlane_Hit
AddScore (10)
Playsound "swoosh"
End Sub

Sub LeftOutline_Hit
AddScore (10)
Playsound "swoosh"
End Sub

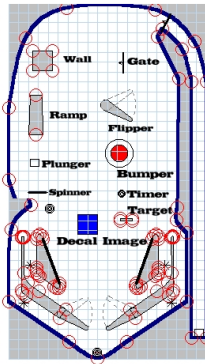
***** Triggers Ends here
```

[zum Index](#)

[Nächste Seite](#)

Editor Objekte

Timer



Timer

Timer werden benutzt um nach Ablauf einer Zeitspanne, irgend etwas auszulösen.

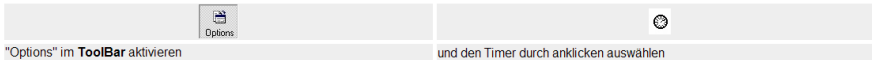


Timer Schaltfläche

Einen Timer erzeugen: Mit File / New einen neuen Tisch öffnen	Auf die Timer Schaltfläche klicken.	er sollte jetzt so aussehen, klicken, wo der Timer platziert werden soll. (es ist im Prinzip egal, wo der Timer hinkommt, manchmal setzt man in zum dazugehörigen Objekt oder man macht eine "Timer Sammlung", Geschmacksache)	so wird er dann angezeigt.

Notiz: Die Timer sind übersichtlicher wenn sie sie auf der Seite de Tisch aufstelle um Mit Aussagekräftig Namen versehen. Man kann den Namen auch auf die Rückwand legen wenn man ihn für Anzeige Manipuéationen verwendet

Jetzt betrachten wir uns die Timer Optionen:



"Options" im **ToolBar** aktivieren

und den Timer durch anklicken auswählen

Die Timer Eigenschaften werden angezeigt

	<p>Timer - Art des Objektes das ausgewählt wurde.</p> <p>Name - <i>Timer1</i> Der Name des Timers, kann umbenannt werden.</p> <p>Timer Enabled - (Befähigt) ist voreingestellt, d.h. er läuft immer. Oft wird das ausgeschaltet und der Timer erst im Script aktiviert.</p> <p>Timer Interval - Die Zeitspanne bevor des Timers das Script ausführt. Voreingestellt ist 100. (1000 = 1 Sec.)</p> <p>User Value - Variable utilisable librement dans le script. Il m'arrive de l'utiliser comme marqueur(flag)</p> <p>Note: Le Timer peut être placé sur la Table ou le panneau d'affichage.</p>
--	--

Timer Skript

Beispiel [demo6timer.zip](#)

Timer sind einfach zu aktivieren. Im Beispiel Tisch kannst du beobachten, wie Timer arbeiten. Das Skript für die Timer ist am Ende des Skriptes.

Hinweis: In den Timer Optionen ist Enabled ausgeschaltet, Aktivierung erfolgt erst, wenn beide Targets "droppen".

Im Sub CheckTarget wird der Timer1 auf "enabled" gesetzt.

```
Sub CheckTarget()  
If Target.IsDropped = True And Target1.IsDropped = True Then 'also nur wenn beide Targets versenkt sind, dann  
Timer1.Enabled = True 'wird der Timer aktiviert  
End If  
End Sub  
*****soweit zu den Targets
```

Im Sub Timer1_Timer werden der Timer & die Targets zurückgesetzt, der Timer bleibt deaktiviert, bis wieder beide Targets getroffen werden.

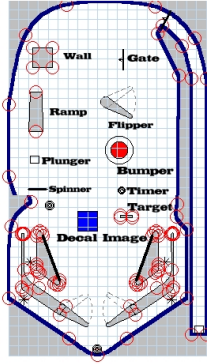
```
Sub Timer1_Timer()  
Timer1.Enabled = False 'nach Ablauf des Intervalls schaltet der Timer sich selbst aus  
PlaySound "targetup"  
Target.IsDropped = False 'Das Target geht hoch  
Target1.IsDropped = False 'Das Target geht hoch  
End Sub
```

[zum Index](#)

[Nächste Seite](#)

Editor Objekte

Flipper



Flipper

Flipper heißen eigentlich nur die Teile, die den Ball wegschlagen.



Flipper Schaltfläche

<p>File Edit Insert Table Preferences Window Help</p> <p>New Ctrl+N Open... Ctrl+O Close Save Ctrl+S Save As... Save As Projected... Export Undo</p>			
<p>Einen Flipper erzeugen: Zuerst mit File / New einen neuen Tisch öffnen.</p>	<p>Die Flipper Schaltfläche anklicken</p>	<p>Bewege den Mauszeiger über den Tisch Er sollte jetzt so aussehen, klicken wo der Flipper platziert werden soll.</p>	<p>Der Flipper im Editor.</p>

Notiz: Der Flipper ist nicht auf der richtigen Seite? Das werden wir später lösen.

Jetzt können wir die Flipper Optionen betrachten.

<p>Optionen im ToolBar</p>	<p>aktivieren und den Flipper durch anklicken auswählen.</p>

Bemerkung: Jeder neue Tisch ist grundsätzlich mit zwei Flippern inclusive script ausgestattet: LeftFlipper & RightFlipper

Flipper - Art des Objektes, das ausgewählt ist.

<p>Flipper</p> <p>Name: Flipper1</p> <p>Color & Formatting</p> <p>Color: []</p> <p>Rubber Color: []</p> <p>Rubber Thickness: 0</p> <p><input checked="" type="checkbox"/> Visible</p> <p>Position</p> <p>X: 381 Y: 457</p> <p>Base Radius: 26.73</p> <p>End Radius: 10.69</p> <p>Length: 142.57</p> <p>Start Angle: 120</p> <p>End Angle: 60</p> <p>Surface: []</p> <p>Physics</p> <p>Speed: 0.05</p> <p>Strength: 6</p> <p>Elasticity: 0.3</p> <p>Misc</p> <p><input type="checkbox"/> Timer Enabled</p> <p>Timer Interval: 100</p> <p>User Value: []</p>	<p>Name - Flipper1 Kann umbenannt werden, das muß aber im Script geändert werden, für jeden weiteren Flipper braucht es auch ein entsprechendes Sub im Script.</p> <p>----- Color & formatting</p> <p>Color - Farbe des Flippers auswählen.</p> <p>Rubber Color - Farbe des Gummis auswählen.</p> <p>Rubber Thickness - Stellt die Dicke des Gummis ein. 0 ist voreingestellt (der Gummi ist nicht sichtbar mit 0). Unterschied ist zu sehen im Bild unten.</p> <p>Visible - Désaktiviert den Flipper und macht ihn unsichtbar (damit kann man aber kein durchsichtigen Flipper machen.)</p> <p>----- Position</p> <p>X - Zeigt die X Position des Flippers auf dem Tisch.</p> <p>Y - Zeigt die Y Position des Flippers auf dem Tisch.</p> <p>Base Radius - Durchmesser am Anfang des Flippers. (normalerweise größer).</p> <p>End Radius - Durchmesser am Ende des Flippers. (normalerweise kleiner).</p> <p>Length - Länge des Flippers.</p> <p>Start Angle - Startposition des Flippers. Siehe Angles unten.</p> <p>End Angle - Endposition des Flippers (Flipper Taste gedrückt). Siehe Angles unten.</p> <p>Surface - Oberfläche worauf der Flipper liegen soll. Der Standard Wert <None> entspricht der Tischoberfläche.</p> <p>----- Physics</p> <p>Speed - Geschwindigkeit der Bewegung</p> <p>Strength - Stellt ein wie hart der Ball geschlagen wird.</p> <p>Elasticity - Höherer Wert lässt den Ball starker zurückprallen, macht also den Gummi elastischer.</p> <p>----- Misc</p> <p>Timer Enabled - Standard nicht aktiv. Der Timer ist in Ruhsellung. Er wird normalerweise im Script eingeschaltet, und wird mit <code>FlipperName_Timer()</code> angesprochen.</p> <p>Timer Interval - Zeitintervall bis der Timer anspringt.</p> <p>User Value - Frei zu verfügung stehende Variable. Braucht keine Dim Deklaration. Muss jedoch vor Gebrauch initialisiert werden. Die Variable kann den Type Zeichenkette, Numerisch, Boolean haben.</p> <p>Notiz: Ich brauche diese Variable ab und zu als Markierer.</p>
---	---

Links Wert 7

Rechts Wert 1

Angles

0 entspricht Norden, oben oder 12 Uhr, wie man es sehen will....

Die Winkel gehen von 0 bis 359° (es geht auch mit negativen Werten).

Das Ganze kann man sich auch so vorstellen:

```
..... 0
.....305.....45
.....270.....90
.....225.....135
.....180
```

Tip! Eine sehr einfache Methode um den Flipper auf andere Seite zu legen ist einfach den positiven Wert in ein negativen Wert zu ändern.

Start Angle 120 wird Start Angle -120

End Angle 60 wird End Angle -60

Flipper Script

Zum [demo7flipper.zip](#)

Flipper sind leicht zu aktivieren. Das unten gezeigte Script bezieht sich auf Flipper1 in dem Beispiel. Du findest es in den KeyCodes gleich am Anfang des Scriptes.

Beachten: Script aus der demo7 grün geschrieben, nur das Flipper1 Script ist hinzugefügt.

```
Sub Table1_KeyDown(ByVal keycode)
```

```
If keycode = LeftFlipperKey Then
```

```
LeftFlipper.RotateToEnd
```

```
Flipper1.RotateToEnd 'Bewegt den Flipper zu seiner Endposition
```

```
PlaySound "FlipperUp"
```

```
End If
```

```
Sub Table1_KeyUp(ByVal keycode)
```

```
If keycode = LeftFlipperKey Then
```

```
LeftFlipper.RotateToStart
```

```
Flipper1.RotateToStart 'Bewegt den Flipper zu seiner Startposition
```

```
PlaySound "FlipperDown"
```

```
End If
```

Flipper

Name

Color

Rubber Color

Rubber Thickness

95.5 567

Base Radius

End Radius

Length

Start Angle

End Angle

Speed

Strength

Elasticity


Visible

Surface

Timer Interval

Hier die Flipper Optionen für Flipper1

Beachte die Winkel (angle) um in in die richtige Position zu bringen, evt. ein wenig herumprobieren, um den Flipper deinem Tisch anzupassen.



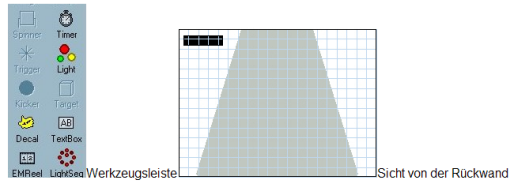
Die Wall wurde dem Flipper angepasst. Rechtsklicken und Punkte hinzufügen um die Wall zu modellieren. (die Punkte fürRundungen durch rechtsklicken auf smooth setzen)

[zum Index](#)

[Nächste Seite](#)

Editor Rückwand Objekte

EMReel



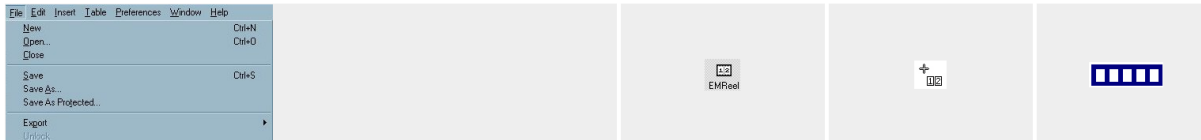
Notiz: Es war Black der die Option EMReel für VP entwickelt hat. Die Beschreibung kommt von einer Homepage. Sie können dort auch noch mehr Infos bekommen.

EmReel

Die EmReels sind Simulationen von Anzeigerollen wie man sie auf alten Flipperkästen kenn.



EmReel Schaltfläche

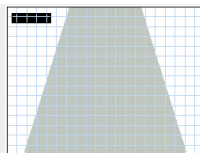


Einen EMreel erzeugen: Einen neuen Tisch öffnen mit **File / New** .

Auf die Wall Schaltfläche klicken.

Der Mauszeiger verwandelt sich in EMReel Mauszeiger. Bewegen sie den Mauszeiger über den Tisch, wo sie den EMReel erzeugen wollen

Sie haben ein EMReel erstellt.



Notiz: Setzen sie das Objekt nicht unbedingt in die graue Zone des Tisch.

Wir betrachten jetzt die EMReel Optionen:



Options



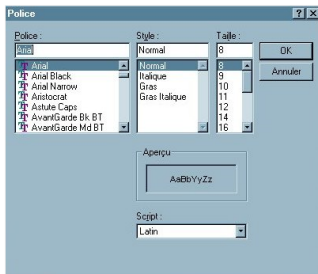
ENReel mit der Maus auswählen

Options in **ToolBar** aktivieren

Jetzt sehen wir die EMReel Eigenschaften.

EMReel	EMReel -Art des Objektes das ausgewählt ist.
Name: EMReel1	-----Colors and formatting
Colors & Formatting	Name - <i>EMReel1</i> der Name kann geändert werden. Die EMReel wird mit dem Name im Script adressiert.
Type: ReelText	Type - Sie können ReelText oder ReelImage wählen. ReelText wird von Font aus geändert. ReelImage verwendet eine Bilderserie in Auswahlfenster, das zuvor geladen wurde. (Mehrere Bilder sind auf einer oder mehreren Linien vereinigt.)
<input type="checkbox"/> Background Transparent	Background Transparent - Die Option aktiv lässt den Hintergrund durchsichtig erscheinen
Back Color: [Color Picker]	Back Color - Öffnet das Farbwahlfenster für den Rand des Hintergrunds.
Text Color: [Color Picker]	Text Color - Öffnet das Farbwahlfenster für die Textfarbe im Reeltext modus
Reel Color: [Color Picker]	Reel Color - Öffnet das Farbwahlfenster für die Farbe der Trommel(Reel)
Font: Times New Roman	Font - Auswahl für Textattribute.
Digit Range (0 -> 9)	Digit Range (0->) (Anzahl Bilder auf der Trommel ohne die 0) Standard 9 für die Zahlen 0 bis 9 Maximal 15
Images	Image - Auswahl der Bilderserie in der Liste.
<input type="checkbox"/> Use Image Grid	Use Image grid Standard nicht aktiv, aktiv wenn mehrer Linien auf einem Bild
Images Per Row: 1	Images per row Anzahl Bilderlinien auf dem Bild
<input type="checkbox"/> Reel Shading	-----Position
Position	X - Position X von EMReel
X: 759	Y - Position Y von EMReel
Y: 100	Reels (Max 32) - Anzahl von Trommeln (reel). Standard 5 für eine Anzeigetafel bis 99999 Punkte. Es gibt eine Reset function im Script zur Rückstellung der Trommeln auf 00000.
Reels (Max 32): 5	Reel Width - Aendert die breite der EMReel .
Reel Width: 30	Reel Height - Aendert die Höhe der EMReel .
Reel Height: 40	Reel Spacing - Aendert den Zwischenraum von den EMReel
Reel Spacing: 4	Reel Shading - wird nicht verwendet
State	-----State
Motor Steps: 2	Motor Steps - Aendert die Geschwindigkeit der Trommelrotation. Ein hohen Wert ist Flüssiger aber auch Langsamer. . Verwenden sie nicht mehr als 5 sonst ist es zu langsam.
Update Interval: 50	Update Interval - Bildwiederholrate in Millisekunden
Sound: [Sound Picker]	Sound - Sound Auswahl für die bewegten Trommeln
Misc	-----Misc
<input type="checkbox"/> Timer Enabled	Timer Enabled - Standard nicht aktiv. Der Time ist in Ruhsellung. Er wird normalerweise im Script eingeschaltet, und wird mit <i>EMReelName_Timer()</i> angesprochen.
Timer Interval: 100	Timer Interval - Zeitintervall bis der Timer anspringt.
User Value: [Input Field]	UserValue -Frei zur verfügung stehende Variable. Braucht keine Dim Deklaration. Muss jedoch vor Gebrauch initialisiert werden. Die Variable kann den Type Zeichenkette, Numerisch, Boolean haben.

Notiz: Ich brauche diese Variable ab und zu als Markierer.



Font öffnet die Schriftauswahl.

Notiz: Die spezial Schriften für einige Tische müssen zuerst in der Schriften Verwaltung von Windows installiert werden.

Sample: Vorschau für den ausgewählten Schriftzug

Script: Brauchen sie nicht zu ändern.

Notiz: Hier hat Script nichts zu tun mit dem Script Editor.

Verwenden Sie [demo8emreel.zip](#)

Hier die Basis Aenderungen die mit Demo8.

Man ersetze den Befehl Addscore durch EMReel1.addvalue(#) um das Resultat zu berechnen. Demo8 besitzt auch ein EMReel2 um die Funktion der Grafischen Trommeln vor zu führen.

Beispiel:

AddScore (10) 'Fügt 10 Punkte zu Resultat (score) zu'
EMReel1.addvalue(10) 'Das gleiche mit EmReel'

Wie kann man die EmReel im Script verwenden?

Es gibt 4 Methoden um EMReel zu verwenden. Das sind:

ResetToZero() 'auf null setzen

AddValue(Value) 'value zum Resultat zufügen

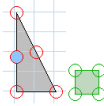
SetValue (Value) ' mit dem Wert Value initialisieren

SpinReel (ReelNumber, PulseCount)

Siehe Black's Seiten

[Zum Index](#)

Kontrolpunkte bearbeiten

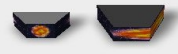


Ob sie Wall, Target, Lights oder Trigger ändern, wird ein Kontrollpunkt immer blau wenn er gewählt ist.

Options	Option muss activ sein
Control Point Colors & Formatting <input type="checkbox"/> Smooth Position X: 735 Y: 1015	Light und Trigger in freien Form modus sowie die Rampe haben diese Optionen angezeigt wenn man sie mit der Maus auswählt. Smooth diese Option ist im standart nicht activ. Sie erlaubt ein ausgewählten Punkt zu glätten. X - X Coordinat des Punkt auf dem Tisch Y - Y Coordinat des Punkt auf dem Tisch
Smooth Slingshot	Dieses Fenster erscheint neben den Punkt wenn man mit der rechten Taste drückt. Für schnellere Bedienung Smooth : Glättet den gewählten Punkt. Slingshot : Bringt die Kante wieder zurück. Notiz: Verwendet man ein Wall oder Target entsteht links von gewählten Punkt ein Gummiband (Slingshot). Soll jedoch nur die Kante wieder hergestellt werden, muss man nochmal die Option wählen um den Hacken der Option zu entfernen.
Control Point Colors & Formatting <input type="checkbox"/> Smooth <input checked="" type="checkbox"/> Auto Texture Coord. Texture Coord: 0 Position X: 663 Y: 655	Für Wall und Target werden diese Optionen angezeigt, wenn man ein Kontrollpunkt auswählt. Smooth diese Option ist im standart nicht activ. Sie erlaubt ein ausgewählten Punkt zu glätten. Auto Texture Coord. Die Option ist beim erstellen eines neuen Punkt activ und bestimmt die Anwendung der Seitenbilder. Texture Coord. Kann de Wert 1 oder = annehmen X - X Coordinat des Punkt auf dem Tisch Y - Y Coordinat des Punkt auf dem Tisch

Wie man Seitenwände bekleidet.

Die Vorgehensweise ist mit einem Beispiel leichter zu verstehen. Zuerstes kann nur ein Bild für eine Mauer oder Zielscheibe verwendet werden. Die Optionen **Auto Texture Coord.** und **Texture Coord.** von jedem Punkt sind verschieden Angelegt ob es eine Mauer (Wall) oder Zielscheibe (Target) ist. Ich habe hier die neu erstellten Objekte verformt und mit ein SeitenBild bekleidet ohne die Parameter der Punkte zu verändern. Man sie 3 Seiten die vierte kann man sich leicht vorstellen.



Target->Wall

Auf dem Objekt Target: wird das Bild einmal auf der Vorderseit geklebt und einmal auf die 3 restlichen Seiten überzogen.

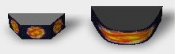
Auf dem Objekt Wall überzieht ein einziges Bild alle 4 Wände der Startpunkt kann nicht gewählt werden.

AT=AutoTextureCoord. TC=Texture Coord. On=aktiv Off=nicht aktiv

Hier die Parameter der einzelnen Punkten nach dem Erstellen neuer Objekte.

Target		Wall	
AT=on TC= 0	AT=on TC= 0	AT=on TC= 0	AT=on TC= 0
AT=off TC= 0	AT=off TC= 1	AT=on TC= 0	AT=on TC= 0

Geänderte Objekte



Target		Wall (2 points smooth)	
AT=off TC= 1	AT=off TC= 0	AT=off TC= 0	AT=off TC= 1
AT=off TC= 0	AT=off TC= 1	AT=on TC= 0	AT=on TC= 0

Erklärung:

1/ Ein Bild wird zwische ein Punktpaar gelegt das TC=1 und hat und mit AT off ist. (AT=off TC=0 et AT=off TC=1)

2/ Um ein Bild über mehrere Punkte zu überziehen müssen die Punkte zwischen eine **Paar (TC=0 Anfang, TC=1 Ende AT=off)** alle die Parameter (AT=on TC=0) haben.