Visual Pinball
Visual Guide

Original version by Cold1
PDF version by forchia

This is the beginning of the graphic tutorial for Visual Pinball by cold1
"Demo's" Include work along Table where noted!

Editor - Learn the Basics of the Visual Pinball Editor. Menu Bar Functions - Importing Images & Sounds.

Table Options - Ball Color - This explains Table options, importing a ball image & Changing ball color.

Right Click Options - Explains All Right Clicking Options for All Objects.

Building a simple Sling Shot - Learn how to make a slingshot from scratch using a basic wall.

Decals - Explains Decal Options, & shows how to import an image for a decal. Pinball DemoDecal

Plungers & Kickers - How to use a Plungers & Kickers & a Simple **Demo**

Walls & Targets - Explains Walls & Targets & Pinball **Demo1**

Lights Collections - Learn how to use lights in a collection using collection manager. **Pinball DemoLights**

Lights & Bumpers - Learning Light & Bumper States & Lights, Bumpers Option Bar. Pinball **Demo2**

Gate & Sound Events - Explains Gate options & how to use sound events in keyup & keydown. Pinball **Demo2**

TextBox & Making Score count - Shows you how to use TextBoxes & AddScore & Dim Score. Pinball **Demo3**

Ramps - Explains Ramp Options & shows you how to build a simple Ramp. Pinball **Demo4**

Spinners - Explains Spinner Options & shows script to make a spinner work. Pinball **Demo4**

Triggers - Explains Trigger Options & shows script to make a Trigger work. Pinball **Demo5**

Timers - Explains Timer Options & shows script to make a Timer work. Pinball **Demo6**

Flippers - Explains Flipper Options & shows script to make a Flipper work. **Pinball Demo7**

EMReel - Shows you how to use EMReels. **Pinball Demo8**

Appendix

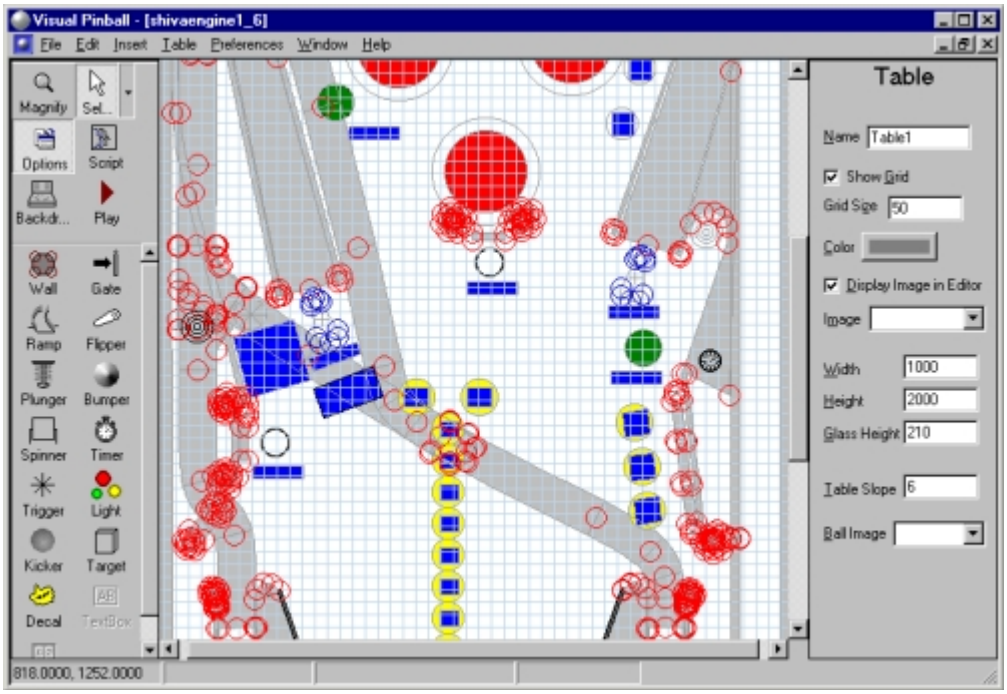
A: Keycodes - Additional Keycodes reference.

B: Faq - A somewhat arranged faq on visual pinball copy & pasted together from Shiva's VP forums by Cold1

C: Tables List - A list of table up to 1-27-02 some may be doubles because of newer version or VP/VPM versions

The Visual Pinball Editor

The **Editor** may look a little intimidating at first, but we will cover all the basics to get you well on your way to creating your very own pinball table. The Editor is the main part of building a table. From here we add our walls, lights, bumpers, targets, and all the rest of the objects that are part of a pinball table.



Note. In the picture to the right, the **Options** box is selected.
With **Options** selected you gain the access to the properties of selected items.
You can see the **Options** to the far right of the screen. In this Picture, the **Table** is selected.

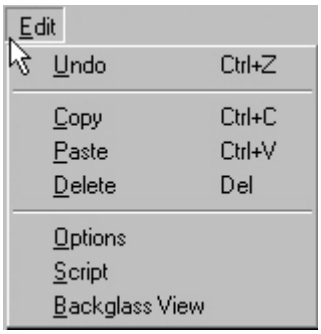
First off, lets get the flyout menu bar out of the way, this will show us how to load/save our tables and lots more.

The very top (Dark Blue) is the name of the table that we have loaded into the VP Editor.
First we'll look at the **File** options

File/

New – Creates a new basic table
Open – select a table you want to open
Close – Closes the selected table
Save – Saves your table that's opened
Save As... - saves your table with a name change
Export – Exports a blue Print of your table for editing in a drawing program
Exit - Exits Visual Pinball or (VP) for short

Next is the **Edit** options **Edit/**



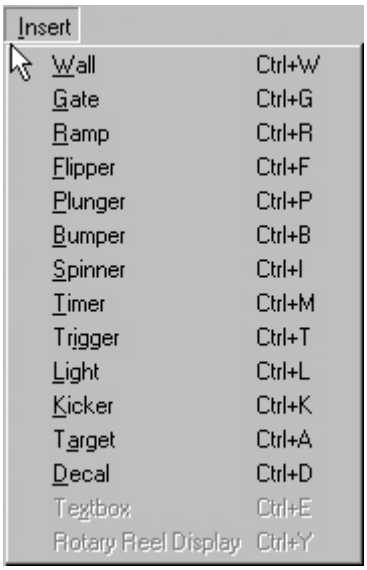
Undo – undoes the last thing you did. (Ex. Say you put a bumper on the table & didn't want it, Click this & it's gone
Copy – copies a selected item. (Highlight or click on an item you want to copy & select copy
Paste – Pastes the Item you just copied (it puts the new item right over the original)
Delete – Deletes a selected item or a group of selected items.

Options – Opens up the option properties, clicking on different objects show different options
Options also has a button in the **Toolbar**

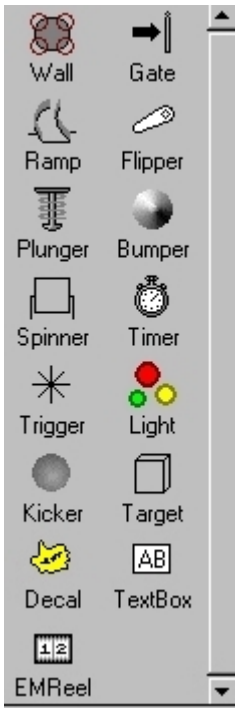
Script – open's your script editor. This is where you type to add sound, drop targets ect.
Sript also has a button in the **Toolbar**

Backglass View – gives a view of the backglass, usually where your points, credits, ect. Are displayed.
Backglass also has a button in the **Toolbar** called **Backdrop**

Next is the **Insert** options



Or it's the same as using the **ToolBar**



These are the objects that can be added to a Pinball Table. Ill go into these in the next part, along with their option's attributes

---

Next is the **Table** options **Table/**

Play – After you have a table loaded in the editor, use this to Play it
Play also has a button in the **Toolbar** and **F5** is a shortcut.

Sound Manager – Use this to import sounds into your table & use them in your
script by typing PlaySound "nameofsound"

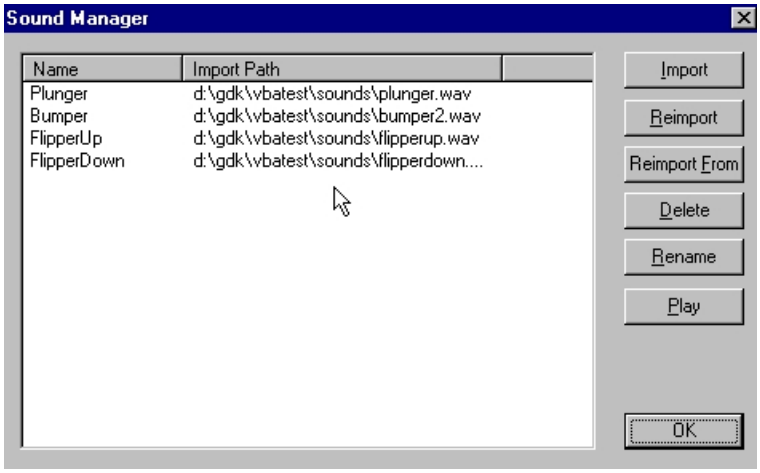
Image Manager – Use this to import images into your pinball table.
Click on an object & in the objects options, select an Image in the scroll down list.

Font Manager –

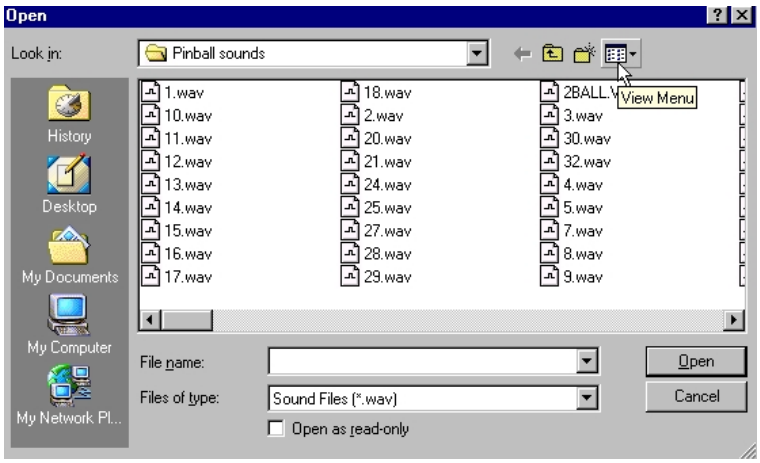
Collection Manager - select a group of items & make a collection with them

Magnify - Zooms in on a selected part of the table in the editor.
Magnify - also has a button in the **Toolbar**
Right click while it's selected will zoom out.

Sound Manager



After **Import** is selected, it will open this window(Below), from here you locate the directory your sounds are located in & select the Sound you wish to import. From there it will show up in the sound manager above.



Import- This will open up another window with a directory, look for a wave file you would like to add.

Once it's imported you need to save your table, then you would add it in the script Playsound "Bumper"

Reimport- Lets you replace the selected sound from the same Place you got the original Import

Reimport From- Lets you replace the selected sound, from another Directory.

Delete- Select the sound(Click on it) & click delete to get rid of it.

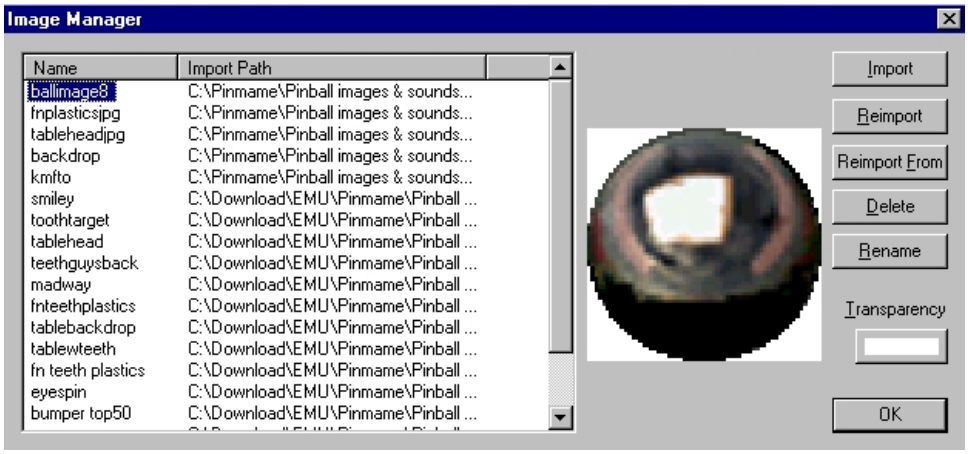
Rename- Select the sound(Click on it) & type over the hilighted text to rename it.

Play- Select the sound(Click on it) & click play to hear the Sound.

Note: Only .wav files are used right now for ingame sounds.

Mp3's can be used as music though. But they are used from the script. See the Faq. for more info on music.

Image Manager



Import- This will open up another window with a directory, look for a graphic file you would like to add.

Once it's imported you need to save your table.

Reimport- Lets you replace the selected graphic from the same Place you got the original Import.

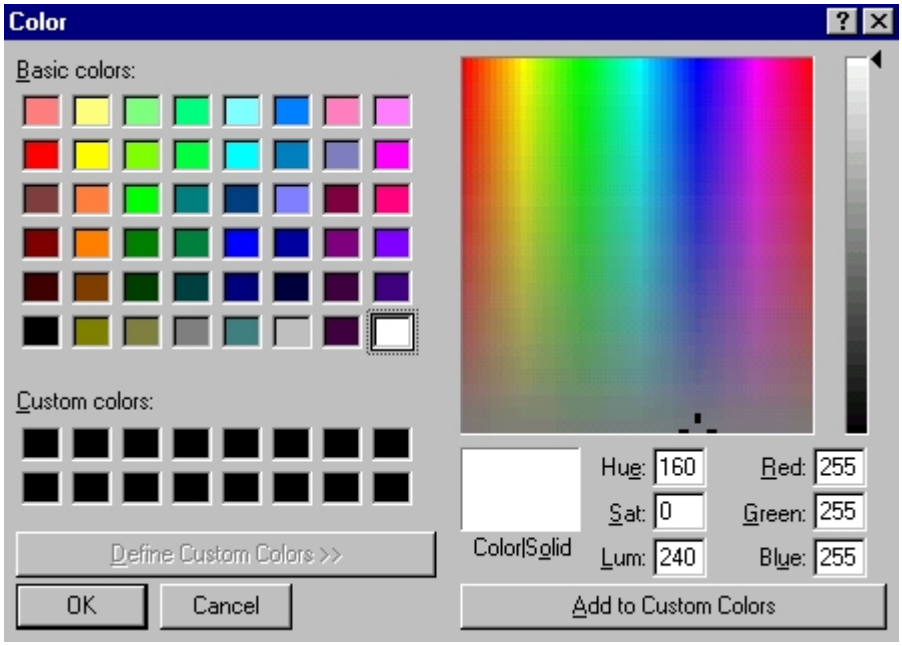
Reimport From- Lets you replace the selected graphic, from another Directory.

Delete- Select the graphic(Click on it) & click delete to get rid of it.

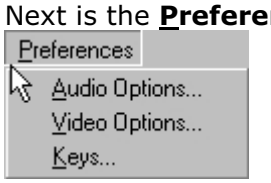
Rename- Select the graphic(Click on it) & type over the hilighted text to rename it.

See the Faq. for more info on graphics.

Transparency - Select the color you want to be Transparent. Notice the color white is selected for transparency, so this color will not show up when playing the Pinball Table.

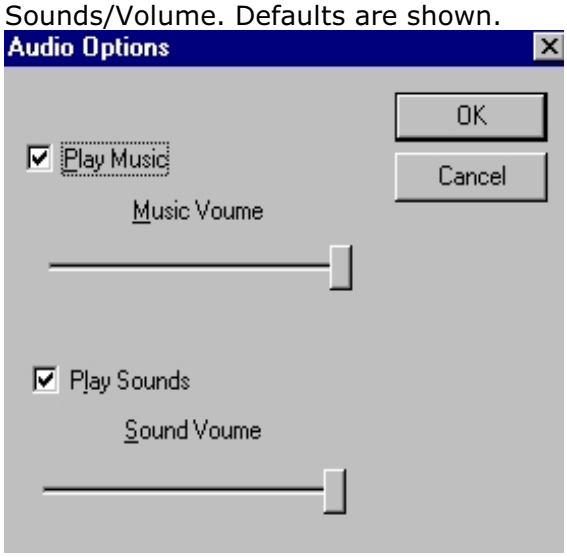


Next is the **Preferences** options

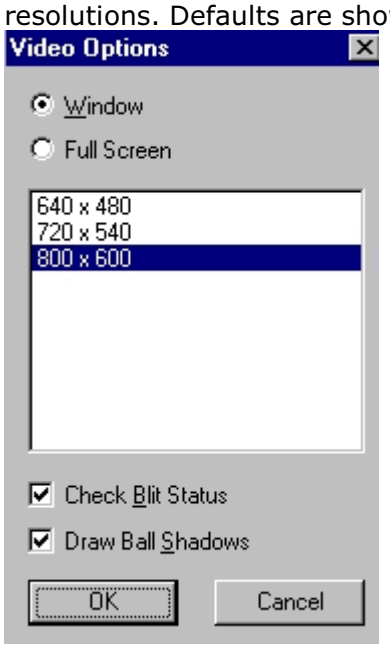


Preferences/

Audio - From here you can adjust the ingame Sounds/Volume. Defaults are shown.



Video - From here you can adjust ingame resolutions. Defaults are shown.



Keys- From here you can reassign keyboard commands for the keys used to play the Pinball games. Defaults are shown.

More Keycodes

Note:all keycodes are in the script of a basic table except the add credit & start game

Ex.The Plunger key is assigned to the Enter Key in the keys **Preferences**

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

Plunger is the Enter key as default, so if you change it to.....lets say L, the you would need to use L to launch the ball instead of enter.

So for add a coin & start game,

```
If keycode = AddCreditKey Then 'assigned key is 5
If keycode = StartGameKey Then 'assigned key is 1
```

you would need to add this into your script to use these(See Shivaengine.vps) you can open the script up (ShivaEngine.vps) in a text editor. This is the script for Shivaengine.

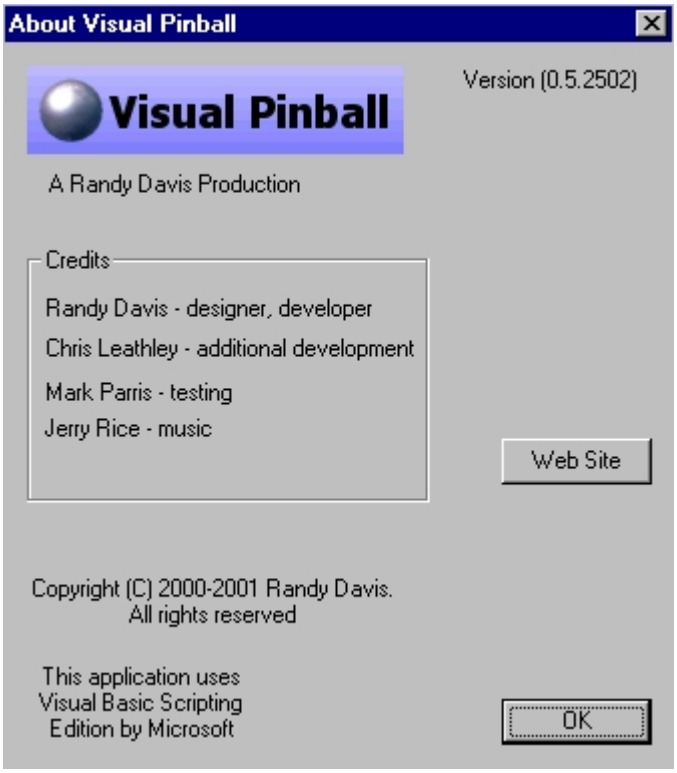
Next is the **Window** options

Not sure what the functions do here, but if you have more than 1 Table opened you can select between them. they will show up below the **Arange Icons**

Next is the **Help** options

About...

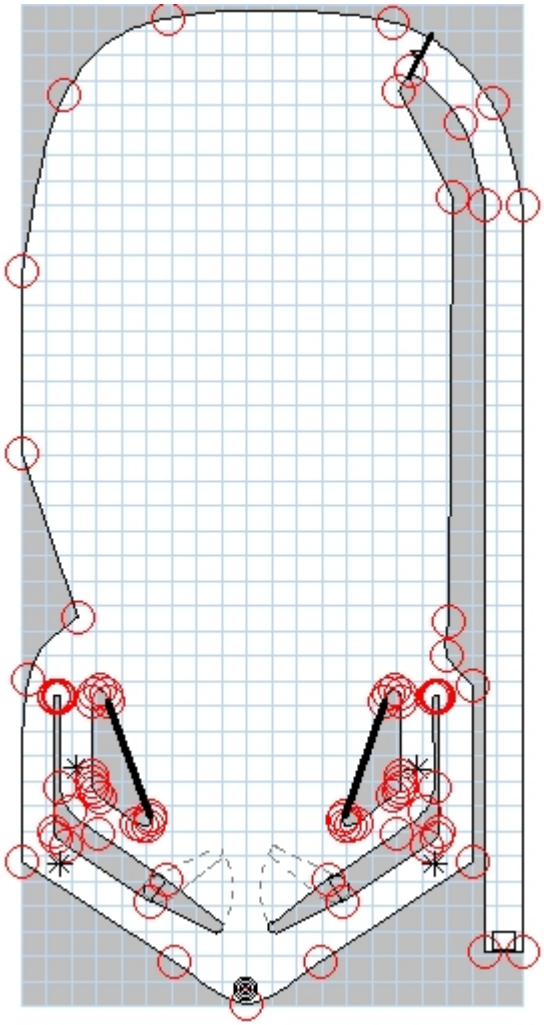
Shows you the current version of Visual Pinball you are running.



[Back to Top](#)

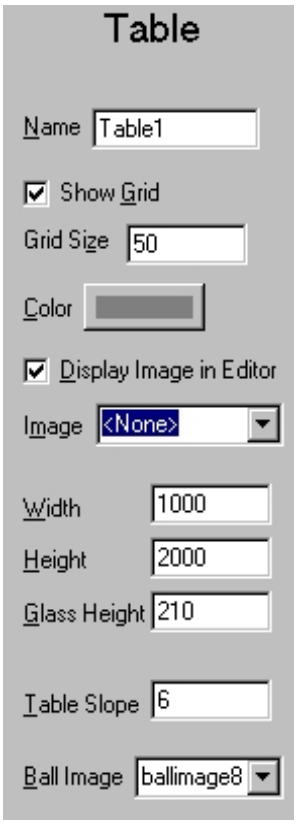
Editor Table Options

This is the basic **table** from the **File/New** option.



Click on the **table** and select options. Doing this brings up the Table Options

Table Options



you would **click on the table** & have the **options** selected from the toolbar for **Table Options**.

Table - The name of the object you have selected, which would be the table.

Name - *Table1* The name of the table, you can change this to whatever you want. Its ok to leave this as the name. Since this is the only table you'll be able to have in the template.Saving the table to a file doesn't change this.

Show Grid - It already will have a checkmark in it as default. Unchecking it will take away the horizontal & vertical lines. These are useful to have on when you need to line up object on the table.

Grid Size - Already set to 50, to make it farther apart, you enter a bigger number, & a smaller number to make the grid into smaller squares. You may need to change these later to get more detailed with the objects.

Display Image in Editor - If you have a table image imported, you would be able to see it. If your editor seems to be running slow, you can uncheck this box so Images don't show in the editor. That should help speed things up a little.

Image - Since table is selected, this would be your table image, after it's imported you would select it from the scroll down list. Table images are usually 512 x 1024 Pixels.

Width - Changes the Width of your table. 1000 is the default size.

Height - Changes the height of your table. 2000 is the default size.

Glass Height - The height of the glass, real pinball machines have glass and so does this 1. If you make an object(wall, Ramp ect.) Higher then the glass, your ball will get stuck. 210 is Default. Also The ball in VP is 50 units, So maximum heights of objects should be 50 Less then the 210

Table Slope - This changes the steepness or incline of the table, Default is 6.

Ball Image - You need to Import an image for the ball, if you want to use one. Default is just a plain ball but you can use the image supplied here. To use it you need to select it in the scroll down in **Ball Image**.

To use the ball image on your table just select it in the **Ball Image** drop down, It's that easy!. You used to have to add it in the script by adding Plunger.CreateBall.image = "ballimage8" in the script. "I used Plunger because thats in a new basic table script." But kickers can also create balls. You can still change the image or ball color in the table in script as needed.

64x64 pixels **Tip** 64x64 Pixels is a good size for creating Balls in a paint program.& export as.bmp not .jpg
ballimage8 Download

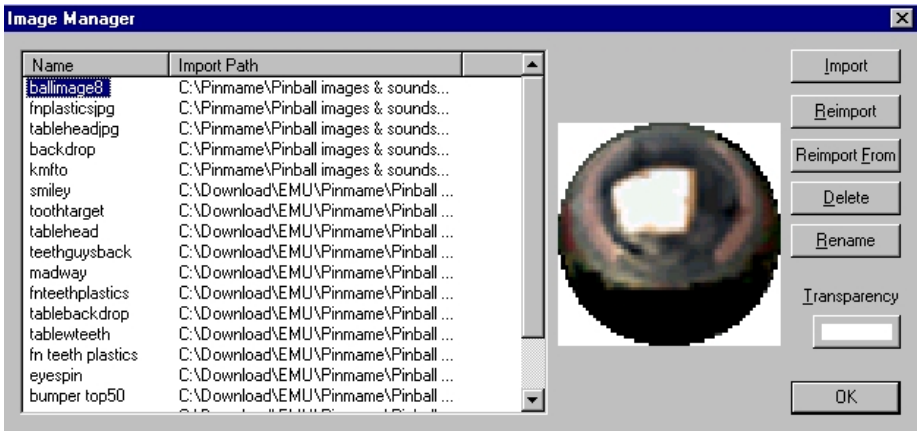
Back to Top

Changing Ball Colors RGB

You can also Change the color of the basic ball from within the script.
Ex. Plunger.CreateBall.color=rgb(255,255,255) 'White

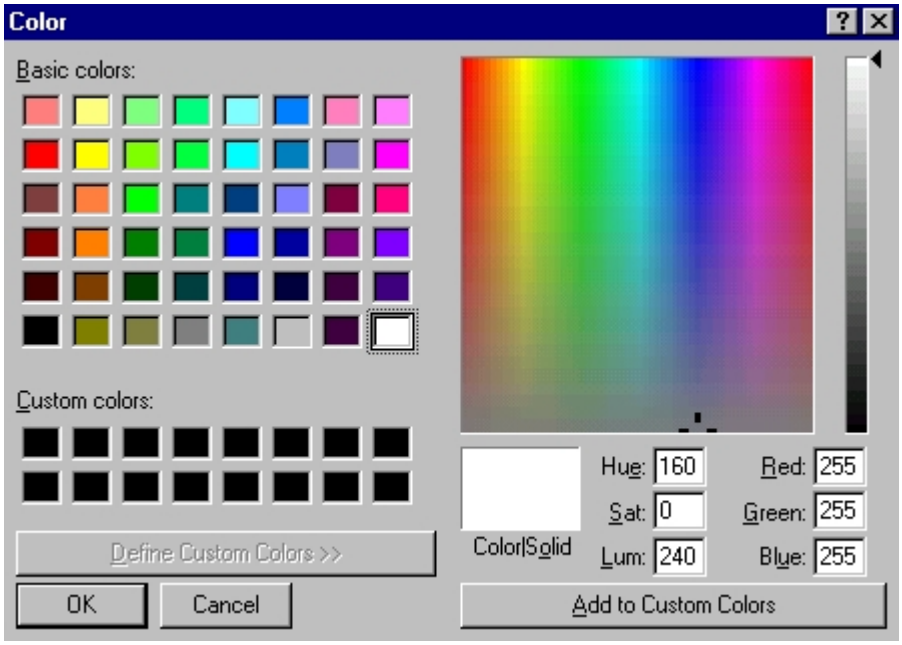
To get additional RGB Colors for Ball.Color you could use the Program Paint(Color/Edit Colors... Click on define custom colors)usually comes with windows or follow these steps.

1)Select Table/Image Manager...



2)Click on Transparency.

Note: A new table won't have any Images in the Image Manager.Just Ignore the Images & worry about Transparency, Button In the Picture.



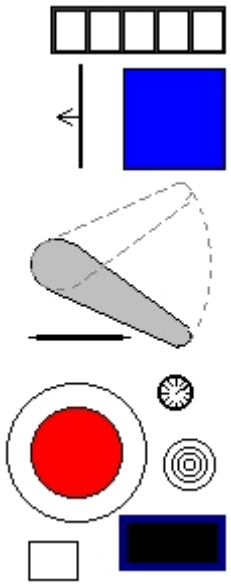
3)Click on a color in Basic Colors.
Note the
Red:255
Green:255
Blue:255
These are the **rgb** colors. Clicking on another Basic color changes the **rgb** colors. Use the **rgb** numbers to change the Ball Color

CreateBall.color=rgb(64,0,128) would give you a dark blue ball. The color right above the white.

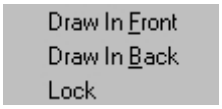
Back to Top

Right Clicking

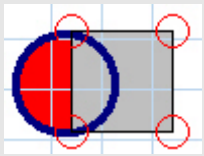
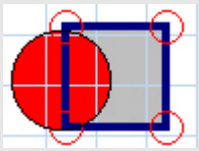
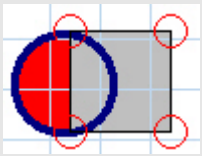
Gate, Decal, Flipper, Spinner, Timer, Bumper, Kicker, Plunger, TextBox, EMReel



All Have 3 options to choose from when you right click on them.



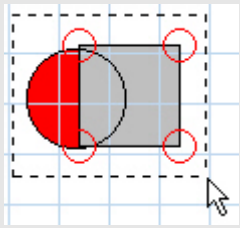
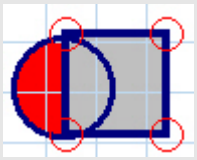
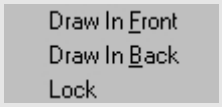
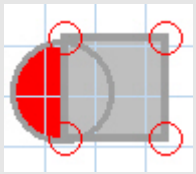
Lets go over **Draw in Back** First.

		
Lets say we added a light 1st to the table & then a wall over the light. Like this.	And these are the spots that we want the objects in. But the wall is in the way to edit the light. So we need to **select** the wall & **right click** on it & select the option **Draw in Back**	Now the light is in front of the wall. So now we are done with our light & want to edit the wall. **Select** the Wall & **right click** on it & select the option **Draw in Front.** Now the wall is Back in front of the light.

Lock

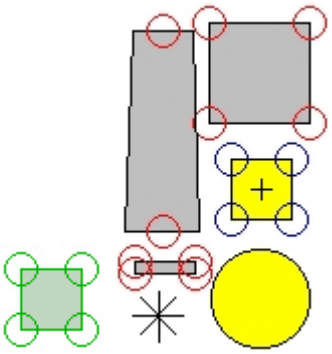
Lock is for Locking the objects to a table, It locks them in place, so you don't have to worry about accidentally moving them.

Lets say we wanted to lock the light & wall we were just editing.

			
Click & drag the mouse over both items.	While they are highlighted.	Right click the mouse over them. & select **Lock** from the options.	Now the are both Locked.

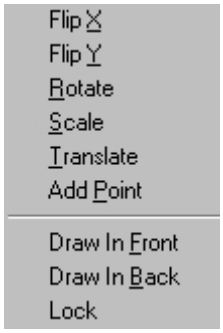
If you need to unlock them follow the same steps & select **Lock** again. It will have a check by it showing they are locked.

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

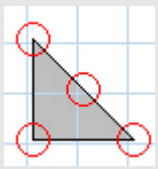


All Have 9 options to choose from when you right click on them. 3 we already covered.

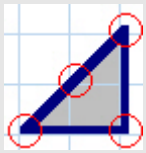
Note: **Trigger ShapeCircle** & **Light ShapeCircle** have no **Add Point** right click options.



Flip X

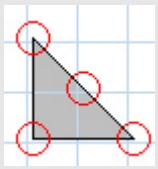


Using a triangle to show this. Click on the triangle to hilight it.
Select The Triangle & Right click over the Triangle & select **Flip X**

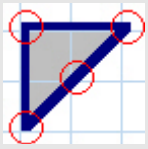


And this is what you get.
It Flipped The **X** Coordinates.

Flip Y

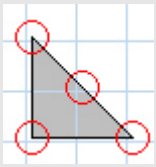


Using a triangle again to show this. Click on the triangle to hilight it. Right click over the Triangle & select **Flip Y**

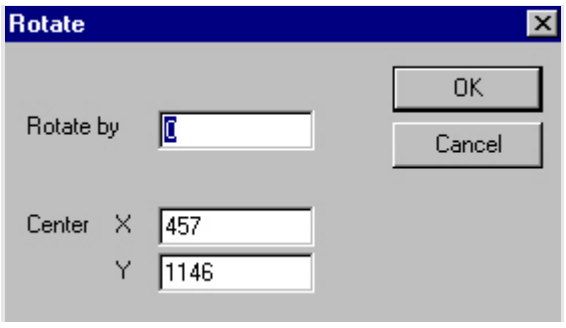


And this is what you get. It Flipped The **Y** Coordinates.

Rotate - Used to rotate objects

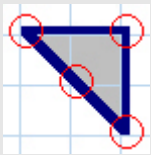


Using a triangle again to show this. Click on the triangle to hilight it. Right click over the Triangle & select **Rotate**



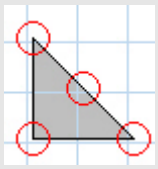
Where the 0 is enter 180.

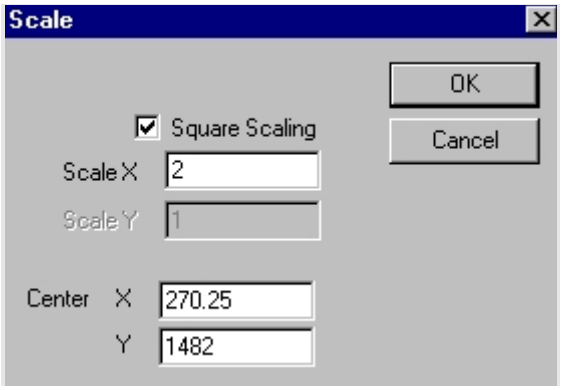
Note: **Center X** & **Y** are showing the location of the triangle on the table.

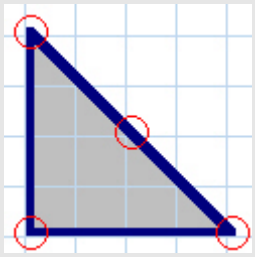


And this is what you get.
It **Rotate**d The Triangle 180 Degrees.

Scale - used to resize objects







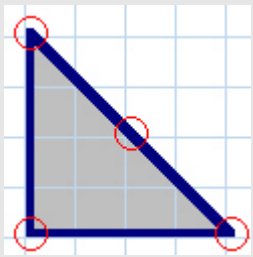
Using a triangle again to show this. Click on the triangle to hilight it. Right click over the Triangle & select **Scale**. It will open the Scale box.

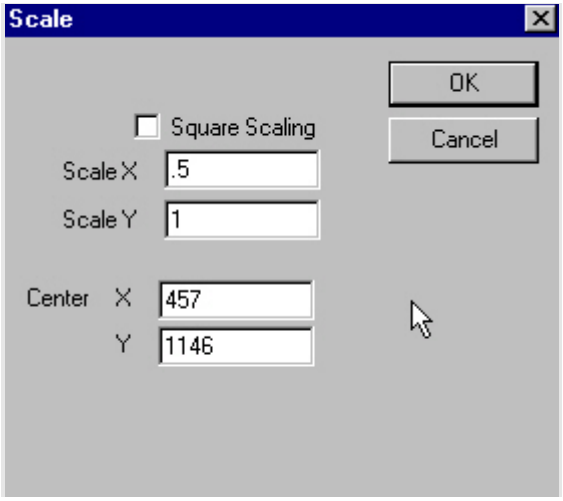
In **Scale X** where the 1 is enter 2 & then click ok.

Note: **Center X** & **Y** are showing the location of the triangle on the table.

And you get a triangle Twice as big.

Square Scaling Unchecked





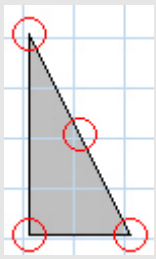
Using the triangle thats twice as big to show this. Click on the triangle to hilight it. Right click over the Triangle & select **Scale**. It will open the Scale box.

Take the check out of **Square Scaling** . Enter .5 for **Scale X** & Enter 1 for **Scale Y**

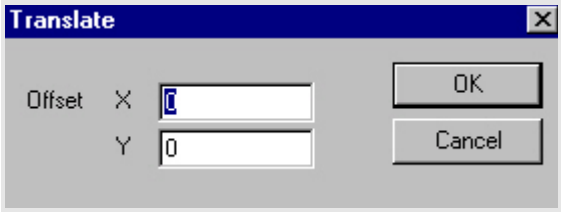
Note: **Center X** & **Y** are showing the location of the triangle on the table.

.5 was half & 1 was the same.

Translate - used to move objects



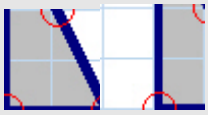
Using a triangle again to show this. Click on the triangle to hilight it. Right click over the Triangle & select **Translate**. It will open the Translate box.



Select 50 in **Offset X**

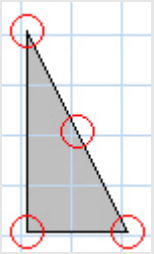
Note: **Grid Size** was set to 50 in Table Options. so it moved 1 whole grid square.

X - Left & Right
Y - Up & Down



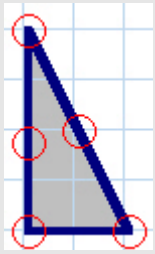
It moved it 50 Units on the grid to the right.

Add Point - used to add editing Points to objects for changing shapes.

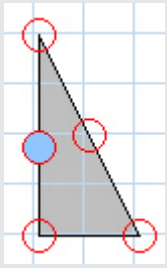


Using a triangle again to show this. Click on the triangle to hilight it. Right click over the Triangle & select **Add Point**

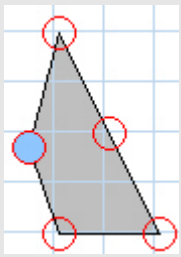
Note: Right click closer to the spot you want a Point.



A point was added to the Left Center of the Triangle.



Select The point by left clicking on it & holding.

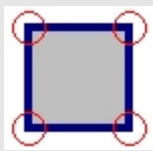


& drag it with the mouse & let go of button when its at the spot you want it.

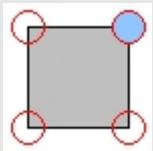
Back to Top

Building A simple slingshot from scratch

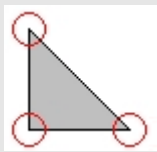
Using **File** / **New** to open a new Table.



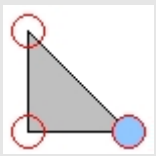
1) Create a wall. It Should be called wall1



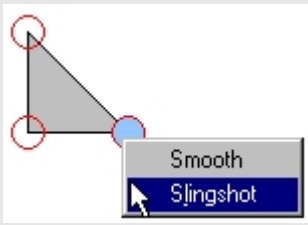
2) Select its top left Point



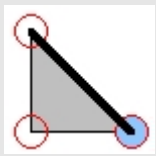
3) Delete the Point selected



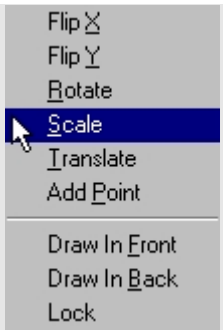
4) Select it's lower left Point



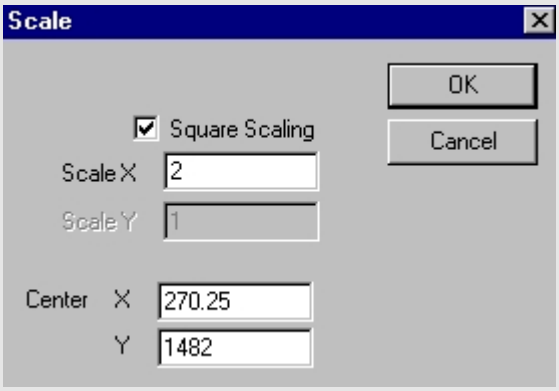
5) Right Click over Point & Select Slingshot



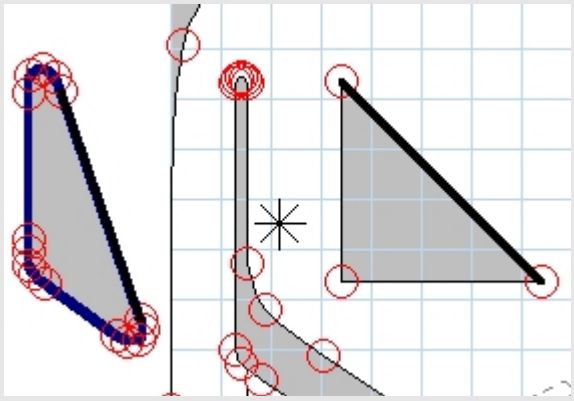
6) You now made that part of the Wall a slingshot



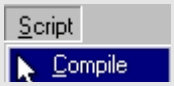
7) Right Click over the Triangular wall, Select Scale



8) Enter 2 in the ScaleX box & click ok. That will make it twice as big.



9) Swap The Triangle you made with the LeftSlingshot, just drag the LeftSlingshot off the table for now.

		
10) Add it to the script. Sub Wall1_SlingShot() PlaySound "Bumper" End Sub You may have to type the Sub in the script, copy/paste sometimes doesn't work. You can put this at the very end of script.	11) & compile it.	12) Give it a try Note: If you use Sub Wall1_Hit() PlaySound "Bumper" End Sub You won't get the slingshot sound effect.

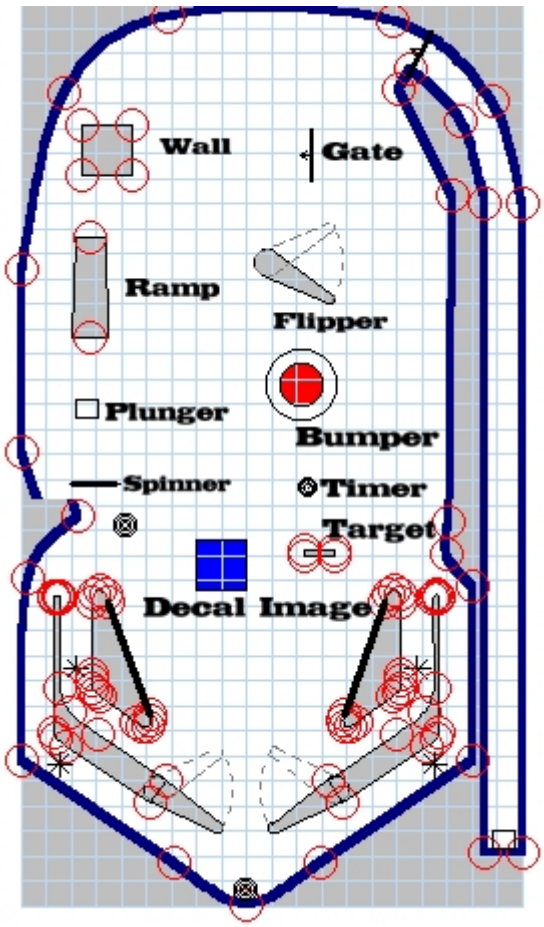
To make a more realistic target using Slingshots so side & back hits will not drop it:

Advanced User

Note: Slingshots are more sensitive near the center than the edge. So you need to divide the surface of the drop target into about ten sections or so, and make all of them slingshots. Now any ball that hits the front side will trigger the slingshot event, which will have code to drop the wall. Balls striking the side will not cause it to drop.
Another more reliable method. Simply place TWO drop targets lined up one behind the other, and only have the one in front has the hit event. Have them both dropped when the front is hit.

Back to Top

Editor Objects
Decal



Decal

Decals are used to place images or text on the Backdrop or Table.

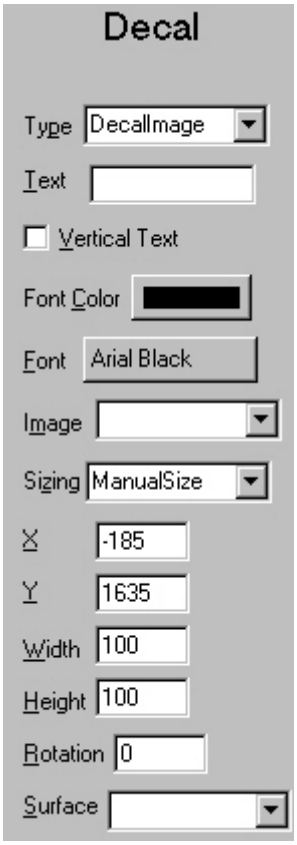
Decal Button

			
First Lets Create a Decal: If you don't have a table already opened **File** / **New** to open a new Table.	Click on the Decal button	Move Cursor towards the table you are building. It should turn into a Decal cursor, click this where you would like your Decal placed.	And you get a Decal.

Now that we have a Decal Created we can look at the Decal Option Properties.

	
Click on options in the **ToolBar**	And select the Decal by clicking on it.

Doing this will show us the Decal properties.



Decal - Name for type of object you have selected.

Type - Select **DecalImage** or **DecalText**

Text - Type the text you would like place in here for **DecalText** only

Vertical Text - Unchecked is Default. Check it to create vertical text. for **DecalText** only

Font Color - Select the color of the Font. for **DecalText** only

Font - Select the Font. for **DecalText** only

Image - Select an image you have imported. for **DecalImage** only

Sizing - Select from: ManualSize, AutoSize, AutoWidth.

X - Shows The X position of the Decal on the table.

Y - Shows The Y position of the Decal on the table.

Width - The Width of the Decal.

Height - The Height of the Decal.

Rotation - Changes the Angle of the Decal.

Surface - Selects a surface for the Decal to be placed on.

This Demo is to show you how to Import an Image for a decal & to show how an image with less pixels then it should have looks distorted.

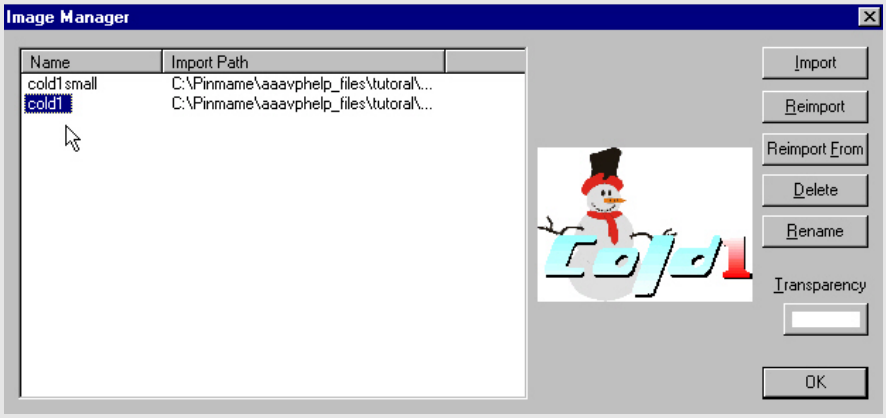
Using DemoDecal.zip

In this Demo there are 2 Bmp.files supplied. cold1small.bmp is 50x36 pixels & cold1.bmp is 300x215 pixels.

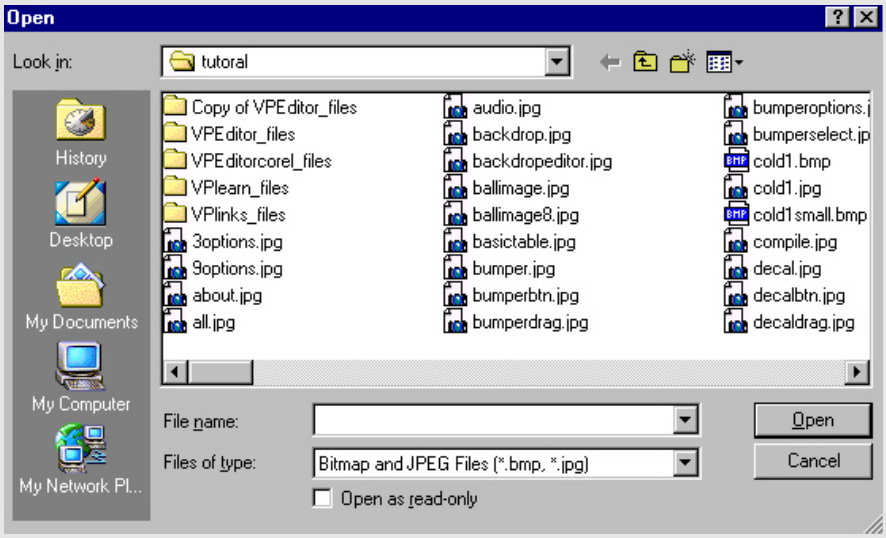
When looking at the demo when it's played you can see the smaller image isn't clear enough.

How To: Importing an image for a Decal

Selecting **Table/Image Manager** Opens up the Image Manager Box.

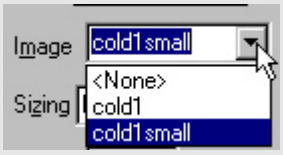


Selecting import will open up the next window.
Note: The Transparency color, this color you select will be transparent or invisible. Sometimes though when editing in a paint program you may be left with "Freckles" small pixels with color left in them. These will show up on your import , so to make sure you get them all try changing the part you want to be transparent to an odd color 1st, then change them to black or white.
DescibesUsing Paint & export as bmp



From here you would select the image you want to import.
Note: Directories will be different on your computer.
Once the image is selected it will show up in the **Image Manager** box. Click OK

From here you want to click on the decal (the blue square) & with options opened for the Decal

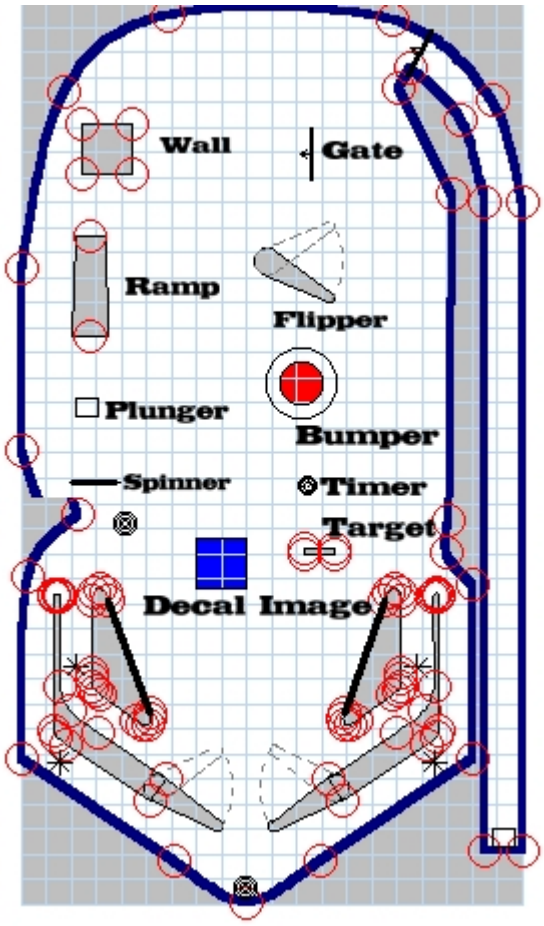


Select the **Image** scroll down menu & select your image.

Back to Top

Editor Objects
Plunger & Kicker

The Tool Bar



Plunger

The Plungers is used to launch the ball onto the table usuaully using the enter key.

Plunger Button

Note:The next step is to show how to create a plunger but can be skipped & use the plunger thats already on the new table. For a plunger to work correctly it needs walls around it to hold the ball in place or the ball would fall off.

First Lets Create a Plunger: If you don't have a table already opened **File** / **New** to open a new Table.

Click on the Plunger button

Move Cursor towards the table you are building.
It should turn into a Plunger cursor, click this where you would like your Plunger placed.

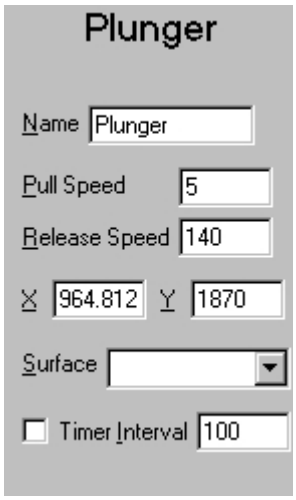
And you get a plunger.

Now that we have a Plunger Created we can look at the Plunger Option Properties.

Click on options in the **ToolBar** 

and select the Plunger by clicking on it. 

Doing this will show us the Plunger properties.



Plunger - Name for type of object you have selected.

Name - *Plunger* The name of the Plunger, you can change this to whatever you want. **Tip** Keep that name because it is already in the script, if you change the name you'll need to change it in the script also. The name is used to address the object through the script.

Pull Speed - The speed of the plunger being pulled.

Release Speed - The speed of the plunger when it is released (Launches Ball)

X - Indicates the **X** positon of the Plunger on table.

Y - Indicates the **Y** positon of the Plunger on table

Surface - In this scroll down box, you select which surface to place the Plunger on<None> is Default. Which is Actually the table itself. Remeber: A Plunger needs walls to keep the ball in place.

Timer Interval - If enabled will call PlungerName_Timer() in the script .Unchecked is default.

The Script That works A **Plunger**.Note: This is Just showing you the basic Script for a Plunger, you could add a Playsound"squeekylunger" for pullback & Playsound"balllaunch" for the fire, to give it some realistic Pinball sounds. Also this is just part of the keycodes Sub, you always need to end a sub with 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

The Kickers can be used to launch the ball onto the table.

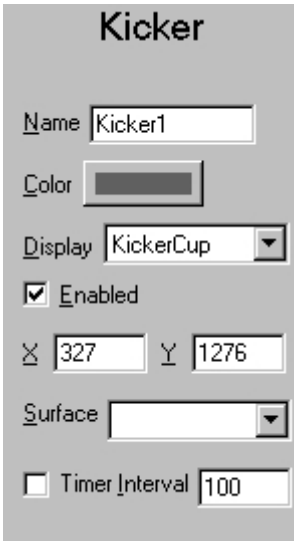
Kicker Button

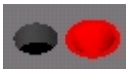
			
First Lets Create a Kicker: If you don't have a table already opened **File** / **New** to open a new Table.	Click on the Kicker button	Move Cursor towards the table you are building. It should turn into a Kicker cursor, click this where you would like your Kicker placed.	And you get a Kicker.

Now that we have a Kicker Created we can look at the Kicker Option Properties.

Click on options in the **ToolBar** 

and select the Kicker by clicking on it. 
Doing this will show us the Kicker properties.





KickerHole,KickerCup

Kicker Hidden- Doesn't Show up on table, but is there.

Kicker - Name for type of object you have selected.

Name - *kicker1* The name of the Kicker, you can change this to whatever you want. **Tip** I Usually keep the name, & only rename it if I have a bunch of kickers or it's being used as part of something. The name is used to address the object through the script.

Color - Opens the color pop up window, so you can change the color of it.

Display - Scrolldown Has 3 options, KickerHole(Default)KickerCup & KickerHidden

Enabled - Have this checked to enable the kicker to work. A Kicker also needs some script for it to work.

X - Indicates the **X** positon of the Kicker on table.

Y - Indicates the **Y** positon of the Kicker on table.

Surface - In this scroll down box, you select which surface to place the Kicker on<None> is Default. Which is Actually the table itself.

Timer Interval - After some script, this works with the Kicker. If enabled will call KickerName_Timer() in the script .Unchecked is default. See. demo0kickertimer.zip for a demo of this function.

Script is Green so you know it's script.
Kicker Functions:
A Kicker can:
Kicker1.DestroyBall - Destroys the Ball
Kicker1.CreateBall - Creates A Ball
Kicker1.Kick

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

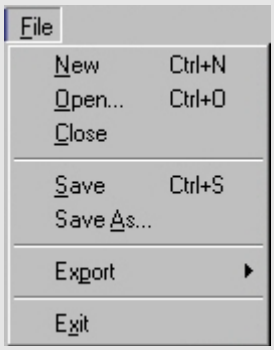
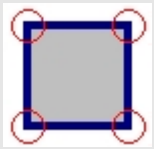
The first parameter is direction, with 0 being due north.
The second parameter is 10 which is the strength of kick.
The angles are 0 to 359 technically though negatives appear to work as well.
Think of the directions as a clock face that goes
---------- 0
---------305-------45
--------270---------90
---------225-------135
----------180
For the direction you want to kick it. See Faq for more on Kickers

Back to Top

Editor Objects
Walls & Targets

The Walls

Wall Button

			
First Lets Create a wall: If you don't have a table already opened **File** / **New** to open a new Table.	Click on the wall button.	Move Cursor towards the table you are building. It should turn into a wall cursor, click this where you would like your wall placed.	And you get a wall. The red circles are called Points. Clicking on them & Dragging will change the wall's shape. More On **Points** Later.

Now that we have a wall Created we can look at the Wall Option Properties.

Click on options in the **ToolBar**  and select the wall by clicking on it.  doing this will show us the wall properties.

Wall - Name for type of object you have selected.

Name - *Wall1* The name of the wall, you can change this to whatever you want. Its ok to leave this as the name.**Tip** I usually only change wall names if I refer to them in the script.The name is used to address the object through the script.

Image - Since wall is selected, this would be your wall's image, after it's imported you would select it from the scroll down list.Note:Wall images can be used from the table image, the table image is usually 512 x 1024 Pixels.

Image Alignment - ? doesn't seem to do anything yet.

Display Image in Editor - If you have an image imported, you would be able to see it inside the wall if you have this box checked.Note: If your editor seems to be running slow, you can uncheck this box so Images don't show in the editor. That should help speed things up a little.

Top Color - Clicking this opens the Color Box, from there you can select the top color of your wall.

Side Color - Clicking this opens the Color Box, from there you can select the side color of your wall.

Top Height - Changing this number makes your walls top height, higher or lower Default is 50 on any new wall you make.

Bottom Height - Changing this number makes your walls bottom height, higher or lower Default is 0 on any new wall you make.Note: making this -1 puts them below the table's surface when dropped, so if can drop is checked and you got the script to go with it.

Has Hit Event - This needs to be checked if the wall is scripted to do something.

Hit Threshold - The ball has to hit a wall at least this hard in order to register a hit event, the Higher the threshold number, the more power the ball needs behind it before the target will drop.

Slingshot Force - The strength of the force the ball will be reflected off the object.

Elasticity - The Elasticity of an object, Similar to **Slingshot Force**.

Can Drop - If you have the wall dropping in the script you will need to have this checked.

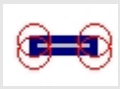
Timer Interval - If enabled will call WallName_Timer() in the script .Unchecked is default.

Note:Wall 's & Targets are almost the same when making new ones.
Differences are:Color, Size/Shape & a Target has a check in it's **Has Hit Event** box. Targets are even called walls when made, but I usually rename these to Targets, because they are referred to in the script & it makes it easier to remember.

Back to Top

The Targets

Target Button

			
First Lets Create a Target: If you don't have a table already opened **File** / **New** to open a new Table.	Click on the Target button	Move Cursor towards the table you are building. It should turn into a Target cursor, click this where you would like your wall placed.	And you get a Target. The red circles are called Points. Clicking on them & Dragging will change the Target's shape. More On **Points** Later

Now that we have a Target Created we can look at the Target Option Properties.

Click on options in the **ToolBar**  and select the Targetby clicking on it.  doing this will show us the Target properties. Actually shows as Wall

Wall - Name for type of object you have selected. Even though this is a **Target.** I refer to it as a Target below.

Name - *Wall1* **Changed Name to Target** for easier reference & script uses.**Tip** I usually change all Target names because I refer to them in the script. Ex. Target1, Target2. The name is used to address the object through the script.

Image - Images on Targets are a bit tricky and most use Hacks of Ball Images & Kickers to simulate the effect. So it's best to leave this blank & just change its colors. Maybe in a future version of VP we will have a better solution to work with.

Image Alignment - ? doesn't seem to do anything yet.

Display Image in Editor - Doesn't use images yet. See **Image.**

Top Color - Clicking this opens the Color Box, from there you can select the top color of your Target.

Side Color - Clicking this opens the Color Box, from there you can select the side color of your Target.

Top Height - Changing this number makes your Targets top height, higher or lower Default is 50 on any new wall/Target you make.

Bottom Height - Changing this number makes your Targets bottom height, higher or lower Default is 0 on any new wall you make.

Note: making this -1 puts them below the table's surface when dropped, so if can drop is checked and you got the script to go with it.

Has Hit Event - This is checked by Default for the Target. needs to be scripted to do something.

Hit Threshold - The ball has to hit a wall at least this hard in order to register a hit event, the Higher the threshold number, the more power the ball needs behind it before the target will drop.

Slingshot Force - The strength of the force the ball will be reflected off the object.

Elasticity - The Elasticicity of an object, Similar to **Slingshot Force**.

Can Drop - If you have the Target dropping in the script you will need to have this checked.

Timer Interval - If enabled will call WallName_Timer() / TargetName_Timer() in the script .Unchecked is default.

This is Basic Script for a Target:See demo1targetdrop.zip
Script is Green so you know it's script.

```
Sub Target_Hit() 'The Sub for Target, inside the sub is what happens when Target is hit

Playsound "targetdropping" 'Plays this Sound when hit.
'AddScore (50) 'Adds a Score of 50 when hit but we have no score in script yet
Target.IsDropped=True 'This Drops The Target
CheckTarget() 'This would check that Sub to see if targets are dropped & reset them

End Sub

Sub Target1_Hit() 'The Sub for Target, inside the sub is what happens when Target is hit
Playsound "targetdropping" 'Plays this Sound when hit.
'AddScore (50) 'Adds a Score of 50 when hit but we have no score in script yet
Target1.IsDropped=True 'This Drops The Target
CheckTarget() 'This would check that Sub to see if targets are dropped & reset them
End Sub
```

at some point you would wan't your Target to pop back up, it's usually put into a CheckTarget Sub

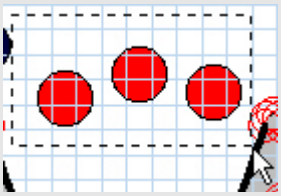
```
Sub CheckTarget()
If Target.Isdropped = True And Target1.IsDropped = True Then
Target.IsDropped=False
Target1.Isdropped = False
End IF
End Sub
```

See. Targets-BasicTutorial.txt for more info on Targets From the ShivaEngine1_7.zip

---

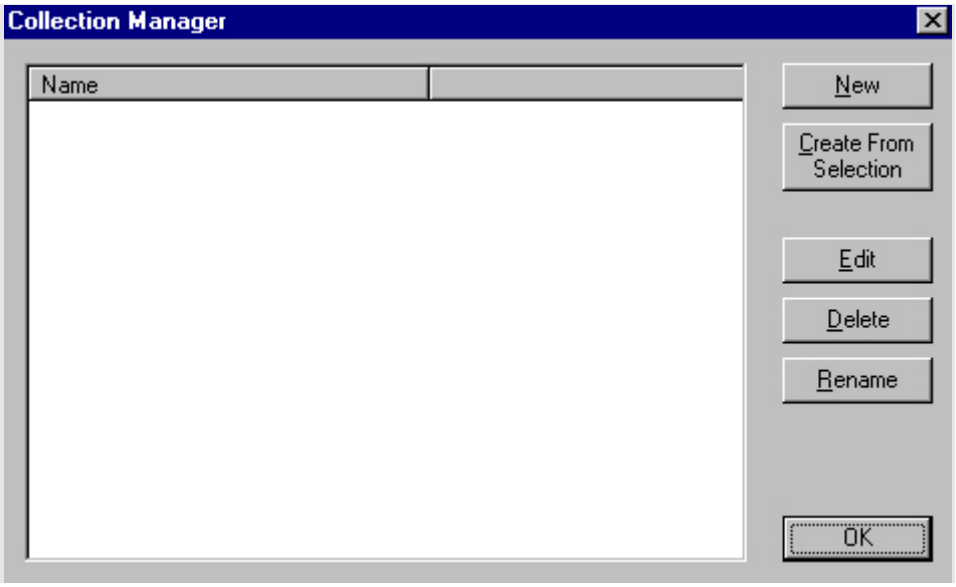
Back to Top

Editor Objects
Making Light Collections with Collection Manager

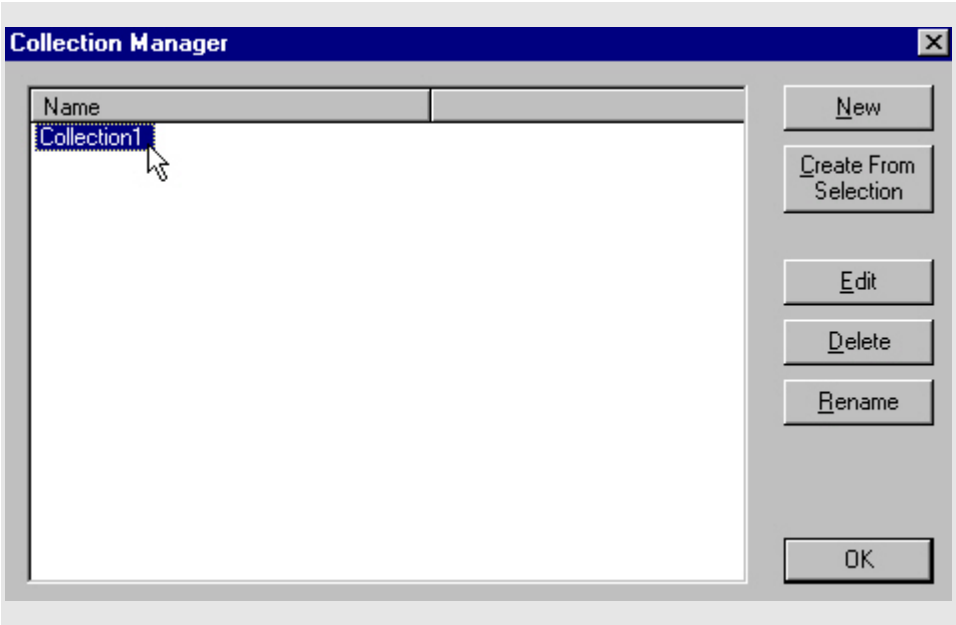


Select a group of Lights you want in the collection

On the Options Bar select Table/Collection Manager...



Click Create From Selection



It Creates a collection called Collection1

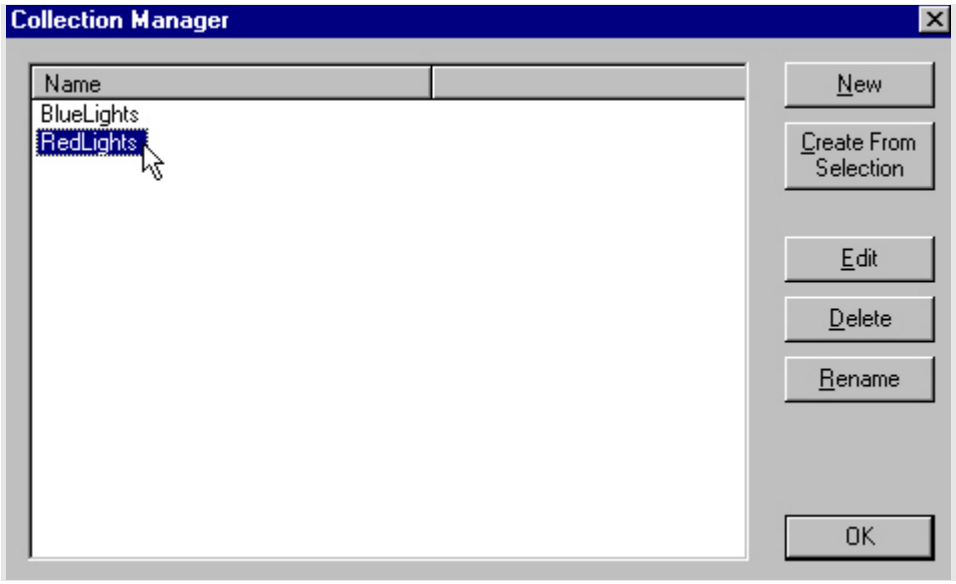
New - Creates a new Collection

Create From Selection - Creates a collection from the objects you have selected.

Edit - Opens up the Collection Edit Window.

Delete - Deletes selected Collection

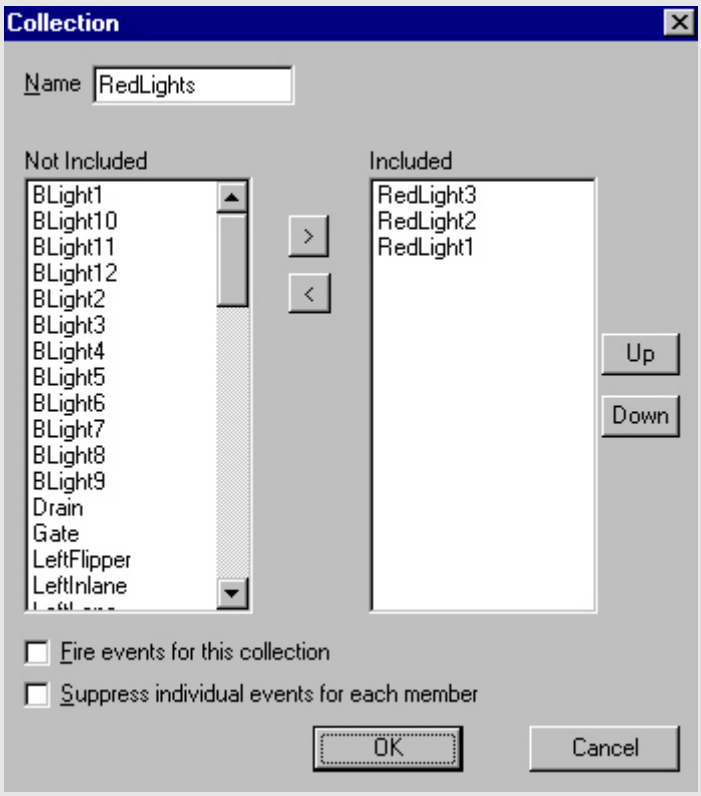
Rename - Renames Selected Collection



Using DemoLights.zip There are already 2 collections in the file.

From here Select RedLights, & click **Edit**

Clicking **Edit** opens up the window below.



Name - Name of the collection to edit. The name is used to address the object through the script.

Not Included - List objects not in the collection

Included - Lists objects that are in the selected collection.

> - Adds objects to the collection.

< - Takes objects out of the collection.

Up - Moves a selected item up the list.

Down - Moves a selected object down the list.

Fire Events for this collection -

Suppress individual events for each member -

Here is A Visual Pinball Tutorial That demonstrates the script below on Light.State
Script is Green so you know it's script.
Using DemoLights.zip

"For Each" support in script!

Example.

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

Where 'light' is just a temporary name you assign to each element as you use it. This saves the trouble of messing with the size of the collection.
Note: These are the Keycodes used in the script from the above demo that change the light states.

```
If keycode = PlungerKey Then
Plunger.Fire
'This section will turn scripted lights to the Off State when Plunger/Enter is released

If keycode = LeftFlipperKey Then
LeftFlipper.RotateToStart
PlaySound "FlipperDown"
'This section will turn scripted lights to the on State when LeftFlipper/Left Shift is pressed
```

```
If keycode = RightFlipperKey Then
RightFlipper.RotateToStart
PlaySound "FlipperDown"
'This section will turn scripted lights to the Blinking State when RightFlipper /Right Shift is pressed

'States are as followed 0 = LightStateOff so you can type 0 or LightStateoff
' 1 = LightStateon
' 2 = LightStateBlinking
'************************ Array
'Using an array to change a group of light states
LightsArray = Array(Light1,Light2,Light3,Light4,Light5,Light6,Light7,Light8)
For a = 0 to 7 'a can be any letter 0 to 7 = 8 lights
LightsArray(a).State=2 'makes the 8 lights blink
Next
'************************ Collections
'The Green Lights & Red Lights & Blue Lights are Collections of lights
'To see their collection, In the Editor select
'Table / Collection Manager (Select a collection) / Edit
' Collection Using "For Each" support!
For Each Light in GreenLights
Light.state = LightStateBlinking 'or 2
Next

For Each Light in Redlights
Light.State=2 'or LightStateBlinking 'These are the 3 red lights above the flippers
Next
'It can also be done like this but the "For Each" is simpler
For a = 0 to 11 '0 to 11 = 12 lights, 0 counts as the 1st number
BlueLights(a).State=2 'these are the 12 BlueLights blinking clockwise
Next

'************************ Setting Lights
'And The most time consuming way to control your lights (lots of typing :)
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 to 11 = 12 lights
Red12Lights(a).State=2 'these are the 12 Red Lights blinking counter clockwise
Next
```

Back to Top

Editor Objects
Lights & Bumpers

Lights

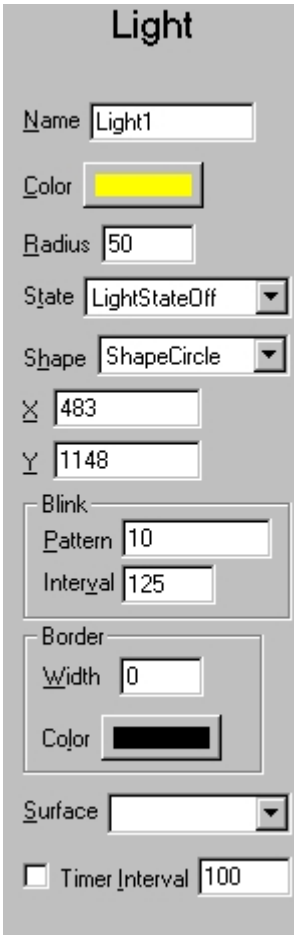
Lights have 3 states on, off & blinking, & can be placed anywhere on table or backdrop.

Light Button

			
First Lets Create a Light: If you don't have a table already opened **File** / **New** to open a new Table.	Click on the Lights button	Move Cursor towards the table you are building. It should turn into a light cursor, click this where you would like your light placed	And you get a light.

Now that we have a Light Created we can look at the Light Option Properties.

Click on options in the **ToolBar**  and select the Light by clicking on it. 
Doing this will show us the Light properties.



Light - Name for type of object you have selected.

Name - *Light1* The name of the light, you can change this to whatever you want. **Tip** I usually change light names to similar names when I have a group of them. Ex. HeartLight1,HeartLight2. The name is used to address the object through the script.

Color - Clicking this opens the Color Box, from there you can select the color of your Light.

Radius - Changing the number Larger makes a bigger Light circle, Smaller Number for a smaller circle. Default is 50

State - has 3 options - LightStateOff - LightStateOn - LightStateBlinking **or** 0 - 1 - 2

Note:When entering LightStates in script - you could type it either way below.

Light1.State=LightStateOff		Light1.State=0
Light1.State=LightStateOn	**or**	Light1.State=1
Light1.State=LightStateBlinking		Light1.State=2

Shape - Has 2 Options - ShapeCircle  & ShapeCustom 

X - Indicates the **X** position of the Light on table.

Y - Indicates the **Y** position of the Light on table

Blink:

Pattern - 10 is Default. which is on-off(Repeat) If you have it set to LightStateBlinking or 2(script)

Interval - 125 is is the time it takes for each light to change to the next pattern. 1000 = 1 Second

Border:

Width - Higher Number will increase the **Border Width**.

Color - Selects the color of the **Border Color**.

Surface - In this scroll down box, you select which surface to place the light on<None> is Default. Which is Actually the table itself. Surfaces could be another wall or ramp, other surfaces are usually higher than the Table.

Timer Interval - Interval bypasses this unless scripted. If enabled will call LightName_Timer() in the script .Unchecked is default.

Note:In **ShapeCustom**  The small blue circles are called Points. Clicking on them & Dragging will change the Light **ShapeCustom**s shape.See Building a simple Sling Shot for more info on Points.

Bumpers

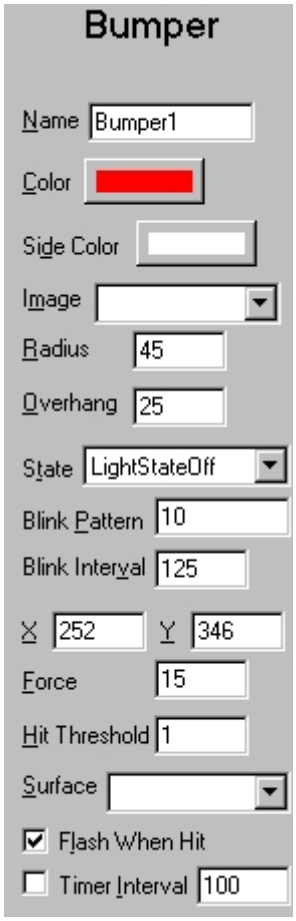
Bumpers have 3 states on, off & blinking, & can be placed anywhere on table.

Bumper Button

			
First Lets Create a Bumper: If you don't have a table already opened **File** / **New** to open a new Table.	Click on the Bumper button	Move Cursor towards the table you are building. It should turn into a Bumper cursor, click this where you would like your Bumper placed.	and you get a Bumper. .

Now that we have a Bumper Created we can look at the Bumper Option Properties.

Click on options in the **ToolBar**  and select the Bumper by clicking on it. 
Doing this will show us the Bumper properties.



Bumper - Name for type of object you have selected.

Name - *Bumper1* The name of the Bumper, you can change this to whatever you want. **Tip** I usually don't change Bumper names. The name is used to address the object through the script.

Color - Clicking this opens the Color Box, from there you can select the top color of your Bumper.

Side Color - Clicking this opens the Color Box, from there you can select the top color of your Bumper.

Image - Select an image from the drop down list after you have one imported. When drawing In a drawing program, bumper images I usually make them 300 x 300 pixels & export them as a .bmp (*Bumper images now light up)

Radius - Changing the number Larger makes a bigger Bumper circle, Smaller Number for a smaller Bumper circle. Default is 45

Overhang - Changes the size of the overhang of the Bumper or top part.

State - has 3 options - LightStateOff - LightStateOn - LightStateBlinking **or** 0 - 1 - 2

When entering LightStates in script for bumpers - you could type it either way below. Usually bumper states aren't changed, a check in **Flash When Hit** usually takes care of what a bumper needs to do but you could also make give them States when hit if needed. See Below.

Bumper1.State=LightStateOff		Bumper1.State=0
Bumper1.State=LightStateOn	**or**	Bumper1.State=1
Bumper1.State=LightStateBlinking		Bumper1.State=2

Blink Pattern - 10 is Default. which is on-off(Repeat) If you have it set to LightStateBlinking or 2 in the (script)

Blink Interval - 125 is is the time it takes for each Bumper to change to the next pattern. 1000 = 1 Second

X - Indicates the **X** position of the Bumper on table.

Y - Indicates the **Y** position of the Bumper on table

Force - How Strong Bumper bounces the ball off of it after it is hit.

Hit Threshold - The amount of force from the ball it would take to register a Bumper Hit. (Default is 1)

Surface - In this scroll down box, you select which surface to place the Bumper on<None> is Default. Which is Actually the table itself. Surfaces could be another wall or ramp, other surfaces are usually higher than the Table.

Flash When Hit - a checkmark in this box will make the Bumper flash when the ball hits it, take out the checkmark & the Bumper wont flash when hit.

Timer Interval - Blink Interval bypasses this unless scripted. If enabled will call BumperName_Timer() in the script .Unchecked is default.

For Using Lights in a collection see The Light Collections Page
Here is A Visual Pinball Tutorial That demonstrates the script below on Light.State & Bumper.State
Script is Green so you know it's script.
demo2light&bumperstates.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 is 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 is inside the sub
PlaySound "Bumper"
Light1.State=2 ' 2 is ame 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 is inside the sub
PlaySound "Bumper"
Bumper2.State = LightStateBlinking ' or 2
End Sub

'*************************Light Stuff Ends Here
```

Back to Top

Editor Objects
Gate & Sound Events

Gate

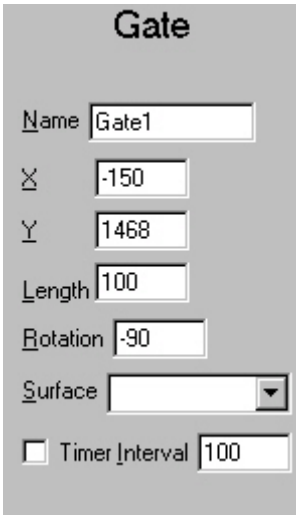
Gate are used to make ball travel 1 directional after passing them, & can be placed anywhere on table as longs as the ball can travel through them.

Gate Button

			
First Lets Create a Gate: If you don't have a table already opened **File** / **New** to open a new Table.	Click on the Gate button Move Cursor towards the table you are building	It should turn into a Gate cursor, click this where you would like your Gate placed.	And you get a Gate.

Now that we have a Gate Created we can look at the Gate Option Properties.

Click on options in the **ToolBar**  and select the Gate by clicking on it. 
Doing this will show us the Gate properties.



Gate - Name for type of object you have selected.

Name - *Gate1* The name of the Gate, you can change this to whatever you want. **Tip** I usually Keep Gate names. The name is used to address the object through the script.

X - Indicates the **X** position of the Gate on table.

Y - Indicates the **Y** position of the Gate on table

Length- 100 is Default. Here you can Increase(100+)or Decrease(Smaller number)the **Gate** Length.

Rotation - -90 is Default. Changing this changes the angle of the **Gate**.

Surface - In this scroll down box, you select which surface to place the **Gate** on<None> is Default. Which is Actually the table itself. Surfaces could be another wall or ramp, other surfaces are usually higher than the Table.

Timer Interval - If enabled will call GateName_Timer() in the script .
Unchecked is default.

Using demo2light&bumperstates.zip
Here is some basic script for adding sounds to events. Script is Green so you know it's script.

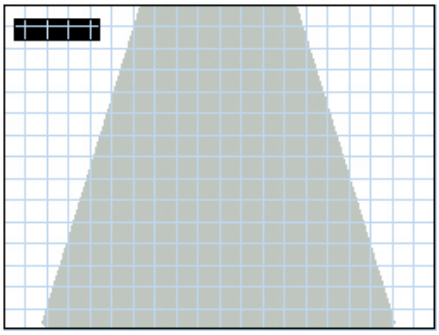
```
'Gate_Hit sub will turn on the gatelight when hit

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
```

Notice the PlaySound in the PlungerKey .PullBack & .Fire

```
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
```

Back to Top

Editor BackDrop Objects
Textbox & Making Score Count

The Backdrop Tool Bar  Back Drop View of Editor

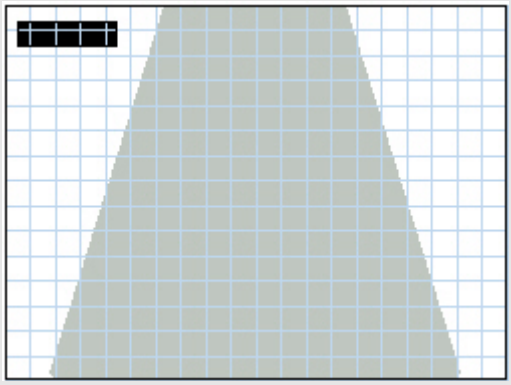
---

TextBox

TextBoxes are used to Display Text.

TextBox Button

			
First Lets Create a TextBox: If you don't have a table already opened **File** / **New** to open a new Table.	Click on the TextBox button Move Cursor towards the Backdrop of the table you are building	it should turn into a TextBox cursor, click this where you would like your TextBox placed.	And you get a TextBox.



Note:It can be placed anywhere but, usually kept off the gray area which is the table.

Note: A New Table already comes with a TextBox, called ScoreText. But isn't scripted to do anything yet.
Now that we have a TextBox Created we can look at the TextBox Option Properties.

Click on options in the **ToolBar** 

and select the TextBox by clicking on it. 
Doing this will show us the TextBox properties.

TextBox - Name for type of object you have selected.

Name - *ScoreText* The name of the TextBox, you can change this to whatever you want. **Tip** I usually name the **TextBox** to what it displays. ScoreText is a perfect name for displaying text. The name is used to address the object through the script.

Text- You can type any text in here. It will be displayed in the **TextBox** when table is played.

Transparent - Makes the **Back Color** invisible or Transparent.

Back Color - Change the color behind the Text.

Text Color - Changes the color of the Text.

Font - Opens a box to select the Font, Font Style & Size.

Alignment - Sets the text to Align Center, Left, Or Right, in the **TextBox.**

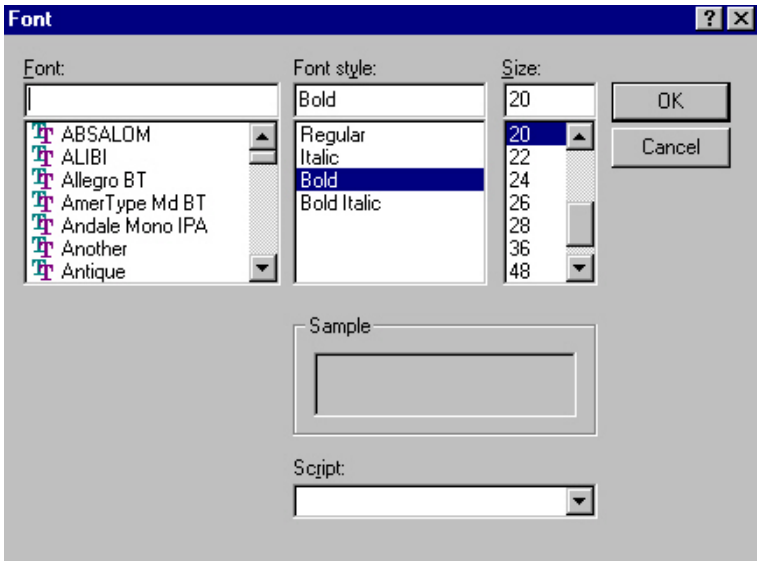
X - Indicates the **X** position of the TextBox on table.

Y - Indicates the **Y** position of the TextBox on table.

Width - Changes the Width of the **TextBox.**

Height - Changes the Height of the **TextBox.**

Timer Interval - If enabled will call TextBoxName_Timer() in the script. Unchecked is default.Unchecked is default.



Selecting **Font** Opens up the Font Option's box. From here you can select the Font, Font Style & Size.

The **Sample** box shows a preview of the selected font.

Using demo3scoretextbox.zip

Here is the basic script changes to go with the above Demo , for adding score and having it displayed in the **TextBox**. Script is Green so you know it's script.

In the beginning of the script we will add the next line. Note: the ' is a Rem & isn't read in the script. Use ' to leave notes in script.

```
Dim Score 'see sub addscore this is a variable meaning the number changes.
```

Run the Table 1st, to see it adding the score & Then try this below if you don't understand Dim.
Dim declares Score as variable, Score is now a variable, meaning it changes, so when we Dim it it will be remembered & added to.Without it, Score will only display the last AddScore hit event to the TextBox Display. To show This & to understand better what I'm getting at, put a ' or rem in front of the Dim Score line. & Compile the Script. Now the Score won't add, it will just diplay the Score for the last AddScore that registered.

This is the Sub for adding the Score.

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

Sub AddScore(points) 'we also need to Dim Score in the beginning of script. all Variables should go in the beginning of script.
Score = Score + points ' This adds your score + the points in the (Brackets) when something is hit & it contains AddScore(#)
ScoreText.Text = FormatNumber(Score, 0, -1, 0, -1) 'this Displays the points added in scoretext.Text box on the backdrop
End Sub

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

Note:Check the Script for AddScore (50) or any AddScore (#) Bumpers Targets & SlingShots now have AddScore. The AddScore (#) is the amount being added & displayed in the **TextBox**. Besides the green script these were also added.

Back to Top

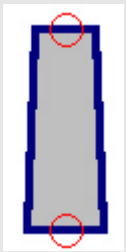
Editor Objects
Ramps

Ramp

Ramps are used to get the ball to a higher elevation level.

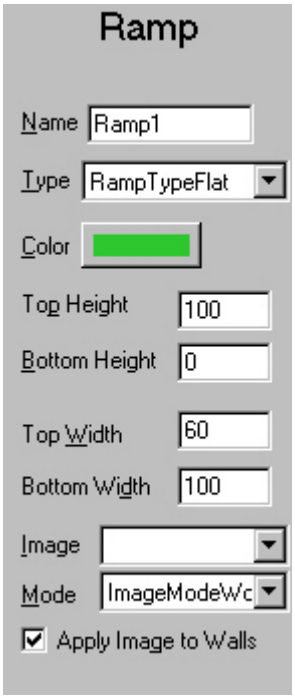
Ramps cannot move, or go up & down yet. Maybe in a later version of Visual Pinball

Ramp Button

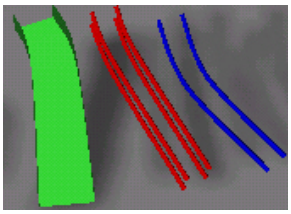
			
First Lets Create a Ramp: If you don't have a table already opened **File** / **New** to open a new Table.	Click on the Ramp button Move Cursor towards the table you are building.	it should turn into a Ramp cursor, click this where you would like your Ramp placed.	and you get a Ramp.

Now that we have a Ramp Created we can look at the Ramp Option Properties.

Click on options in the **ToolBar**  and select the Ramp by clicking on it. 
Doing this will show us the Ramp properties.



Ramp **Type**



Shows the 3 types of ramps to choose from. Flat, 4Wire, 2Wire.

Ramp - Name for type of object you have selected.

Name - *Ramp1* The name of the Ramp, you can change this to whatever you want. **Tip** I usually don't rename the **Ramp**.

Type - RampTypeFlat, RampType4Wire, RampType2Wire. See Picture below Ramp Options

Color - Change the color of the Ramp.

Top Height - Changes the Top Height of the **Ramp.** Default is 100

Bottom Height - Changes the Bottom Height of the **Ramp.** Default is 0

Top Width - Changes the Top Width of the **Ramp.** Default is 60

Bottom Width - Changes the Bottom Width of the **Ramp.** Default is 75

Image - Select an Image you have imported from the scroll down list.

Mode - See picsonramps.zip & click on the ramps & look at their options, you'll see both Modes used in the example.

ImageModeWorld - Is for using a whole table image in wich you have a ramp drawn on already. Gives you the image thats under the Ramp. Usually a 512 x1024 image.

ImageModeWrap - Will stretch an image to fill in the ramp. So you don't have to use a full table image.

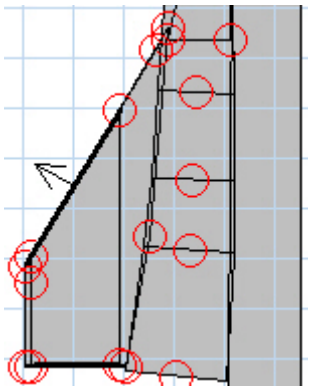
Apply Image to Walls - Puts the selected image on the walls of the Ramp also, if checked.

Note: You will need a graphic program to make Images. **File/Export/BluePrint...** After you save the Image you can open it in your graphics program. If you made a ramp & exported a blue print, the Ramp will show up on the BluePrint.bmp Open it up in your Graphic Program to edit it. You can draw over the Ramp part of BluePrint to make a ramp image. Export the whole image at 512 x 1024 & Import it Using **Table/Image Manager...** Then you can select it in the Ramp Options in the **Image** Scroll Down bar. This describes making Ramp graphics for ImageModeWorld.

Ramps are A little hard to work with. When you make one they come with a few problems, The Ball falls off the side easily so you need to add walls on the sides to help keep the ball on. The Ball can get stuck under or behind the bottom part of the ramp, so you may need to build a wall to keep the ball out of there.

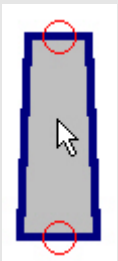
Using demo4rampspinner.zip

In the Demo you can see the walls I had to build to help keep the ball on the ramp.

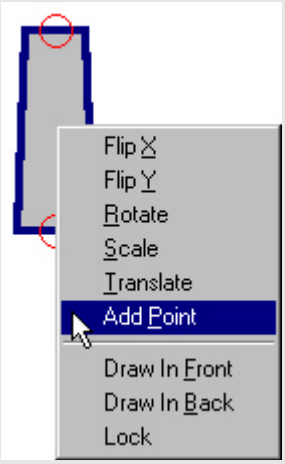


I built a Tunnel next to the ramp also, In the Editor take note to the Wall Options & look at the walls Top Heights & Bottom Heights surrounding the ramp. This will give you an idea of what is needed if you have Ramp troubles. Also Take a look at the Ramp, it starts with 1 Ramp & goes into another. Take a look at the Ramp Options & see how this is done.

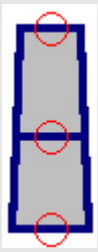
Building A Simple Ramp



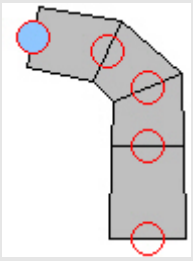
Make a Ramp, move cursor close to the spot you'd like to add a point.



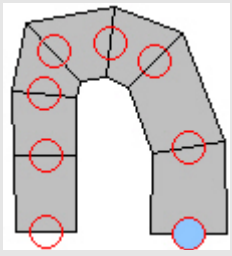
Right click & select add a point.



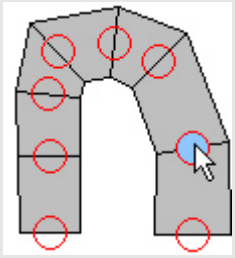
You now have a new point added to the ramp.



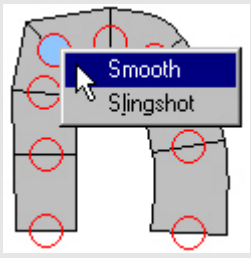
Add a few more points to the ramp



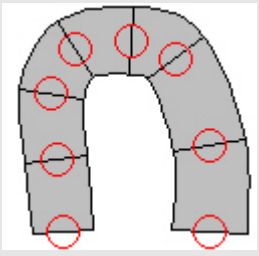
Keep adding points to make more spots to curve the Ramp.



Select all of the Points in the Ramp.



Right Click over each Point, & select smooth



And you get a Ramp thats is smoother.

Back to Top

Editor Objects
Spinners

Spinner

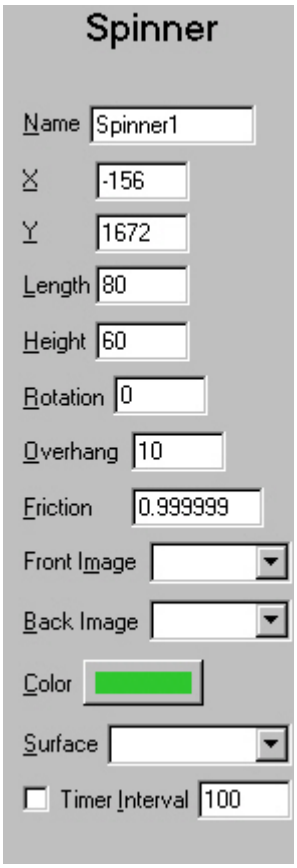
Spinners are an object that spins when the ball hits it.

Spinner Button

			
First Lets Create a Spinner: If you don't have a table already opened **File** / **New** to open a new Table.	Click on the Spinner button. Move Cursor towards the table you are building.	it should turn into a Spinner cursor, click this where you would like your Spinner placed.	And you get a Spinner.

Now that we have a Spinner Created we can look at the Spinner Option Properties.

Click on options in the **ToolBar**  and select the Spinner by clicking on it.
Doing this will show us the Spinner properties.



Spinner - Name for type of object you have selected.

Name - *Spinner1* The name of the Spinner, you can change this to whatever you want. **Tip** I usually don't rename the **Spinner**. The name is used to address the object through the script.

X - Shows The X position of the Spinner on the table.

Y - Shows The Y position of the Spinner on the table.

Lenght - Changes the Top Lenght of the **Spinner.** Default is 80

Height - Changes the Bottom Height of the **Spinner.** Default is 60

Rotation - Turns the Spinner's angle.

Overhang - Changes the lenght of the wire holding the **Spinner.**

Friction - Higher Number Tightens (Spin Less) or Lower Number Loosens(Spins More)the Spinner

Front Image - Select an image for the front of the spinner.

Back Image - Select an image for the back of the spinner.

Color - Change the color of the Spinner.

Surface - Selects a surface for the spinner to be placed on.

Timer Interval - If enabled will call SpinnerName_Timer() in the script .Unchecked is default.

Note: You will need a graphic program to make Images. I usually make Spinner Images 300 x 300

Spinner Script

Using demo4rampspinner.zip

Spinners are pretty simple to get working. In the above example you can open it up & check the spinner out, in action. The script for the Spinner is at the bottom of the script in this demo.

Note A spinner uses Spinner1_Spin() not Spinner1_Hit()

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

Back to Top

Editor Objects
Triggers

Trigger

Triggers are objects that are low to the surface & can be scripted for hit events.

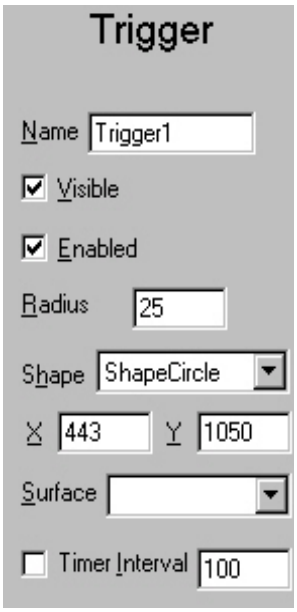
Trigger Button

			
First Lets Create a Trigger: If you don't have a table already opened **File** / **New** to open a new Table.	Click on the Trigger button. Move Cursor towards the table you are building.	it should turn into a Trigger cursor, click this where you would like your Trigger placed.	And you get a Trigger.

Now that we have a Trigger Created we can look at the Trigger Option Properties.

Click on options in the **ToolBar**  and select the Trigger by clicking on it. 
Doing this will show us the Trigger properties.

Note:All new Tables come with 4 Triggers already on them. They Are called LeftOutlane, RightOutlane, LeftInlane, RightInlane but they are not scripted to do anything yet.



Trigger- Name for type of object you have selected.

Name - *Trigger1* The name of the Trigger, you can change this to whatever you want. The name is used to address the object through the script.

Visible - Visible checked or unseen unchecked. Default is checked

Enabled - needs additional script. Default is checked

Radius - Changes the radius of the trigger. Bigger number makes it bigger. 25 is Default

Shape - ShapeCircle  & ShapeCustom  - Custom lets you add points & edit the Triggers shape.

X - Shows The X position of the Trigger on the table.

Y - Shows The Y position of the Trigger on the table.

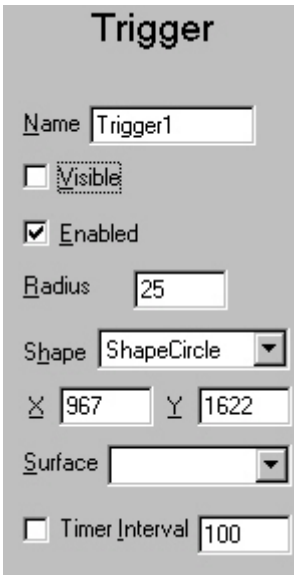
Surface - Selects a surface for the Trigger to be placed on.

Timer Interval - If enabled will call TriggerName_Timer() in the script .Unchecked is default.

Trigger Script

Using demo5triggersinlaneoutlane.zip

Triggers are pretty simple to get working. In the above example you can open it up & check the Triggers out, in action. The script for the Triggers is at the bottom of the script in this demo.



In Trigger Options, on Trigger1 I unchecked visible. To Hide the Trigger1 when Table is played.

Note: LeftOutlane, RightOutlane, LeftInlane, RightInlane have events now also.

```
'************************* Triggers

Sub Trigger1_Hit()'Inside this Sub is what the Trigger1 will do
Playsound "LaunchBall"
End Sub


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

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

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

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

'************************* Triggers Ends here
```

Back to Top

Editor Objects
Timers

Timer

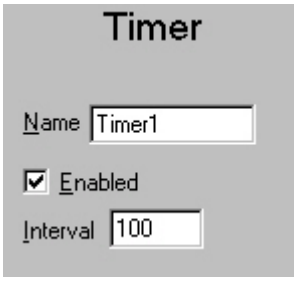
Timers can be used to add a pause after a Hit event.

Timer Button

			
First Lets Create a Timer: If you don't have a table already opened **File** / **New** to open a new Table.	Click on the Timer button. Move Cursor towards the table you are building.	It should turn into a Timer cursor, click this where you would like your Timer placed.	And you get a Timer.

Now that we have a Timer Created we can look at the Timer Option Properties.

Click on options in the **ToolBar**  and select the Timer by clicking on it. 
Doing this will show us the Timer properties.



Timer- Name for type of object you have selected.

Name - *Timer1* The name of the Timer, you can change this to whatever you want. The name is used to address the object through the script.

Enabled - needs additional script. Default is checked

Interval - Default is 100.

Note: Timers can be place anywhere, on Table or Backdrop.

Timer Script

Using demo6timer.zip

Timers are pretty simple to get working. In the above example you can open it up & check the Timers out, in action. The script for the Timers is at the bottom of the script in this demo.

Note: In the Timer Options, Enabled is unchecked, it is Enabled by dropping both targets.

In Sub CheckTarget Timer1 is enabled

```
Sub CheckTarget()
If Target.Isdropped = True And Target1.IsDropped = True Then 'checks to see if both targets are dropped
Timer1.Enabled = True ' if they are both dropped timer will start
End IF
End Sub
'**************************Target Stuff Ends Here
```

In Sub Timer1_Timer the Timer & Targets are reset, Timer is no longer enabled until both targets are knocked down again.

```
Sub Timer1_Timer() 'in timer options look at the interval, its 2500 2 1/2 seconds
Timer1.Enabled = False 'after 2 1/2 seconds Timer1 stops and finishes the Sub
PlaySound "targetup"
Target.IsDropped = False 'pops the target back up
Target1.Isdropped = False 'pops the target back up
End Sub
```

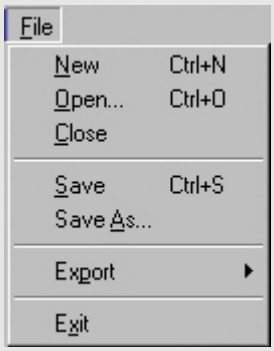
Back to Top

Editor Objects
Flippers

Flipper

Flippers are used to hit the ball. (In case you didn't know :))

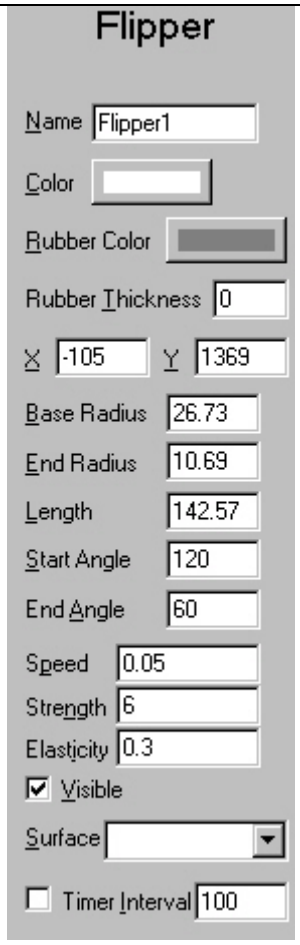
Flipper Button

			
First Lets Create a Flipper: If you don't have a table already opened **File** / **New** to open a new Table.	Click on the Flipper button. Move Cursor towards the table you are building.	It should turn into a Flipper cursor, click this where you would like your Flipper placed.	And you get a Flipper.

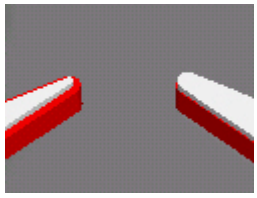
Now that we have a Flipper Created we can look at the Flipper Option Properties.

Click on options in the **ToolBar**  and select the Flipper by clicking on it. 
Doing this will show us the Flipper properties.

Note:All new Tables come with 2 Flippers already on them. LeftFlipper & RightFlipper.



Rubber Thickness



Left Flipper set to 7
Right Flipper set to 1

Flipper - Name for type of object you have selected.

Name- *Flipper1* The name of the Flipper, you can change this to whatever you want, but don't need to. Remember Flipper1 must be scripted for it to move. The name is used to address the object through the script.

Color - Select the color of the Flipper.

Rubber Color - Select the color of the Flipper's Rubber.

Rubber Thickness - Select The Rubbers Thickness. 0 is Default. See Picture below Flipper Options

X - Shows The X position of the Flipper on the table.

Y - Shows The Y position of the Flipper on the table.

Base Radius - The Thickness of the base of Flipper. Base of Flipper is usually made thicker.

End Radius - The Thickness of the end of Flipper. End of Flipper is usually made thinner.

Length - Changes the length of the Flipper.

Start Angle - The Starting point for Flipper to begin moving from. See **Angles** below.

End Angle - The Farthest point for the Flipper to move. See **Angles** below.

Speed - The time it will take the Flipper to move from Start Angle to End Angle

Strength - Determines how hard the Ball will be hit with the Flipper.

Elasticity - Gives the object more bounce or Elasticity. Can be seen when ball will is reflected off the object. Making this number higher gives it more bounce.

Visible - Visible checked or unseen unchecked. Default is checked

Surface - Selects a surface for the Flipper to be placed on.

Timer Interval - If enabled will call FlipperName_Timer() in the script .Unchecked is default.

Angles

0 being due north.
The angles are 0 to 359 technically though negatives appear to work as well.
Think of the directions as a clock face that goes

---------- 0
---------305-------45
--------270---------90
---------225-------135
----------180

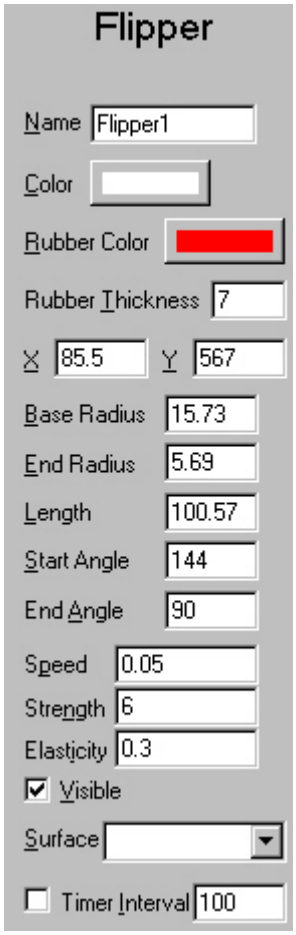
Flipper Script

Using demo7flipper.zip

Flippers are pretty simple to get working. In the above example you can open it up & check the Flipper1 that was added in. The script for the Flipper1 is at the top part of the script in this demo. In the KeyCodes section of script.

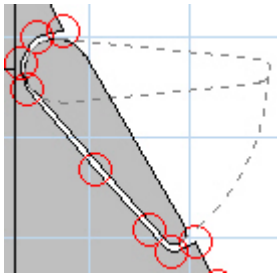
Note:The Green represents the part of the script from demo7, Only the Flipper1 script was added.

```
Sub Table1_KeyDown(ByVal keycode)

If keycode = LeftFlipperKey Then
LeftFlipper.RotateToEnd
Flipper1.RotateToEnd 'Adds To End Movement for Flipper1
PlaySound "FlipperUp"
End If

Sub Table1_KeyUp(ByVal keycode)

If keycode = LeftFlipperKey Then
LeftFlipper.RotateToStart
Flipper1.RotateToStart 'Adds To Start Movement for Flipper1
PlaySound "FlipperDown"
End If
```



Showing Flipper Options, for Flipper1

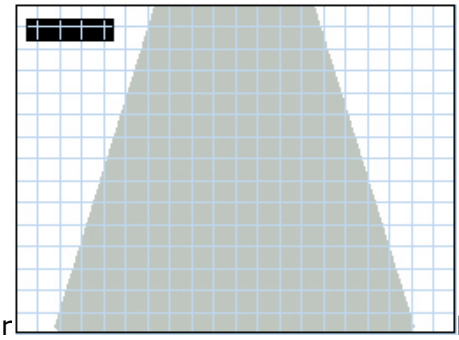
Notice the angles for getting it in this position, you may need to change them a few times on your own tables to get the to the exact position you want them in.



I edited the wall to make the flipper spot look more realistic by right clicking the wall behind Flipper & adding points & moving them.

Back to Top

Editor BackDrops Objects
EMReel

The Backdrop Tool Bar  Back Drop View of Editor

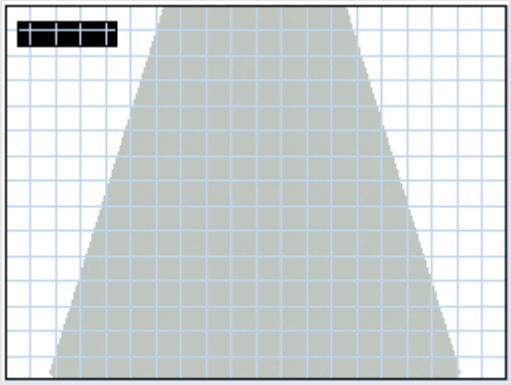
Note: Black (he made the EMReel option in VP) goes into detail on using this option. Check his page out! & look for his page on the EMReel. Most of the EMReel functions on this page were used from Blacks Page.

EmReel

EmReels are used to simulate an old Pinball machines way of displaying scores.

EmReelButton

			
First Lets Create a EmReel: If you don't have a table already opened **File** / **New** to open a new Table.	Click on the EmReel button. Move Cursor towards the Backdrop of the table you are building.	It should turn into a EmReel cursor, click this where you would like your EmReel placed.	And you get a EmReel.

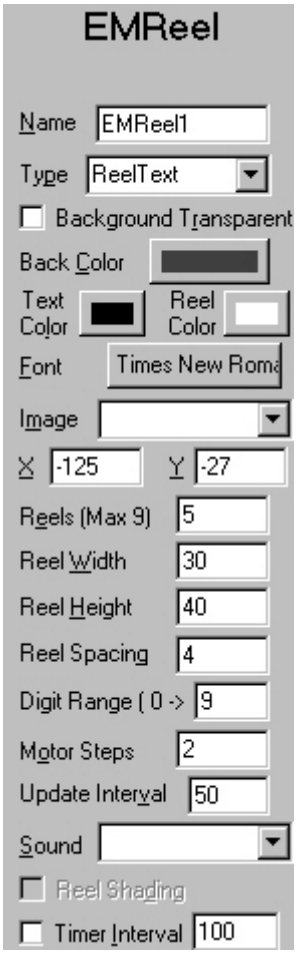


Note:It can be placed anywhere but, usually kept off the gray area which is the table.

Now that we have a EmReel Created we can look at the EmReel Option Properties.

Click on options in the **ToolBar** 

and select the EmReel by clicking on it.  Doing this will show us the EmReel properties.



EmReel - Name for type of object you have selected.

Name - *EMReel1* The name of the EMReel, you can change this to whatever you want. The name is used to address the object through the script.

Type - Choose from ReelText or ReelImage. ReelText can be changed from the **Font**. ReelImage uses the image scroll down box for an imported .bmp (fruits, deck of cards ect.)

Background Transparent - Checking the box Makes the **Background** invisible or Transparent. Unchecked makes the **Background** solid.

Back Color - Changes the Color surrounding the individual reels.

Text Color - Changes the color of the Text, if Reeltext is selected in **Type**

Reel Color - Change the color of the EMReel.

Font - Opens a box to select the Font, Font Style & Size, if Reeltext is selected.

Image - Select an Image to use from the image scroll down.

X - Indicates the **X** position of the EMReel on table.

Y - Indicates the **Y** position of the EMReel on table.

Reels (Max9) - Number of reel displays to have in the EMReel. 5 is Default. So a score of 99999 can be reached & the Reel will reset to 00000

Reel Width - Changes the Width of the **EMReel.**

Reel Height - Changes the Height of the **EMReel.**

Reel Spacing - Changes the Distance between the individual Reels.

Digit Range (0->9 - ReelText is always 9. Changes the range of digits for ReelImage. up to 16 can be used, but remember 0 counts as a digit so 15 is the Max. number. Ex. 0->15 It can also be less then 9 to simulate a slot machine.

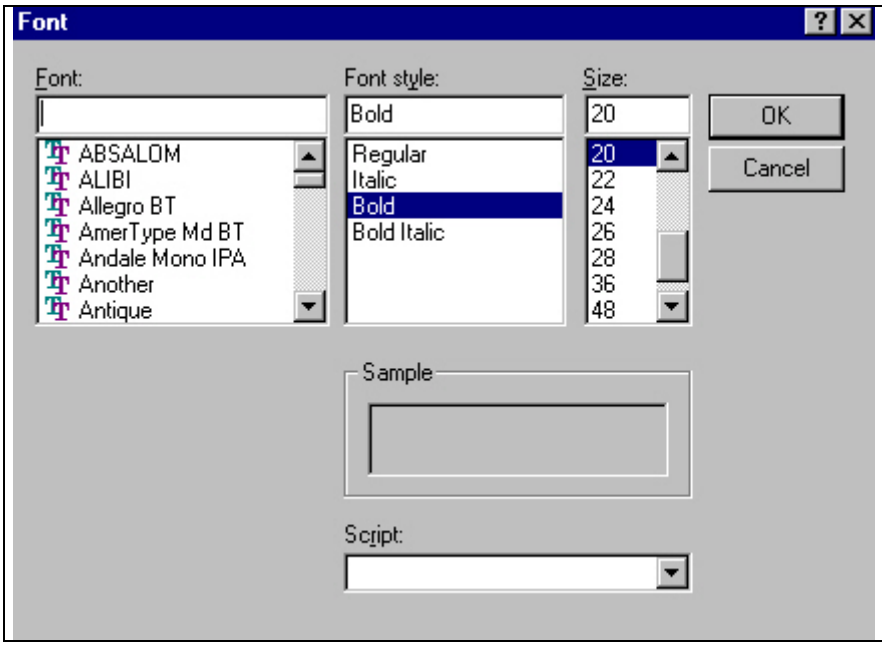
Motor Steps - Changes the speed the EMReel turns. The higher the number the smoother it will look. To high of a number may cause it to look jittery.

Update Interval - The time in milliseconds between each frame update.

Sound - Selects a sound to play while reel spins, from the Sound drop down list after you have imported one.

Reel Shading - Not Used yet

Timer Interval - If enabled will call EMReelName_Timer() in the script .Unchecked is default.



Selecting **Font** Opens up the Font Option's box. From here you can select the Font, Font Style & Size.

The **Sample** box shows a preview of the selected font.

Using demo8emreel.zip

Here is the basic script changes to go with the above Demo8.

Just Added an EMReel1.addvalue(#) wherever there was an Addscore. Has an EMReel2 also to demonstrate a graphic.

Example:
```
AddScore (10) 'Adds a Score of 10 when hit
EMReel1.addvalue(10)
```

How do I control the Reel in my Script?

There are 4 methods available for control of a EMReel in you table. These are..

.ResetToZero()

.AddValue(Value)

.SetValue (Value)

.SpinReel (ReelNumber, PulseCount)

To Find out more on EMReel's Check out Black's Pages below! He has Demo for it also.

Note: Black (he made the EMReel option in VP) goes into detail on using this option. Check his page out! & look for his page on the EMReel. Most of the EMReel functions on this page were used from Blacks Page.

Back to Top

Appendix A
Additional Keycode

Additional	Keycodes
0	= 11
1	= 2
2	= 3
3	= 4
4	= 5
5	= 6
6	= 7
7	= 8
8	= 9
9	= 10
A	= 30
ADD	= 78
APOSTROPHE	= 40
AT	= 145
AX	= 150
B	= 48
BACK	= 14
BACKSLASH	= 43
BACKSPACE	= 14
C	= 46
CALCULATOR	= 161
CAPSLOCK	= 58
COLON	= 146
COMMA	= 51
D	= 32
DECIMAL	= 83
DELETE	= 211
DIVIDE	= 181
DOWN	= 208
DOWNARROW	= 208
E	= 18
END	= 207
EQUALS	= 13
ESCAPE	= 1
F	= 33
F1	= 59
F2	= 60
F3	= 61
F4	= 62
F5	= 63
F6	= 64
F7	= 65
F8	= 66
F9	= 67
F10	= 68
F11	= 87
F12	= 88
F13	= 100
F14	= 101
F15	= 102
G	= 34
H	= 35
HOME	= 199
I	= 23
INSERT	= 210
J	= 36
K	= 37
L	= 38
LALT	= 56
LBRACKET	= 26
LCONTROL	= 29
LEFTARROW	= 203
LSHIFT	= 42
LWIN	= 219
M	= 50
MINUS	= 12
MULTIPLY	= 55
N	= 49
NUMLOCK	= 69
NUMPAD0	= 82
NUMPAD1	= 79
NUMPAD2	= 80
NUMPAD3	= 81
NUMPAD4	= 75
NUMPAD5	= 76
NUMPAD6	= 77
NUMPAD7	= 71
NUMPAD8	= 72
NUMPAD9	= 73
NUMPADCOMMA	= 179
NUMPADENTER	= 156
NUMPADEQUALS	= 141
NUMPADMINUS	= 74
NUMPADPERIOD	= 83
NUMPADPLUS	= 78
NUMPADSLASH	= 181
NUMPADSTAR	= 55
O	= 24
P	= 25
PAUSE	= 197
PERIOD	= 52
PGDN	= 209
PGUP	= 201
Q	= 16
R	= 19
RALT	= 184
RBRACKET	= 27
RCONTROL	= 157
RETURN	= 28
RIGHT	= 205
RIGHTARROW	= 205
RMENU	= 184
RSHIFT	= 54
RWIN	= 220
S	= 31
SCROLL	= 70
SEMICOLON	= 39
SLASH	= 53
SPACE	= 57
STOP	= 149
T	= 20
TAB	= 15
U	= 22
UNDERLINE	= 147
UPARROW	= 200
V	= 47
W	= 17
X	= 45
Y	= 21
Z	= 44

Back to Top

Appendix B
FAQ (Frequently Asked Questions)

This is mostly a collection of posts from Shiva Site.
This is just easier to use, but it came from Shiva. If your creating a table this could help ya out a lot. The biggest tip of all, keep messing with the program & ask for help at Shiva forum.

Testing Tips

Option explicit

Putting this at the top means you have to declare all variables. It prevents spelling errors.
Many people had mentioned it takes them way too long to test out their creations, so I thought I'd put this here. This will save time.
You can map any key to any subroutine in the KeyUp and KeyDown sections of the script. Or you can set up a routine whenever the shift keys or return/enter keys are pressed. Example:

```
If KeyCode=PlungerKey then
Kicker1.CreateBall
Kicker.Kick 0,40
End If
```

If you need to match a keycode with a key, add this line to the keydown routine.
Score.text=Keycode.
Anyway, whenever you press that key, it creates a ball and launches it onto the playfield. You then have practical ball-in-hand power for testing purposes. You can launch a ball in any direction, at any speed which is good for testing out ramps and loops.

You can also set up multiple kickers on the playfield. If (Scared Stiff example) to get coffin multiball, you need to hit the left ramp, then hit the left loop. Do that three times, and coffin multiball starts.

Now sure I could sit there playing the table, and play it for a long time to test that out, or I could use some kickers to do the entire sequence for me.
One kicker above the plunger, named Test1, one kicker in front of the left ramp named Test2, one kicker at the left ramp deposit to the right inlane named Test3, one kicker in front of the left loop named Test4. The script would be something like this.

```
Sub Test1_Hit()
Test1.DestroyBall
Test2.CreateBall
Test2.Kick 350,30
End Sub

Sub Test3_Hit()
Test3.DestroyBall
Test4.CreateBall
Test4.Kick 145,30
End Sub
```

Whenever a ball is launched, it can't miss kicker Test1, and when I'm finished it's easy to comment or delete the test script section and the kickers, or just move them off the table, or disable them

Add variables to this

```
dim f 'force of kick
dim a 'angle of the kick
f = 50
a = 350

Sub Test1_Hit()
Test1.DestroyBall
Test2.CreateBall
Test2.Kick a,f
f = f + 5 ' adds 5 to the force
if f > 90 then f = 50 ' resets the force to default
End Sub
```

You can change the angle the same way.

Back to Index

Shrinking File Size

Most of your size is probably in the graphics or sound. Makes your sounds 8 bit mono with a maximum sample rate of 11k. In most cases this is more than high enough quality, and can instantly cut the size of your table by 7/8th!.

As for graphics, I like to keep the main playfield graphics at 24 bit (while many argue that 8 or 16 bit depth suffices) but any decals or supplementary graphics I take down to the lowest bit depth that is suitable...such as a decal on a light... a 1 bit black and white is usually just as suitable as a higher bit depth image, and can reduce the file size big time. In Skateball, the playfield is 24 bit, but all the light decals are 1 bit, and the flashing 'plastics' are 8 bit. In Skyrocket all of the graphics were kept at 24 bit for clarity and maximization of resolution, but the sounds, all being 8 bit mono 11k, kept the file down to a reasonable size.

I find that if your playfield picture is clear enough, you can reduce it to 256 colours easily without any noticeable loss of quality. Since the bitmap is used as a texture map, VP will smooth out the pixels and it won't look any different if the original bitmap is of a high enough resolution. It makes editing and retouching the playfield graphics that much easier too. All my table conversions so far have used 256 colour playfields split into lower and upper levels; the only one causing any problems so far is 4 Million BC, due to a lower quality image to work from. But a bit of time spent with Photoshop to tighten up the wall edges a little should sort out any difficulties. :>

Back to Index

Graphics

Start your *original* playfield art (or any other for that matter) at a huge res.... my jetsons psd file is 1500x3000 as that is all I can do with my avail ram!

Then take that pic when it's ready to be imprted to VP and scale it down to 512x1024 (width aproximate) But! scale it down in seval steps.... 1500x 3000 - 1350xxx - 1100xxx - 900 etc til you arrive at 512x1024
Play at 1280 x 960 for proper alignment.

Default table Image size should be --- 512 by 1024

Table Measurment Used
Okay, so now that Wpcmame was nice enough to point out to me that I had the real world ball size wrong, here's the accurate (6% smaller) figures for sizing your WPC and WPC95 machines. Note that these are only for these era machines and that the System1-11 and EMs will be different.
The ball in VP is 50 units which equals 1.0625" or 27mm in the real world. That means that your VP to real world conversion ratios are:

1" = 47 units
1.85mm = 1 unit

(Note: not nearly as clean as my mis-figure, so if you like the roundness of the previous figures, you can think in terms of 1"=25mm=50units and then just scale down your board and all objects by .94 when you're done. Big, big, BIG thanks to Dorsola for pointing out that you can select all objects on a playfield and downsize them (other than circle radii), that's a **really** handy way to adjust a table.)
This means that the actual measurements of real world tables in VP units is:

WPC era VP Playfield Sizes:
Standard - 962x2190
Widebody - 1095x2190

The recommended size for a playfield pic remains unchanged because the ratio is still the same as before, so to have a playfield pic be as large as possible and still fit the playfield properly, use these sizes:

VP Playfield Pic Sizes:
Standard - 450x1024
Widebody - 512x1024

That's it for the basics, go wild from there and your ball will 'fit' to a blueprint from a manual very nicely. Below are some revised details for sizing of playfield objects, so read on if you're interested.

I'm still pretty unsure of how large flippers should be since it appears that VP is using the flipper with rubber for it's sizing, so my measurements didn't work out too well. Here's what I came up with when matching up to a blueprint of the flippers on a properly sized board: Flippers seem to fit the blueprint when made to be length 128, base radius 23.0 and end radius of 9.5. The secret to properly emulating Fliptronics-era flippers seems to be in higher speed and lower speed than the defaults. The figures of 0.11 speed and 2.1 strength that Dorsola and Metallik suggested in the TZ mods seem to work pretty well. For 80s and EM pins, the default 0.08 speed and 5.0 power work quite well.

Most rubber rings would be about 12 units (1/4") wide, and even though in the real world, they actually make contact with the upper part of the ball before the midpoint, Randy pointed out that the behavior of the ball when it hits a surface that isn't connect with it's midpoint is currently undefined in VP, it's a good idea to make sure your rubber rings cross the 25 unit area. I've been using a rubber height of 24 to 36 units and that seems to look pretty good in the game.

A standard post would be 50 units high and plastics are sitting about 60 units off the playfield.

Even though these figures do seem a bit odd, I have to say that they make a world of difference when laying out the playfield. My previous figures made the game play much better with my Indy500 table, but downsizing it by 6% to hit these figures was really cool and just nailed the 'feel' of the table!

Limit Size of Bitmaps in VP

I beleieve the problem is the actual image size since I've succesfully loaded 500x1000 images that had huge filesizes, yet have been unable to import a playfield with a similar file size, but larger image size.

The trick I've used to keep as much detail as possible is to start with an image of 1562x2135x32 @ 300dpi. If an image has to be resampled to get to that size, I make sure that any anti-aliasing is disabled, and continue not allowing any anti-aliasing throughout the edit. When that is scaled down to 96 dpi (which looks much better when rendered than 72 dpi) it becomes a perfect 500x1000. This downsampling is the first stage where I turn anti-aliasing ON. Using this method, the smaller resolution image loses very little detail from the original, and edges of images turn out beautifully smooth.

It's still not as good as it would be to use the full size image but it at least makes the downsizing much less "lossy".

Table Graphics

How does one go about making Sling-shot bumpers like in the older Pinball tables two walls on top of each other?

Set the high wall at 39 – 40

Set the second one 18-30 and make it a different color. Stretch the control points a few pixels outside and make it slingshot. Should do the trick.
Draw in Back really does have a purpose .

Using other stuff for moving graphics

You could use transparent text boxes too if you wanted - since the background has a graphic - not sure how that would work out for you.

For a ball image though, you'd place a kicker outside the table and create a ball in it - and with a timer you can change the image so it animates - using a few kickers you could get the explosion, and the bullet flying across the table.

Since droppable walls provide similar functionality, those have been popular up to this point. The new Monster Bash using balls for animation of dracula, and Indiana Jones redux uses them to simulate drop targets. The temple table uses custom images for the gems. Pagan Pinball uses a ball image for the bubbling cauldron. Besides those, I don't think much work or research has been done on it in the area of animatrion. (Rock Fantasy has nifty animation for the ball, but that's just for the balls themselves - not trying to animate other stuff)

Back to Index

Lights

What does the pattern property of a light do?

A pattern is just that, a pattern. "1\" represents on, and "0" represents off (of course if you were a computer, everything is represented that way)...so you have a light and you want it to blink "on,off,on,on,off,off,on,off" then the pattern would be "10110010", now you may ask why you'd want that, so consider that you have three lights that you want to blink in in sequence...the patterns would be:

```
light1.blinkpattern="100"
light2.blinkpattern="010"
light3.blinkpattern="001"
```

The "1" is on and the "0" is off, so you can easily see how the pattern will play-out.

As for interval, it is simply how long a timer lasts. an interval of 1000 lasts approximately 1 second (so 100 would be a tenth of a second etc...) In the case of our three lights above, if you gave them each an interval of 1000, then:

Light1 stays on for a second then goes off for two seconds
Light2 stays off for 1 second, comes on for 1 second, then goes off for 1 second.
Light3 stays off for 2 seconds, then comes on for 1 second

The pattern will repeat as long as the timer is active. It's cancelled with "...timerenabled=0" ("0" here is the same as false here)

Changing colors during game play
You can put two lights of different colors on top of each other. When you want to change colors, turn the old light off and the new light on in that order. The light you last changed the state of will remain visible on top.
Or you could use droppable walls.
In short, assign a variable to a group of objects (Lights) and/or their perimeters (events)? So that

```
Dim A
```

controls

```
Sub Spinner1_Spin()
A = A + 1
Light[A].State = LightStateOn
End Sub
```

In order to light Light1-10 (think Bonus)
I used brackets because parenthesis are confused with an array (that's what the error message says).
Sort of. You have to declare an array of controls first, like so: :

```
Dim A

Dim SpinnerLights(10)
Set SpinnerLights(1) = Light1
Set SpinnerLights(2) = Light2
Set SpinnerLights(3) = Light3
Set SpinnerLights(4) = Light4
'etc.

Sub Spinner1_Spin()
  A = A + 1
  If A > 10 Then A = 1
  Light(A).State = LightStateOn
End Sub
```

It's tedious if you have a lot of lights, but it works. See just about any table for an example. I think Top Speed (that comes with VP) uses this technique. That's where I learned it.
There are several ways to do it. The easiest one is to create a collection. Go to the collection manager and create a collection called "LightsCollection" and put all your lights in there. Then you can do "LightsCollection(1).State" etc.
The only drawback is that you can't change the order in the collection so put them in in the right sequence from the beginning.
If you don't want to use collections you can create an array in the script.

```
Dim LightsArray
LightsArray = Array(Light1, Light2,Light3)
```

On my table, I have a set of lights which light in sequence each time a particular bumper (of which there is more than one) is hit. When I programmed the bumpers, I put an "if" for every possible permutation . e.g.:

```
if x=1 then
light1.state=light.stateon
end if
if x=2 then
light2.state=lightstateon
```

etc.
I have 12 lights- and 5 bumpers- and I've copied and pasted the list for each bumper_hit subroutine. It works fine, but is there a way that you can program the script so that the application knows automatically what to do if x=N?
So, for example, input all of the possible states in once only, and point all of the bumper_hit subroutines at that subroutine.
I think what you need to do is make arrays of your lights and bumpers. That way you can index them. Define two arrays:

```
Dim LightArray(11)
Dim BumperArray(4)
```

Then in the table's init-method, make the arrays point at the objects:

```
Set LightArray(0)=Light1
Set LightArray(1)=Light2
....
Set LightArray(11)=Light12
```

And the same for the bumpers. Note that arrays start at zero, so an array with 12 elements has indices from 0 to 11.
Now you can call one bumper subroutine from all bumperX_hit, using the number of the bumper as a parameter if you need to distinguish them:

```
Sub Bumper1_Hit
BumperStuff(0) 'numbering the first bumper zero here
End sub
```

To set a light based on a your x (if x starts at 1):

```
LightArray(x-1).state=LightStateOn
```

Balls Speed
I want to write a formula (possibly polar coordinate orientated)
That uses a kicker to speed the ball up, but maintains the direction.
I'm not such a Math wizard (surely you're understating!)
Does anyone know a good formula for this?
It must look as if the ramp simply isn't that steep.
Just multiply all of .Vel's with the same constant, say 1.1 or 1.2.
When you preserve the direction, no polar coordinates thing is necessary.

```
boostFactor = 1.2
With ActiveBall
.VelX = .VelX * boostFactor
.VelY = .VelY * boostFactor
.VelZ = .VelZ * boostFactor
End With
```

P.S.
You don't need a kicker. A trigger would suffice for this purpose.

Back to Index

Ball Color

Changing Color
You must have the absolute latest Tech beta 4 and then do

```
Kicker.CreateBall.Color = x
```

Note that it does not set the colour of the ball, just the tint.

```
Plunger.CreateBall.Image = "YourBallImageNameHere"
```

```
Sub Kicker1_Hit()
ActiveBall.Color = RGB(255, 0, 0)
End Sub
```

The order of the numbers = same as color control in the Windows "Paint" program.
Seems to work with all Hits; Triggers too.

Back to Index

Decals

hmm...Does the wall drop? Can't put a decal on a droppable wall.
Also JPGs don't make very good decals - I don't like using them that way.
So check the surface property of your decal.

By check, he meant check it for something (look at it). If you select your decal in the editor and the options bar is showing you will see an option "surface". By selecting a wall from the drop down list, the decal is placed at the height of that wall.

A question, though. Since you weren't familiar with the surface property, did you really do a decal? It sounds to me that what you might have done is imported your picture, then set the image property under the options for your wall to that image. The image property for walls assumes an image the full size of the table, not part of it. Your picture would have been stretched to fit the whole table, and the part that would show on the wall would just be that portion of the stretched image where the wall happens to be located. That would explain why you're getting only blue fuzz.

So you have two options (I recommend option#1):
1) Create a blank image the same size as your playfield image (512x1024 or whatever) and paste your picture onto this in the location where the wall is, and use that for the image property of your wall.
2) Put a decal on the table with the decal button on the left of the screen (it should show as a blue square in the editor). In the options for the decal, make sure the "Type" option says DecalImage, set the Image property to your picture (the original, not the full table blank), and the Surface property to the name of the wall you want the decal on. Then make sure Sizing says ManualSize and adjust the Width and Height to make a rectangle that frames your wall.

Do decals not show up on a droppable wall?
Just making sure that I'm not going insane...
Is there any other way to get an image on a droppable wall besides making the image part of a whole-table image layer and setting the wall to display that full-table image?

You can place a decal on top of a droppable wall by selecting the wall from the "Surface" list in the Options for the decal. The only problem is that the decal doesn't drop with the wall, so it is left floating in air when the wall drops.
So yes, as far as I know if you want the image on top of a wall to drop with it, you have to use a full table image.

Decals on Ramps

I'm having a problem placing a decal on a ramp.
I have a decal that goes the entire length of an upward-sloping ramp. I've got the decal sized right and everything. However, only the bottom part of it shows on the ramp, like this.
What am I doing wrong?

on the decal's surface property - set it to the ramp. Currently it defaults to table level which the ramp covers up.
so you set the decal to the ramp height?
Make a wall as high as the ramp is, or 1 unit higher then and set the decal surface property to that wall.

Back to Index

Table Animation

Borrowing some ideas from Berimon Aghar Truesilver's Transformers table and several other tables that use the "plastics lighting" trick, I've whipped up a quick demo that should be of some interest to original table designers. This example shows a technique that can be employed to create animations on the playfield surface of a table. This example is of a very simple five-frame animation controlled by a timer, but the technique being used could be set off by an event trigger just as easily. I have also illustrated an apparently unintended consequence that I discovered working on this: bumpers that are invisible except for the shadow. Depending on your table, this could either be a useful idea, or something to watch out for.
The .VPT example I have created can be found : here (Tm_Test1). I didn't bother commenting the example, but it shouldn't be too difficult to understand. Let me know if you have questions. To adjust the animation speed, adjust the value of the timer to the left of the table.

Back to Index

Sound

Save as
Mp3-Mpeg layer 3
Wav-windows pcm

To Soft/Loud

Sometimes, the volume of a VPinMAME game is too loud or too soft for you tastes. There are two different ways of changing the volume which depends on what type of game it is.
If it's a Williams/Bally game using the DCS system, then you'll have to open the coin door (End key), and use the service keys:

7 8 9 0
Escape - + Enter in Menu
Srv. Crdt - + Enter otherwise

Don't enter the menu (if you did, just hit Escape (7) until you're out of the menu) -- just use the 8/9 keys to adjust the volume. If you set it to maximum, there is the possibility of distortion.

If the game is not Williams/Bally DCS, (i.e., every other game), you'll need to use the MAME menu -- click on the PinMAME window, hit the key beside the 1 (tilde on US keyboards), and you'll see a little text box shown on the DMD. Use the up/down arrow keys to pick the various settings, and use left/right to alter the volume. When you're finished, hit that key again (tilde) to clear the menu, and click on the VPinball Player window (the pinball table).

Help turning sound off in sub to play once
To stop a sound that's currently playing use:

```
StopSound <sound name>
```

I only want the sound to play once after the wall is dropped, but after the wall is dropped each time the ball hits a trigger the "boing" is played

```
sub checktriggers
if light1.state=lightstateoff and light2.state=lightstateoff and light3.state=lightstateoff then
```

```
block1.IsDropped = false
if block1.isdropped = false then
playsound "boing"
end if
end if
end sub
```

I tried lots of stuff but I'm out of ideas.
Here's the rest of the script that calls checktriggers.

```
Sub trigger1_Hit()
playsound "swoosh"
light1.state=lightstateoff
addscore(25)
checktriggers
end sub

Sub trigger2_Hit()
playsound "swoosh"
light2.state=lightstateoff
addscore(25)
checktriggers
end sub

Sub trigger3_Hit()
playsound "swoosh"
light3.state=lightstateoff
addscore(25)
checktriggers
end sub
```

I just gave your routines a really quicky glance but you don't need the second "if then" in your checktrigger routine.
Instead just make it

```
sub checktriggers()

if light1.state=lightstateoff and light2.state=lightstateoff and light3.state=lightstateoff then
block1.IsDropped = false
playsound "boing"
end if

end sub
```

Doing that the "boing" will only play if light1, 2 and 3 are off.
if that's not quite what you want (hard to read a segment out of context), then adding another variable should work, such as...

```
sub checktriggers()

if light1.state=lightstateoff and light2.state=lightstateoff and light3.state=lightstateoff then
block1.IsDropped = false
playit=true
end if
if playit=true
```

```
playsound "boing"
end if

end sub
```

you could include playit=false in each of your trigger_hit routines. to ensure that the sound only plays if all the lights are off AND the wall is dropped.
Don't forget to dim playit.

Sound Looping

Leave the sound as .wav and loop it via
`playsound "music" ,-1`
stop it using
`stopsound "music"`
that´s the only way for nice looping music right now

0 is infinate sound
LoopCount is optional... 0 is infinit, i think. 2, i think will play the sound twice... Volume, i don't know how that works, but it's optional too.

Anyway:
`PlaySound TheSound, 0`
that may play *TheSound* until you finish the table or you use StopSound TheSound.

Back to Index

Music

Save Mp3 in stereo or they play fast 160

Can Music Be Looped (uses to cut mp3 size down)

You can fake it reasonably well if you break your mp3s into three parts - intro, loops, ending. Each part is a separate file, perhaps music-intro.mp3, music-loop.mp3, and music-end.mp3.

When you play your music, you first play music-intro.mp3, then loop play music-loop.mp3 over and over again until something forces it to change to music-end.mp3, signalling the end of the song.

Try this after you have your PlayMusic"musicfile.mp3" in you code then have this....

```
Sub Table1_MusicDone()
PlayMusic"musicfile.mp3"
End Sub
```

Or you could take a look at my Psycho 1.3 and it has this in there for a reference.
Hope this helps.
I actually got the music working from the psycho script
but I wasnt sure what the
`Sub Table1_MusicDone()` was for
Table1 is the name of your table, it starts as Table1, but many people change it.
When the mp3 stops, it goes to this sub. So for example by placing the

```
PlayMusic "filename.mp3"
End Sub
```

there it would play filename.mp3 when done playing the other mp3.

To take it one step further which I did for my table, if you want to add different music to different things, like attract mode, in game play, high score, multiball etc, you can set a variable such as

```
DIM music
```

then after each event use

```
music = "filename.mp3"
PlayMusic "filename.mp3"
```

Where filename is the name of the file to play for that portion such as attract mode, multiball etc.

In the sub for endmusic, change PlayMusic "filename.mp3" to
PlayMusic music and it will then play the current music over and over when its done, but allows you to change which music to play when the music is done.
When you go to another mode, such as when the game ends, I just
use the EndMusic to stop whatever was playing, and then set music and PlayMusic to the new music, which again will repeat when done if you have the PlayMusic music in the endmusic sub.

You can even take this a step further which I havent yet had to do such as playing from a selection or rnd selection of music. You would just create a select for what sets the music variable, this is done at the point of the playmusic. The endmusic will automatically repeat that one since it will play the music set to the variable.
This would work well for example for mario teeshot if he wanted to change to mp3, or a similar reason where you want to play multiple choices of music or multiple different music one after another during for example in game play. Enjoy! My Super Android table uses this, it should be released very shortly, havent had much time until now to work on it again.
For TB4, here are some helpful music tips, from basic to more advanced:

To play a mp3 music file as background music, you would place the mp3 file in the /music sub directory, in the script you would call it using in the script:

`PlayMusic "android.mp3"` where android.mp3 is the name of the music file to play, and the .mp3 is needed.
This would play the file ONCE then stop.
To Stop the Music File in Mid Play, such as at a game over or ball end you want no music to play anymore, you use the
Command

```
EndMusic
```

You don't need to enter the song title, only that command using that case exactly. This stops the music but doesn’t hold
The place it was in the file- if you start this or another song using PlayMusic again it starts from the beginning.
To Repeat the exact same music file after the complete music file plays, simply add a subroutine in the script:

```
Sub Table1_MusicDone()
PlayMusic "android.mp3"
End Sub
```

Where android.mp3 is the exact same filename as the original. The above is if you use only one single music file total in the table to play once or repeating. If you want to play more then one music file in different
Parts of the table or multiple files one after another, see more advanced below.

More Advanced: How to play multiple music files, such as one music file that can repeat in attract mode, a different
One that can repeat in in game music, a different one that can repeat during an event like multiball etc...

This is NOT for playing multiple files one after another, this is for playing different files at different times, repeating the looping of the correct file when its done playing based on where you currently are in the board (otherwise you'd always play the wrong file when the current music file ends).

First you dim a variable, I use Song, so you add:
Dim Song
To the variables at the top of the table.
Next add the following sub routine:

```
Sub Table1_MusicDone()
PlayMusic Song
End Sub
```

Then, at ANY point to play whatever music you want for that period of time you would use:

```
EndMusic
Song = "android.mp3"
PlayMusic "android.mp3"
```

Where android.mp3 is the name of the file to play during that part of the table.
For example, I use the above during attract mode, so that it will now play android.mp3 during attract mode, and
Whenever it finishes it calls the sub Table1_MusicDone, and plays the correct android.mp3 again since I've set
Song to that filename as well.

When you go into the in game music, I use:

```
EndMusic
Song = "android1.mp3"
PlayMusic "android1.mp3"
```

Now it immediately stops any current music playing (this is why you can use this at any event also, as say you JUST hit
Multiball, you can stop the current music and immediately change the music to play and repeat).
Then starts the new song android1.mp3, and whenever the file stops it will replay the correct song, in this case android1
And NOT the song used in another mode such as android.mp3 which I was using in attract mode only. This is very flexible
As you can stop and play different songs at different times, and repeat the correct music.

This mode is NOT for playing one after another after another, or random song choices, changing Song = to another name will play that song after the first is done, but then would just repeat that 2nd song, it wouldn't allow a 3rd, 4th etc or even to go back to the 1st song during that same mode using these routines.

This could be taken further if you want to play multiple music files one after the other during the same event, such as playing 3 files one after another during in game play not related to event changing (which the above advanced routine does that), by setting up a routine within Table1_MusicDone that would step to another song based on what played before, then every time its called it should check what played before to decide what to play next. This is easy coding but I havent had the need to use, as all my music changes are related to event changes, such as playing an in game music then you get a multiball or other event reason to change the music, or going from in game to game over (and therefore attract music) which uses the above Advanced routine. If anyone uses this, feel free to add to this to give a code example for someone else.

How do I loop my tables background music?
Leave the sound as .wav and loop it via playsound "music" ,-1
stop it using
stopsound "music"
that´s the only way for nice looping music right now

Back to Index

Triggers

Can triggers destroy balls?
A trigger can not destroy a ball.

Can triggers play 3 sounds?
You could use timers.

Timers are a little bit hard to wrap your head around at first but get to know how to use them, and you can do anything.
The basic concept is this...

A timer is comprised of the timer itself, an event subroutine an interval setting, and a state (enabled true/false). When a timer is first activated(with "timername.enabled=true", it will wait for the assigned interval to pass before executing anything in it's subroutine. It will then repeat the commands in a loop until it is told to stop "timername.enabled=false". When it disabled "enabled=false" it will stop as soon as the current interval length expires. Ofcourse the interval length can be set on the fly, so it can be sped up or slowed down as required. Every object on your table has timer built in that you can make use of, the only difference being that the state is "timerenabled" true/false rather than "enabled" true/false as it is with an actual timer object.
Timers are best learned by playing with a few of them, so I suggest you start with a blank table and try a couple things before using them in the table that you're working on.
Most VPtables that use kickers use timers for sound and kick timing, so a quick look at any of these should help you out.

Like anything else though experimentation is the key.
Then a timer is exactly what you need...
set the timerinterval for the length of the sound (1000 = 1 second, so if the sound is 2.5 seconds, the interval would be 2500).

Three sounds with a space (first space 2.5 seconds, 2nd space 1 second) between could be achieved like this...

```
sub kicker1_hit()
playsound "1stsound"
c=0
kicker1.timerinterval=2500 (you could have this set in the option screen instead if it's never going to change)
kicker1.timerenabled=true

...insert anything you want to happen with the first sound here...

end sub
```

now you have to have a routine for your kicker's timer, remeber, nothing in this sub routine will execute until the interval has passed at least once...

```
sub kicker1_timer()
kicker1.timerinterval=1000 (changes the interval for the length of the second sound...this WON'T happen until the original 2.5 seconds have passed)
c=c+1 (count how many times the timer had looped)
if c=1 then (first loop through...actually the second, since only your 1st sound played during the first)
Playsound "2ndsound"

...insert anything you want to happen with the second sound here...

else (the timer has looped, cause "c" has changed value)
kicker1.timerenabled=false (shuts off the timer, so it doesn't repeat again)
Playsound "3nd sound" (your 2nd sound is already over because of the timer)

...insert whatever you want to happen with the third sound here...

end if

...insert anything you want to happen with EVERY timer cycle (play with some lights?)

end sub
```

Just remember, that everything else in your script is still running while the timer executes...since your timer would be for a kicker, there's not much to worry about, since the ball isn't moving around the table.
Using a few timers together can accomplish some complex tasks easily, but just make sure you always do a "timerenabled=false" for any timers you're not using, or any. If you don't want the timer to repeat more than once, make it the very first line in the timer routine.

Back to Index

Glass Height

How high of a ramp are we talking here? The default glass height in VP is 210. If your ramp is too tall, the ball could be hitting the top glass. You can change the glass height from the options for the table. Make sure it is at least 50 units above your tallest ramp.

Visual Pinball Corruption
I'm cross-posting this to the Pinmame Help messageboard as well because it is very important.
I had just downloaded Windows Media Encoder 7.1 and went to open a table after had installed the media encoder. I don't think that caused the trouble though.
The point is, eventually your Visual Pinball registry settings will become screwed up (provided you use your pc for more than just visual pinball LOL).

Rather than reformatting and reloading all your software, follow these steps to correct it:

1) Rerun the setup file for Visual Pinball -- it will give you the option to REPAIR Visual Pinball installation. Use that option.
2) Rerun VPinmame setup beta 3 and it will tell you the comm object is registered or successful.
3) copy the attached file to your windows\system directory, then open up a dos window, change to the windows\system directory and type:
REGSVR32 MSSTDFMT.DLL

That's all there is to it, and you're back up and running.

Back to Index

---

rnd() function

Either I have done this wrong or the rnd() function is a crock!!!
The user should be getting a random award. I just shot 70 shot and got ZILCH every damn time

```
if CastleShots > 40 then
dim mystery
mystery = rnd(10)
        if mystery >= 0 and mystery < 2 then
        Instructions.Text = Instructions.Text + "ZILCH"
        end if

        if mystery >= 2 and mystery < 3 then
        Instructions.Text = Instructions.Text + "20 MILLION"
        AddScore(20000000)
        end if

        if mystery >= 3 and mystery < 4 then
        Instructions.Text = Instructions.Text + "10 MILLION"
        AddScore(10000000)
        end if

        if mystery >= 4 and mystery < 4.3 then
        Instructions.Text = Instructions.Text + "100 MILLION"
        AddScore(100000000)
```

```
        end if

        if mystery >= 4.3 and mystery < 7 then
        Instructions.Text = Instructions.Text + "5 MILLION"
        AddScore(5000000)
        end if

        if mystery >= 7 and mystery =< 10 then
        Instructions.Text = Instructions.Text + "BONUS AWARDED"
        AddScore(Bonus)
        end if
end if
TextDisplay.Enabled=True
end sub
```

same line is in my music selector and it works perfectly for me and plays both of the options for BGM
You need this near the beginning of your script.

Randomize
Then to use RND for a number between 1 and 10, you'd use this.

```
variable=CINT(RND*(9)+1)
```

While this works `variable=CINT(RND*(9)+1)`,
is this not the same as `CINT(RND * 10)`?
This is what I would have put when programming in VB.
`CINT (RND*10)` allows for rounding - that would give you a possibility of getting 0

Nope, not the same. The first case selects a random number between 1 and 10. (Rnd*9 returns a random floating-point value between 0 and 9, not including 9.0. Then you add 1 to it and round it to the nearest integer).
The second case picks a random number between 0 and 10.
Another possibly less confusing way to write the first statement would be:

```
variable = 1 + CInt(Rnd*9)
```

Okay, that I under stand, but SDB's example went from >=0 to =<10, which as far as I can see is 0 to 10.
I don't consider myself a wonderful programmer, but that is how I interpreted the request.
Hey, if RND(10) works for you, go for it. I'd prefer (feel better about the script) if it was RND*10. I don't trust the parenthesis to work as desired.
For 0-10, just drop off the +1 on the line.

Back to Index

Things on Platforms above 0

How do I place a bumber, or gate, or for that matter anything on a platform not on level zero?
Create a wall at the height you want the bumper or gate and in the surface property box select the wall you want the bumper to sit on.
but be careful with kickers...hehe...unless they are on the actual surface you're using it might work sometimes, and not work other times....

Back to Index

Sling Shot Walls

How do you make a slingshot?
Select a control point, then right click on it. Two choices will pop up smooth and slingshot. Click on the word slingshot.

How do I make a slingshot?
Place a wall on the table. Shape the wall how you want your slingshot to be. Select the wall, and then select a control point with the right mouse button. Two choices will pop up smooth and slingshot. Click on the word slingshot. That will convert the wall into a slingshot.

Back to Index

Balls

Setting # of Balls

Declare a variable for it.

```
Dim CurrentBall
```

In your "drain_hit()" section, add one to this, and if it exceeds your maximum number of desired balls, you'll want to call your end of game routine.
You'd also want to set this variable to 1 when you start the table.
If you award extra balls, you'd need to subtract 1 from it before you check to see if the game is over...

Back to Index

Multi Balls

If it wasn't way past my non-existent bedtime, I'd write a longer reply, but here's a few basic guidelines:

-For a really basic multiball (e.g. one where all balls are put into play at the beginning of the MB, and all extras are drained when the MB ends) Keep a variable that keeps track of how many balls are currently on the playfield (I usually use nBallsInPlay for this purpose.) Ensure that every function that can create a ball increments this count, and every function that destroys a ball (or, depending on what you're doing, the Drain_Hit() event reduces this count. When this count reaches zero, you end the current ball and start a new one as appropriate.

-For what it sounds like you're trying to do (hold balls in kickers to be released when targets are hit) you'll want to keep two variables: Total balls on the playfield, and balls in play (which would be one except when a multiball is active.) In this case, I'll call thse two values nBallsInPlay and nActiveBalls. When a ball is first put into play, you would increment both counts by one (note that I'm not writing actual script here, and this is for illustration purposes:)

nBallsInPlay = nBallsInPlay + 1 = 1
nActiveBalls = nActiveBalls + 1 = 1

If this one ball in play drains, both counts would be reduced to zero, at which point the drain_hit() event would use an if statement to see that no balls are active on the playfield, end the ball with bonus (or whatever) and start a new one. But say that ball goes into a lock:

nBallsInPlay = 1
nActiveBalls = nActiveBalls - 1 = 0

since there are no active balls on the table, hitting the lock should trigger a CreateBall on the plunger, which would increment both counts:

nBallsInPlay = nBallsInPlay + 1 = 2
nActiveBalls = nActiveBalls + 1 = 1

Now we have one ball in the lock and one ball in play. If the ball in play drains, nActiveBalls gets reduced to zero again, and the ball ends. But if the trigger to release the ball is hit, this releases the second ball into play and increments the nActiveBalls value:

nBallsInPlay = 2
nActiveBalls = nActiveBalls + 1 = 2

When one of these balls is drained, both counts drop by one:

nBallsInPlay = nBallsInPlay - 1 = 1
nActiveBalls = nActiveBalls - 1 = 1

...Which puts back to a single active ball on the playfield. In short:

-Have the plunger (or any other item that creates balls and puts them in play) increment both counts;
-Have the drain (or any other item that destroys balls) reduce both counts;
-A ball that gets locked should reduce the active ball count, which should trigger a CreateBall event, thus incrementing both counts again.

If none of this makes sense, feel free to ask questions. Again, I'm writing this at 1:30 in the morning, so coherence isn't exactly my strong point right now.

But the major one is that I can't get it to kill the Multiball...the program recognizes the second ball (the Multiball) as a normal ball and when it goes down the drain even though I have the "If Multiballs > 0 Then Multiballs = Multiballs - 1" and the DestroyBall thing. Could I possibly put a trigger in front of the drain that when hit it would destroy the ball if you had a multiball?

Without seeing the script, this is sort of tough.
Make sure at the beginning of the script you have Dim Multiball.
Somewhere in your table init you have Multiball = 0, also have this when when you reset the table for a new ball.
I'm going to assume when you add a ball in multiball, you also added multiball=multiball +1 to whatever kicker or plunger created the ball?
Check spelling.
Before you have "if multiball>0 then " put multiball=multiball-1 then put your if statement.

You always want the program to know how many balls are in play on the table. Whenever a ball is locked, that needs to be known as well. Whenever a ball leaves play. For multiball, if you have 60-ball multiball (gasp) after 59 balls leave play, you know multiball has ended.

In your drain routine does it check to see ballcount?
If multiball then subtract one ball otherwise advance ball in play.... (?)

Back to Index

Ramps

Soon we will have transparent ramps...but until then, like I did on Scared Stiff, setting the ramp color to the background color will make them barely-visible near the back of the table. For the ramp leading from the coffin, that is actually using a wall at 55 units above the table. The base of the pinball diameter is 12.5 units, and as long as your wall is 12.5 units or wider, the ball will go over it just fine. The pinball is approximately 48 units from bottom to top or across in any direction. So based on this, provided you have a wall on each side of the bottom wall 'ramp' at about +25 and -25 horizontal distance from the ball center location, and -24 to -26 units from the top of the ball, this will hold it in place. The downside to using this which makes it so difficult is if the ramp is on an incline. Placing walls in layers is very time consuming.

For any ramp, transparency can be achived by placing a decal about 50 units abobe the top of the ramp. As long as a wall is not under the decal (between the ramp and the decal), the ramp will be hidden, and the ball will be drawn on TOP of the decal image -- thereby pulling off the effect successfully. This is partially portrayed over the left ramp entrance -- a decal is between the top of the ramp and the bottom two wires -- the bottom two wires are hidden by the decal while the top two remain visible.

12.5 is much more accurate than 13. But for ramps, you want the wall at +24 to +26 -- possibly higher to stop the ball from flying off the ramp if it does become airborn.

From the bottom of the pinball to the lowest possible point of contact with a wall which will impact ball movement, is 12.5 units from table level (0)

The pinball is 48 units high, and 48 units wide. Therefore, the center point of the ball would be approximately +24 to +26 from the current ramp height.

ie - you have a ramp going from 0 (table level) to 100. At the top of the ramp, a 'sidewall' at bottom height of 124 and top height of 126 will keep the ball on the ramp. Adjust the top height of the side wall based on how fast the pinball tends to go up the ramp and the incline of the ramp. This also prevents slow balls from falling off flat ramps too.

Back to Index

Gates

Today's lesson in VPINMAME scripting: **Solenoid Controlled Gates**

There are many ways to handle a solenoid controlled gate, but some ways are better than others. To understand how to best model such a beast, you must look at what the effect of it is. If a ball aproaches from the pass-thru side, it always goes through. If a ball approaches from the other side, it only goes through if the solenoid is Enabled.
Now since gates cannot be dropped or lifted, we must use a Wall object instead. But how will Visual Pinball know if a ball is approaching it? There needs to be a hidden Trigger next to it. And, since you need to know which side the ball is approaching from, there should be one on each side.

So we have our gate, a wall called SolGate1. And we have the inner and outer triggers, InnerTrigger and OuterTrigger. First, let's write OuterTrigger_Hit. This one is simple.

```
Sub OuterTrigger_Hit()
SolGate1.IsDropped=True
End Sub
```

This drops the wall whenever a ball approaches from the ouside. Now we have to write InnerTrigger_Hit. This one is trickier. (note some people may use a Timer instead of an inner Trigger, but this is a kludge that causes slow moving balls to bounce off, and really fast ones to bounce out.) What it needs to do is raise the wall, but only if the solenoid is disabled. So our preliminary version looks somewhat like this.

```
Sub InnerTrigger_Hit()
If Controller.Solenoid(sGate) then
SolGate1.IsDropped=True
Else
SolGate1.IsDropped=False
End If
End Sub
```

But we can do better. Notice that with either case, the state of IsDropped is assigned to the same state at the solenoid itself? This means we can dispense with the If statement entirely! So our new, optimized code is.

```
Sub InnerTrigger_Hit()
SolGate1.IsDropped=Controller.Solenoid(sGate)
End Sub
```

If the solenoid is on, the wall is lowered. If the solenoid is Off, the wall is raised. "But what about the solenoid handler?" I hear you ask. IT IS NOT NEEDED. The ONLY time Visual Pinball needs to know the state of that solenoid is when a ball approaches from the inner side, and our subroutine takes care of that. But if you really want to add the solenoid handler, it's very simple.

```
Sub GateSolenoid(Enabled)
SolGate1.Isdropped=Enabled
End Sub
```

This way, you will see the gates go up and down before a ball approaches.
So to recap, here's our code.

```
Sub OuterTrigger_Hit()
SolGate1.IsDropped=True
End Sub

Sub InnerTrigger_Hit()
SolGate1.IsDropped=Controller.Solenoid(sGate)
End Sub
```

And, optionally.

```
Sub GateSolenoid(Enabled)
SolGate1.Isdropped=Enabled
End Sub
```

Yup, each subroutine is a one-liner, and NO timers are used. You can't write better code than this. Every situation that can come up is handled properly by the first two one line subroutines. If two balls approach, they will either bounce off the wall, or bounce off each other. Which is the case DOES NOT MATTER because the effect is the same.
When proper liftable gates are implemented in VP, you can remove the triggers, and their associated hit routines, replace the wall with a gate, and just use the solenoid handler to drop or raise the gate.

Back to Index

Bumpers

What is the height of the bumper object in VP?
The height of the bottom part of the bumper is 40. The top section of the bumper (the overhang) is a height of 20, making a total height of 60.

How do you get your bumper lights to stay on?
If you set the bumper to LightStateOff in the UI, then when you set it to LightStateOn in the script, it will actually work. Some kinda bug. But not as annoying as the "rename a new item twice = crash" bug. I lost quite a bit of work at least once because of that. Now I save every 5 minutes.
Or add a light on top of bumper

Bumper heights
I'm making sea wolf and i need to build a bumper out of walls so it can drop. What is the height of the base of the default bumper and the height top and bottom of the overhang part??

In STTNG I had to put a ramp over a bumper...the ramp was at 65 units. That's how high a bumper must be (64 or so). No idea how much room is between that and the overhang.
The height of the base and bottom of the overhang should be the same, 40. The top of the overhang is at a height of 60.

Back to Index

Targets/Walls

& messing with walls, if you don't want a ball popping off the table because of a low wall make it 13 numbers higher then the wall or table its on.
if your table or wall(if wall is a base)bottom height is 0 then the shortest height is 13

ex.
bottom height 0
top height 13
ex.
bottom height 3
top height 16

you can make it higher than 13 but any lower & the ball goes right over it.
From the bottom of the pinball to the lowest possible point of contact with a wall which will impact ball movement, is 12.5 units from table level (0).
The pinball is 48 units high, and 48 units wide. Therefore, the center point of the ball would be approximately +24 to +26 from the current ramp height.
ie - you have a ramp going from 0 (table level) to 100. At the top of the ramp, a 'sidewall' at bottom height of 124 and top height of 126 will keep the ball on the ramp. Adjust the top height of the side wall based on how fast the pinball tends to go up the ramp and the incline of the ramp. This also prevents slow balls from falling off flat ramps too.
If I have target1 target2 and target3, when I hit these targets they go down and increase score, however on the next ball thay are still down.
I know I need to reset the table or the targets somehow, but I cannot figure it out. I have checked some other scripts but I must be doing something wrong.
For each target:

```
Sub Target1_Hit
Target1.IsDropped = True
Timer1.Enabled = True
AddScore
End Sub
```

Then, after all three target scripts:

```
Sub Timer1_Timer()
Timer1.Enabled = False
CheckTargets
End Sub

Sub CheckTargets
If Target1.IsDropped And Target2.IsDropped And Target3.IsDropped Then
Target1.IsDropped = False
Target2.IsDropped = False
Target3.IsDropped = False
End If
End Sub
```

if you only want the drop targets to come up after the ball drains or whenever the ball drains put the TargetX.isdropped=false in your Drain_Hit area. (X being the number for your target)
Just put all objects that need to be reseted for next ball in a sub called for example

```
sub Next_Ball_Init()
Plunger.CreateBall
Target1.IsDropped = False
Target2.IsDropped = False
Target3.IsDropped = False
......
end sub
```

and call it in Drain_hit():

```
Sub Drain_Hit()
If GameInProgress then
Next_Ball_Init
Else
GameOverText.Text="Game Over"
End if
```

Back to Index

Collection of targets- VB5

```
Sub TargetCollective_Hit(Index)

' This is triggered when any target in the collective is hit.
' The index starts at zero. (Ex. the third target is index #2)
' If target #2 is hit then the index number equals 1, it drops
' target #2 and plays the sound "Drop Target"

TargetCollective(index).IsDropped=true
PlaySound "DropTarget"

' Now we check to see if all targets in the collective have been dropped.
' If so, we play a "TargetsAllDropped" sound and reset the targets.
' If at any time during this loop that
' TargetCollective.IsDropped not true then we exit the whole subroutine

For Each Target In TargetCollective
If Target.IsDropped <> true Then Exit Sub
Next

'All Targets have been dropped. Play "TargetsAllDropped" and
'reset all targets back to their original state.

PlaySound "TargetsAllDropped"

For Each Target In TargetCollective
Target.IsDropped = False
Next
End Sub
```

Back to Index

Score

Getting scoring to work
First, you need a sub routine for any object that has a hit event and score addition.
For example:

```
sub drop_target_hit() (where "drop_target" is the variable name assigned to that object)
drop_target.isdropped = true
score = score +100 ("score" is the variable you've assigned to the score)
display_score ("display_score" is the name of a sub routine you've written to display the
score)
end sub

sub display_score()
scorebox.text = score (scorebox is the name of the text object you've created)
end sub
```

Make sure you have include a dim statement for any variable (ie. Dim score) that will hold a value and that that statement will not be read again while the variable is being assigned new values - usually the opening lines in your script.

How do you reset score when game is over?
I reset score.text back to 0 when last ball is drained, but when I hit the first scoring target in the next game it resumes with the score from the previous game.
Also just curious, how hard is it to add 2 player mode?

Obviously, the variable you are using for score is not being reset to zero at the end of a game. This isn't automatic; you'll have to insert a '(score variable)=0' line in the required place of your code. If you *do* have one in, then it must be in a place which isn't being accessed at the required point in the game, and you'll need to move it. Put it in the same place as the code that sets up the first ball in a new game.

Score How to

first, you need a sub routine for any object that has a hit event and score addition. for example: sub drop_target_hit() (where "drop_target" is the variable name assigned to that object) drop_target.isdropped = true score = score +100 ("score" is the variable you've assigned to the score) display_score ("display_score" is the name of a sub routine you've written to display the score) end sub sub display_score() scorebox.text = score (scorebox is the name of the text object you've created) end sub Make sure you have include a dim statement for any variable (ie. Dim score) that will hold a value and that that statement will not be read again while the variable is being assigned new values - usually the opening lines in your script.

Reset score after game over
I reset score.text back to 0 when last ball is drained, but when I hit the first scoring target in the next game it resumes with the score from the previous game. Also just curious, how hard is it to add 2 player mode?Obviously, the variable you are using for score is not being reset to zero at the end of a game. This isn't automatic; you'll have to insert a '(score variable)=0' line in the required place of your code. If you *do* have one in, then it must be in a place which isn't being accessed at the required point in the game, and you'll need to move it. Put it in the same place as the code that sets up the first ball in a new game.

Bonus Scoring
"I want to have the bonus adding points runing slow/donw and see the score growing by (for ex) 1000 points"
The only way I could manage it was to have a seperate timer for each bonus level, and give

each timer a slightly longer interval. Then call all the timers in one sub. They will all trigger at the same time, but with different intervals, they appear to exicute one after the other.

I had the same problem in my Stingray Table. This is how I solved it. Create a timer that has the time delay that you want.

For example, for 1 second set the timer to 1000. Be sure to uncheck the enabled box when creating the timer, otherwise the timer will run as soon as the table is initialized. In the script where you want the countdown to occur call the timer by enabling it.

```
Sub BonusScoring
BonusTimer.Enabled = True
End Sub
```

Then do all of the scoring inside the BonusTimer Routine. The routine will run over and over again with a delay of 1 second (or whatever you set the timer to) until the timer is set back to false.

```
Sub BonusTimer_Timer()
If bonuscount > 0 Then
Playsound "Bonus"
AddScore = 1000
bonuscount = bonuscount - 1
Else
Bonus.Timer.Enabled = False
End If
End Sub
```

In the sxample above there is a counter called bonuscount that has some value. If the value coming in is 5 then when the timer is called then the timer routine will run 5 times playing a bonus sound and scoring 1000 points each time with a 1 second delay between each event. The next time through the bonuscount counter will equal 0 and the BonusTimer will be Disabled and the script will continue running from where the BonusTimer was initially Enabled.

Okay.. I've tried and tried to do this.. and cannot manage to create a delay when scoring my end of ball bonus..
my variable fh_xTotalBonus holds the total ball bonus value.. I want to decrease this by 1000, add the points to the Score, and repeat until it hits 0, and pause 500 ms (1/2 second) between repeats.
I "tried" using what jlang suggested.. and I must be doing something wrong. I've tried looking at code in other tables.. and they all have hundreds of lines of code concerning the bonus counts.. I'd think what I want to do is pretty simple.
Here's what I had.. I am sure it will be glaringly obvious how wrong it is.
Here is the Award Bonus Routine:

```
Sub AwardEndofBallBonus()
fh_xTotalBonus = fh_xRunningBonus * fh_xMultiplier
Timer1.Enabled = True
bo_AddPoint 1000
AlertBox.Text = "Ball Bonus " & fh_xTotalBonus
End Sub
And the timer timer is here...
Sub Timer1_Timer()
If fh_xTotalBonus > 0 Then
rem Playsound "Bonus" -- IT'S NOT PLAYING SOUND!!!
AddScore = 1000
```

```
fh_xTotalBonus = fh_xTotalBonus - 1000
Else
Timer1.Enabled = False
End If
End Sub
```

What exactly am I doing incorrectly?

just kidding... ok, you read this far. Your variables must make sense to you. Hopefully ALL your bonus additions are in 1000 flat increments or this script will have problems.

As long as the bonus addition is correct, to the addscore routine when you test it, you might want to increase the timer delay just for testing purposes, and have it write to a textbox after each bonus addition in the timer subroutine so you can see what it's doing. If you are 1000 points off on your bonus (which it looks like you are because of the >0 qualifier, put a second If statement in there to say like if it =0 then addscore 1000 and disable the timer. That way your other variables can't drop into negative numbers

High Score Reset
Open vpmkeys.vbs with notepad, and look for the follwing line:

```
keyHiScoreReset = 7 '(6) Reset Hiscores
```

Just set 7 blank or change to the desired AscII number for the key wanting to use. Remember to save the file on exit...

Back to Index

Bonus: Counters

I want to have the bonus adding points runing slow/donw and see the score growing by (for ex) 1000 points

The only way I could manage it was to have a seperate timer for each bonus level, and give each timer a slightly longer interval. Then call all the timers in one sub. They will all trigger at the same time, but with different intervals, they appear to exicute one after the other.

I had the same problem in my Stingray Table. This is how I solved it. Create a timer that has the time delay that you want.

For example, for 1 second set the timer to 1000. Be sure to uncheck the enabled box when creating the timer, otherwise the timer will run as soon as the table is initialized. In the script where you want the countdown to occur call the timer by enabling it.

```
Sub BonusScoring
BonusTimer.Enabled = True
End Sub
```

Then do all of the scoring inside the BonusTimer Routine. The routine will run over and over again with a delay of 1 second (or whatever you set the timer to) until the timer is set back to false.

```
Sub BonusTimer_Timer()
If bonuscount > 0 Then
Playsound "Bonus"
```

```
AddScore = 1000
bonuscount = bonuscount - 1
Else
Bonus.Timer.Enabled = False
End If
End Sub
```

In the sxample above there is a counter called bonuscount that has some value. If the value coming in is 5 then when the timer is called then the timer routine will run 5 times playing a bonus sound and scoring 1000 points each time with a 1 second delay between each event. The next time through the bonuscount counter will equal 0 and the BonusTimer will be Disabled and the script will continue running from where the BonusTimer was initially Enabled.

Okay.. I've tried and tried to do this.. and cannot manage to create a delay when scoring my end of ball bonus..
My variable fh_xTotalBonus holds the total ball bonus value.. I want to decrease this by 1000, add the points to the Score, and repeat until it hits 0, and pause 500 ms (1/2 second) between repeats.

I "tried" using what jlang suggested.. and I must be doing something wrong. I've tried looking at code in other tables.. and they all have hundreds of lines of code concerning the bonus counts.. I'd think what I want to do is pretty simple.

Here's what I had.. I am sure it will be glaringly obvious how wrong it is. Here is the Award Bonus Routine:

```
Sub AwardEndofBallBonus()
fh_xTotalBonus = fh_xRunningBonus * fh_xMultiplier
Timer1.Enabled = True
bo_AddPoint 1000
AlertBox.Text = "Ball Bonus " & fh_xTotalBonus
End Sub
```

And the timer timer is here...

```
Sub Timer1_Timer()
If fh_xTotalBonus > 0 Then
rem Playsound "Bonus" -- IT'S NOT PLAYING SOUND!!!
AddScore = 1000
fh_xTotalBonus = fh_xTotalBonus - 1000
Else
Timer1.Enabled = False
End If
End Sub
```

What exactly am I doing incorrectly?

just kidding... ok, you read this far. Your variables must make sense to you. Hopefully ALL your bonus additions are in 1000 flat increments or this script will have problems. As long as the bonus addition is correct, to the addscore routine when you test it, you might want to increase the timer delay just for testing purposes, and have it write to a textbox after each bonus addition in the timer subroutine so you can see what it's doing. If you are 1000 points off on your bonus (which it looks like you are because of the >0 qualifier, put a second If statement in there to say like if it =0 then addscore 1000 and disable the timer. That way your other variables can't drop into negative numbers.

Bonus(0) = Whatever bonus you're using *remember to dim Bonus so that Bonus exists either

Dim Bonus (4) If you use it for a 4 player game

or

Bonus = Array (0,0,0,0,0)

Of course you could replace Bonus(0) and then this line

CollectBonus.Enabled=True

So for example

Bonus(0)=Bonus(CurrentPlayer)
CollectBonus.Enabled=True

I took what you'd posted before.. (after I posted) and played with it and it DOES count down the bonus..

Here's the problem I have.. it's still counting the bonus when the next ball gets sent into play. ::chuckle:: Any way to totally stop advancement table action when the timer is running.. instead of the timer running while the subroutines after it continue to be ran by the system? The new ball shouldn't be able to come into play til after the bonus is done.

I sat up the delay to eject a new ball into play.. but still at times it's not long enough. Is this what I have to do, or is there a way to totally stop the routines from executing at the same time as that timer?

Here's what I would do if I understand you problem right. Instead of having a separate timer for the ball, have the ball be ejected in the same timer as the bonus. Whatever your variable for bonus is (lets call it bonus), say something like this in the bonus script part:

Put this at the begining, so it won't still try to take points away from bonus and make it negative where it won't be zero, and the ball won't fire.

```
if bonus = 0 then
plunger.createball
bonustimer.enabled = false
...
(other stuff that needs to happen)
end if
```

My table is working well but I need to know how to set a bonus multiplier for a kicker ,adding say 200 points after the score reached sAY 1000 ,and 500 after 3000,Would I also add a percentage of the Total Score?
Im doing an older pinball so the scoring would not be too high..any info helpfull..

```
Sub Kicker_Hit
If nScore > 1000 then
addBonus 500
ElseIf nScore > 500 then
addBonus 300
Else
addBonus 100
End If
```

End-of-ball bonuses
-First, it'll save you a lot of headaches if you make the bonus lights an array or collection. In fact, I don't know how you'd do it without an array, and I've never bothered figuring it out.

-The way I do an end-of-ball bonus usually involves three subs: AdvanceBonus, SetBonusLightState and AwardBonus. Their respective purposes should be pretty obvious.

-For the sake of this example, I'll stick to a bonus done in a simple, EM-like fashion (Max bonus 10,000, 10 lights. In fact, I'm going to cut and paste this example with some minor modifications from my script for Nip-It.) Whenever a target that advances the bonus is hit, it would call the AdvanceBonus sub. This sub would look something like this, with dims added to show what kind of variables it takes:

code:

```
dim nBonus
dim naBonusLights(10) 'the ten bonus lights are mapped to this array
dim nBonusMult 'if you have a multiplier

Sub AdvanceBonus()
  If nBonus < 10 then
    nBonus = nBonus + 1
    SetBonusLightState
  Else
    Exit Sub
  End If
End Sub

Sub SetBonusLightState()
  For a = 1 to 10
    If nBonus >= a then
      aBonusLights(a).State = LightStateOn
    Else
      aBonusLights(a).State = LightStateOff
    End If
  Next
End Sub
```

The AdvanceBonus should be pretty self-explanitory. The AdvanceBonusLightState() is a quick check that turns on all lights that are less than or equal to the current nBonus value. I will admit that this is a slightly "lazy" way to handle the bonus, but it should do the job in a situation where execution speed of the script isn't an issue, as is the case with VP.

As for the AwardBonus() sub, it would be called by the end-of-ball or any other sub that would award the bonus. The real work in this one is handled by a timer called BonusTimer, and the actual AwardBonus() sub is just a convenient way to activate it. Here's what it would look like:

code:

```
Sub AwardBonus()
  BonusTimer.Enabled = true
End Sub

Sub BonusTimer_Timer()
  If nBonus = 0 then
    BonusTimer.Enabled = false
    NewBall ' a call to the sub that starts the next ball
    Exit Sub
  Else
    Score 1000 'a call to whatever score function is being used
    aBonusLights(nBonus).State = LightStateOff
    nBonus = nBonus - 1
  End If
End Sub
```

Set BonusTimer's interval to be what you want the delay between bonus counts to be. This BonusTimer function will behave how you'd expect a simple EM-style bonus countdown to act. You can also add whatever sounds you need to, or modify it to handle more complex bonus counts.

Back to Index

Tilt

I want to either weaken or disable (preferably) the slam tilt on a certain System 11 William's table. I can't find code for this in the script. How do you do this?

Slam tilt is just a switch which is either on or off.
If the table uses the s11.vbs the normal tilt is simulated and be controlled with

```
vpmNudge.Sensitivity = x
```

Where x is 0-10

While x can be from 0 - 10 (0 = less sensitive, 10 = most sensitive), if you set it a negative number, you can effectively disable slam tilt (unless you *really* bang back it a lot!).
E.g., if you want to see how to get bored easily, set it to -100 or so...

Back to Index

Timers

Whats the correct way to make a message text only appear for 15 secs?

Set a timer for 15000 milliseconds. When the text is put into the messagebox, enable the timer. When the timer expires, disable the timer and write "" to the messagebox. three options.
When renaming objects, after chaning the name, press enter, select something else, press tab...etc. the name doesn't always change -- small vp bug.
2) save the table and reload the table. this sometimes works.
3) for timers, it makes no difference where they are on the table -- so just delete the old timer and make a new one.

Remember, if you changed the timer's name to MessageTimer, first I wouldn't recomment using spaces in names if you can avoid it. Anywayz, the script would use Sub MessageTimer_Timer, and you would use MessageTimer.Enabled and MessageTimer.disabled. If you named it Message, it's easier to remember (for me anyway) -- that way it would be Message_Timer, and Message.enabled/Message.disabled

Short form
Change the name of your timer with no spaces.
use Exactly what you renamed it to in your script.

Change every instance of the old timername to the new one.
That's all there is to it.
(edit/find at the top will save you hours of time -- F3 moves ahead to the next occurance)

Anyone know how to cause a delay between the drain and the plunger?

```
Sub Drain_Hit()
Drain.DestroyBall
Drain.TimerEnabled=True
End Sub

Sub Drain_Timer()
Drain.TimerEnabled=False
Plunger.CreateBall
PlaySound "Plunger"
End Sub
```

Adjust the Timer Interval on the dran kicker to change the length of the delay.

Timer For A shot-check out breakshot
Here is one way, first place a timer called "Countdown" on the table, uncheck Enabled and set the interval to 1000 (1 second)

Add this line to set up a var

```
Dim count
```

When you want to fire the timer add this in the script wherever you want
count=60 'count down 1 minute (or any amount you want)
Countdown.Enabled=True 'start timer

```
Sub Countdown_Timer()
If count > 0 Then
ScoreText.Text="TimeLeft = "&count
Count=count-1
Else
Countdown.Enabled=False
ScoreText.Text="Finished"
End If
End Sub
```

Create a timer, uncheck the enabled box, and set the interval to be 1000. At the beginning of the script, put this in:-

```
Public featuretime
```

This allows the variable "featuretime" to be used between subs. When you want the timer kicking in, put these lines in the sub (where timer1 appears, replace with whatever you call the timer):-

```
timer1.enabled = true 'this starts the timer
featuretime = 60 'change this to how many seconds you want the feature to go on for
```

Somewhere else in the script, create a sub like this:-

```
Sub timer1_timer()
featuretime = featuretime - 1
if featuretime >= 0 then
```

```
timer1.enabled = false 'this disables the timer
'before this comment, enter any variables you want changing relating to the feature, ie.
dropping off walls, removing targets, changing score etc.
End If
End Sub
```

Back to Index

Text

I am having trouble scripting a routine which will use a Sub to write different messages on a textbox. IE - I'd like the textbox to handle messages, as in:
"Press Start"
"Activated"
and so on...

I am trying to use a variable, but either I did it wrong, or VP will just not recognize it (I'm banking on #1).

I have the following:
In the Init part of the code: Dim Message
In the "hit" event(Sub): Message = "Activated"
In the "read" event (Sub): Textbox2.Text = Message

It doesn't work as stated above. I can't "Dim Message as String". I'm wading through VB boooks looking for a clue, but no dice.
Why not in the hiot event just say Textbox2.Text="Activated" or whatever? The problem is probably that DIM is in the table_init routine, which makes it a local variable. Try putting the dim line at the top of the script, outside any sub routines.

```
Message=Array("","Press Start","Activated","message 3","message 4"...)

Sub UpdateMessage(messageno)
Textbox2.Text = Message(messageno)
End Sub
```

In this way you could define the messages presuming that you can fit in one one line and call a routine with the messagenumber to be displayed...

The first "" indiactes message(0), you can add a message there, just remember it'd be message(0) *game over might be nice?*

On Paratech's reply: it looks doable! Instead of writing the message on all hit events, I'd just have to write it in one place, and then assign messageno a value in the hit event, and send it to Sub Update Message?

For example: let's display "Hurry Up Activated" in two texboxes. Would this be OK?

```
Message1=Array("","Hurry Up","Lock Is Lit",...) 'Textbox 1
Message2=Array("","Bonus X","Activated",...) 'Textbox 2

Sub LeftSwitch_Hit()
Messageno1 = 1
Messageno2 = 2
UpdateMessage1
```

```
Updatemessage2
End Sub

Sub UpdateMessage1(Messageno1)
If TextBusy = True Then
Exit Sub
End If
Textbox1.Text = Message1(messageno1)
End Sub
Sub UpdateMessage2(Messageno2)
If TextBusy = True Then
Exit Sub
End If
Textbox2.Text = Message2(messageno2)
End Sub
```

Also, where in the code would I insert Message=Array... ?

Messages

Whats the correct way to make a message text only appear for 15 secs?

set a timer for 15000 milliseconds.
When the text is put into the messagebox, enable the timer.
When the timer expires, disable the timer and write "" to the messagebox.

three options.

1) When renaming objects, after chaning the name, press enter, select something else, press tab...etc. the name doesn't always change -- small vp bug.

2) save the table and reload the table. this sometimes works.

3) for timers, it makes no difference where they are on the table -- so just delete the old timer and make a new one.

Remember, if you changed the timer's name to MessageTimer, first I wouldn't recomment using spaces in names if you can avoid it. Anywayz, the script would use Sub MessageTimer_Timer, and you would use MessageTimer.Enabled and MessageTimer.disabled.

If you named it Message, it's easier to remember (for me anyway) -- that way it would be Message_Timer, and Message.enabled/Message.disabled

Short form
Change the name of your timer with no spaces.
use Exactly what you renamed it to in your script.
Change every instance of the old timername to the new one.

That's all there is to it.
(edit/find at the top will save you hours of time -- F3 moves ahead to the next occurance)

Animate Text

To animate text effectively you have to layer text boxes...there could be other ways, but it's the best solution I've found so far...

By making text boxes different sizes, you can achieve all kinds of growth, wipes and assorted dissolves. Just remember that the most recent text box written to at a given x/y will be the top box, and therefore the one that is displayed. It's also good to keep in mind that a font size that is too big for the box, will simply make PART of the text disappear...you can use this to great effect.

The table I'm currently working on currently has 10 layers of text at a single x/y, and that number will likely quadruple by the time I'm done. BUT, it does allow me to some near DMD quality effects.

Well, a really simple solution to doing things like scrolling text (text that scrolls from right to left or vice versa), text wipes, etc. would be to first of all set up your text box with a mono-spaced font, such as Courier New. Determine the maximum number of characters you can display in the box at the font size you've selected, then write a few functions to display the effects as you want them to appear. You'd run the effects on a timer - that's the best way I can think of to do it.

I'm too tired at the moment to write up code samples, but I'll experiment with some text effects as a VPT file and post up the result when I get a chance. Here's what I'll attempt to tackle:

* Marquee from right to left
* Marquee from left to right
* Wipe in from left
* Wipe in from right
* Wipe out from left
* Wipe out from right
* Wipe in from center
* Wipe out from center
* Blank, then asterisks from center, then text from center
* Text, then asterisks from center, then blank

I came up with pretty much the same idea as Dorsola, but I was writing while he was posting (I'm a morning person, and it's morning here!). Here's some quick code I wrote up for a simple scroll that goes from right to left. I added a timer called ScrollTimer and a text box called ScrollingText with a monospaced font to my backdrop (proportional fonts would be hard, since there's no text width function in VBScript).

code:

```
dim scrollPosition
scrollPosition = 0
'Text has 16 spaces after it so it will go off the edge of the screen
const ScrollText = "I AM SCROLLING TEXT! BOW DOWN BEFORE ME!                "
const ScrollTextLength = 16
        Sub ScrollTimer_Timer()
        'Scroll horizontally based on current position
        dim s
        if scrollPosition = 0 then
          s = ""
          elseif scrollPosition < ScrollTextLength + 1 then
          s = Left(scrollText, scrollPosition)
          else
          s = Mid(scrollText, scrollPosition - ScrollTextLength, _
                                              ScrollTextLength)
          end if
          ScrollingText.Text = s
```

```
    scrollPosition = scrollPosition + 1
    if scrollPosition > Len(scrollText) then scrollPosition = 0
End Sub
```

Going left to right, doing wipes, etc., are all pretty much variations on this theme. As Dorsola says, having a bunch of subs to handle different effects is the way to go.

Vertical scrolling is a little different. Since you can't control any attributes of the text box, you can only scroll one full line at a time. Your best bet is to either do as PeBo does and have multiple boxes for different vertical positions or have a really tall text box.

Yup, it's messy, but the advantage is that you can change fonts, font colour, text size, position and angle on the fly with no more that 10-15 lines of code. This means you can have a string appear at a barely readable 6-8 point, have it grow to 36 point, start flashing, and then use the other suggestions for single box manipulation in this thread to have it scroll off to the side. It also means that using a custom font, you can animate virtually anything from a full DMD style multiball announcement to a dancing baby (if you're not sick of them yet). Additionally if you create an array for all your text boxes, accessing any one of them is very simple and can be a variable value triggered event. You can easily scroll text in a single text box, but if you want any other type of effect, I can't see how it would be possible without using layers.

Okay, I've come up with a sort of library for text effects in a single box. Check out this VPT file (http://www.descent2.com/dorsola/files/vpt_texteff.zip) - it is just a text effect demonstration, so it does nothing else. You can direct the animation using the keys shown in the Instructions box. Examine the Table1_Init() and Table1_KeyDown() subs to see how easy it is to use this system.

This system is fairly preliminary - it'll currently only work with one text box, one string, and assumes that you want to go either from or to a blank text box in the effect. IMPORTANT: It will also only work properly with an EVEN TextBoxSize value! I plan on revising it so these effects will scroll, wipe and otherwise replace existing text, so the whole system will be simpler when it's finished.

In this example, I've set up a text box that can hold 40 characters, and the string you're working with is "THE FUN HOUSE CLOSES IN 15 MINUTES" (34 chars). If your string is larger than the text box size, some effects will truncate the string, while others will allow the whole thing to appear.

There are 16 effects to choose from:

* Scroll in from right
* Scroll out to left
* Scroll across from right to left
* Scroll in from left
* Scroll out to right
* Scroll across from left to right
* Wipe in from right
* Wipe out to left
* Wipe in from left
* Wipe out to right
* Wipe in from center outwards
* Wipe out from center outwards
* Wipe in from edges inwards
* Wipe out from edges inwards
* Blank to asterisks to text from center outwards
* Text to asterisks to blank from center outwards

The last two effects are coded separately from the rest, but are essentially combinations of other effects. They'll be coded as such in the next revision.

Back to Index

Flippers

Is there any way to make the flippers feel more realistic?

Some people find these settings help:

1. Start VP and open any table.
2. Scroll down until you can see the flippers.
3. Highlight the left flipper by clicking on it.
4. Click on the options button on the left side.
5. This will cause the settings for the Left flipper to be seen on the right.
6. In the box marked 'start =' change the number to 112
7. In the box marked 'End = ' change the number to 61
8. Then highlight the right flipper.
9. In the box marked 'Start =' change the number to 248
10. In the box marked 'End =' change the number to 299
11. Click on File drop down menu and save the table.

I did not mention the speed value because it seems to require a different one for each table. For some a speed of 0.135 is just to fast. I have found that a speed range of 0.05 to 0.07 works good for most tables.
The strength value does not seem to affect the shot very much. I could be wrong on that but the speed seems to be more important.

Also keep in mind that flipper radius, particurally the base radius, is really important to propper flipper operation. If the base radius is too large, that can also cause the flipper to seem sluggish. Seems that anything under 21 or so works fine.

How do I render flippers non-operational when the table loads up?

Hi, I assume that you have coded the table so you press a key to start it?

Dim a variable at the top of the table (ie Inplay for example)

In Table_Init - Set this to false
When you start the game, set this variable to True.
When it's game over, set it to False.

In the Table_Keydown sub routine, find the two entries for the flippers and encapsulate them in an If statement:-

```
If inplay = true then
<the two entries for left/right flippers>
end if
```

When the table is not in play, then the flippers will not activate.

How can I increase the strengh of the flippers in the Indy Tables or others?

Just start VPinmame and open the IJ table.
After it is open click the options tab to the left of the table.
Then click on one of the flippers on the table.
A new menu will appear on the right of the table.
One of the options in that menu is "Speed".
I believe it is set to 0.135 on Indy.
Just increase it to 0.2 0.3 0.4 0.5 and on up until you find a speed you like.
You will see a "strength" option below the "Speed" option but it will not do as much for you as the speed option will.
You will then need to repeat this process for the other flipper and save the table.
Hope this helps some.

One of the most common "complaints" I see (besides the usual bugs of balls dissappearing, being stuck in walls, etc) is that the particular user doesn't like the flipper physics. Well, you can change them. I know that most know how to modify the flipper physics, but apparently some of the new user's don't, so I'm writing this for people who are new to the program. Here's how to change flipper physics, and it's quite easy..
After loading the particular table, go into the edit screen (the screen that looks like a blueprint with the icons on the left side). Now carefuly click on one of the flippers so it's highlighted (be careful not to move it). Now go to the menu at the top and click "edit", then click "options". A interface should appear on the right side of the screen. Near the bottom is 2 fields labeled "speed" and "strength". These are the 2 fields that you can change. Experiment with different numbers to find something you like. An easier and safer way however is to find a table that has physics to your particular liking and get the flipper settings on that table and place them onto the table you wish to change. If anyone's interested, I used the settings from Ian's tables (which were input by metallik, if I remember correctly). Attack from Mars and Theater of Magic had physics that I can play well with. I hope this helps anyone who can't make shots that were normally made in real world pins to get more enjoyment out of the tables

I have been experimenting with flipper angles on all of my tables.
I have found out that "Metalliks" numbers work great for every table I have. It allows for center shots to be made the way they are supposed to be. I can now shoot the castle and ball lock in MM with great accuracy. And it has restored the ball lock shot from the left flipper of my TZ table. In fact these settings have improved every table I have put them on and anyone can do it. I had no experience doing this just a day ago but for all of you out there that want great flippers and have never messed with any settings before, I am going to tell you how to do it easily.
1. Start VP and open any table. (AFM and TOM are already set right)
2. Scroll down until you can see the flippers.
3. Highlight the left flipper by clicking on it.
4. Click on the options button on the left side.
5. This will cause the settings for the Left flipper to be seen on the right.
6. In the box marked "start =" change the number to 112
7. In the box marked "End = " change the number to 61
8. Then highlight the right flipper.
9. In the box marked "Start =" change the number to 248
10. In the box marked "End =" change the number to 299
11. Click on File drop down menu and save the table.

Yea. I did not mention the speed value because it seems to require a different one for each table. For some a speed of 0.135 is just to fast. I have found that a speed range of 0.05 to 0.07 works good for most tables.
The strength value does not seem to affect the shot very much.
I could be wrong on that but the speed seems to be more important.

Also keep in mind that flipper radius, particurally the base radius, is really important to propper flipper operation. If the base radius is too large, that can also cause the flipper to seem sluggish. Seems that anything under 21 or so works fine.

Flipper Strenght

Just start VP/VPinMAME and open a table.
After it is open click the options tab to the left of the table.
Then click on one of the flippers on the table.
A new menu will appear on the right of the table.
One of the options in that menu is "Speed".
I believe it is set to 0.135 on Indy.
Just increase it to 0.2 0.3 0.4 0.5 and on up until you find a speed you like.
You will see a "strength" option below the "Speed" option but it will not do as much for you as the speed option will.
You will then need to repeat this process for the other flipper and save the table.

Changing keys for flippers to keys "3" and "4"...
How can this be done? I have all the newest files... It worked for a while, but then just stopped... I unistalled and reinstalled everything and still can't get them to stick! Is there a permanent solution to this, to get the 3 and 4 keys to be the main flipper keys for ALL games?

All keyboard handling is done in two sub:s inside the script called

<TableName>_KeyUp(...) and <TableName>_KeyDown(...)

(search for keyup or keydown in the script)
If the tables are using the WPC.VBS, S11.VBS etc the keyUp and keyDown subs only calls the keyboard functions in the *.VBS file.
Near the end in the *:VBS file you will find a lot of "Const" declarations.

```
Private Const keyShowOpts = 59 '(F1) Show options
Private Const keyShowKeys = 60 '(F2) Show Keys
Private Const KeyReset = 61 '(F3) Reset Emulation
Private Const KeyFrame = 62 '(F4) Toggle Window Lock
Private Const KeyBangBack = 20 '(T) Bang Back
Private Const keyInsertCoin1 = 4 '(3) Inser coin in slot 1
Private Const keyInsertCoin2 = 5 '(4) Inser coin in slot 2
Private Const keyInsertCoin3 = 6 '(5) Inser coin in slot 3
Private Const keyInsertCoin4 = 7 '(6) Inser coin in slot 4
Private Const keyStartButton = 2 '(1) Start button
```

You can find keycodes in the helpfile supplied with Visual Pinball

Back to Index

Kickers

```
Sub kicker1_hit()
kicker1.kick -45,10
end sub
```

Use above or

```
Sub kicker1_hit()
kicker1.kick 305,10
end sub
```

The angles are 0 to 359 technically though negtives appear to work as well. Pinman's will give you the same result..

Think of the directions as a clock face that goes

---------- 0
---------305-------45
--------270---------90
---------225-------135
----------180

For the direction you want to kick it.

How To: Directional Kicker

I got asked how I did my aiming-and-shooting kicker in MiniGolf, so I made up a small pinball where you aim with the Flipper Keys and Shoot with either CTRL-Key. Here's the brief explaination:
Here is a VPT with just a directional shooting kicker. It has two timers: Rotate and Fire.

RotateTimer: set rate to determine speed of rotation. Can't get much faster than 50ms. Incorporated a RotateStep to rotate by 5 degrees at a time for a faster "feel."
FiringTimer: sets the rate in which balls shoot.

I also added several globals, most important are MaxBalls and BallsInPlay. Make sure BallsInPlay don't exceed MaxBalls. In the example, it's 20 balls at a time. Slows down my 500MHz processor. This one requires some tweaking.

Attached is the VPT file. Open up the options window on the table to access timer rates. Open up the script to see what I did. I commented all of my additions to the basic table with the word "HERE". So search for HERE to find all of the changes. It's a very short script, probably can just read it.
As for a directional indicator, that's another lesson. I still don't have it down yet. But here's the gist of it: Create N lights for N possible directions. In Minigolf, I had to round down to the nearest 5th degree. Don't worry if the lights overlap, the one that's lit will light completely. Turn off all lights and light the correct light in that order.

The other way to indicate direction is to use N drop walls for N directions. Drop all walls and raise the correct wall to indicate direction. This is the method I used in MiniGolf.

The attached VPT has greater detail in the script.
Yes, I would like fries with that

Date: 06-17-2001 on 10:23 a.m. **Attachment:** gunkicker.vpt

Random Directions
Please could someone tell me what line is needed for a kicker to kick out in a random direction?,with all directions enabled.Is it possible to have the ball kicked out at a random speed?.For example,between the speeds of say 30 - 80.If so,how do you do that?.Many thanks in advance for any help given.

This is just a guess (works in other basic-style languages, unsure about VP though) try it anyway.

```
Kickerdirection = Int(Rnd * 360) + 1 [angle at which the ball is kicked]
Kickerspeed = (Int(Rnd * 50) +1) + 30 [speed at which the ball is kicked]

Kicker1[name of kicker] = Kickerdirection, Kickerspeed
```

However, you may like to modify it, and delete square brackets bit. But that is the basic code. I would do it this way:

```
Kicker.Kick 360 * Rnd, 30 + 50 * Rnd
```

I would also put a Randomize in your table_init sub so you don't get the same series of numbers every time.

```
Kicker1.Kick int(rnd(1)*359), int(rnd(1)*10)
```

Replace the 10 with whatever you want the max to be, and if you want it say a minimum of 10 use int(rnd(1)*10 + 10) which will make it between 10 and 20.

Make sure you have RANDOMIZE in your script somewhere. You can also use CInt instead of Int.

Can someone tell me what the middle 2 variables on my kicker means?

```
Kicker1.kick 270, 1 + rnd (1) * 40

1 + rnd (1) ?
```

Also when I enter my kickerhole it seems to lose its kick about the 5th time its entered.

Should I just add Kicker1.kick 270, 1 + rnd (1) * 40 to each if statement?

The first parameter is direction, with 0 being due north.
The second parameter is strength of kick.
1+rnd(1)*40 generates a number from 1 to 40. The rnd function generates a number from 0 to .9999. Multiply that by 40, you get 0 to 39.99999. Add one, you get 1 to 40.9999. The decimals are truncated if the kicker function requires an integer parameter.

If you want a kick of strength 20 to 40 instead of 1 to 40 try:

```
20+rnd(1)*20.
```

Kicker shoot along Z-Axis - YES, working!!!

Everybody says that Kickers don't have the ability to shoot along the Z-Axis,
but then you all should take a closer look to the VP Documantation as there
are some properties of the objects not visible in the editor!
Like this one from VPReference.chm):

Kicker.Kick(Angle As Single, Speed As Single, [Inclination As Single])

You notice - INCLINATION!!!
Ok, I built an example-table with some stairs - now, let's play ball....

Attachment: UpKicker.vpt

I just figured out, how the "inclination" parameter has to be set:

as default, the kicker kicks horizontal (0 deg.), a 90 deg. kick (almost straight vertical)
would be something like 1.45 and everything between them is the corresponding angle
(ie. 0.75 is almost 45 deg.)
You have to play a little with the value to find out the needed angle, but it's great.
Don't forget to set the glass-height and a suitable shoot-strength
Oh, btw: ever tried to set the table-inclination to values between 90 and 180 degrees?
Nice look from below the table upwards

Kicker Timer
what line / lines need to be added to kicker script to make the kicker keep hold of the ball for about 5 seconds before releasing it

```
sub kicker1_hit
kicker1.enabled=false
holdtimer.enabled=true
end sub

sub holdtimer_timer
kicker1.kick 180,20
kicker1.enabled=true
holdtimer.enabled=false
end sub
```

Then set the holdtimer interval to 5000 (for 5 seconds).

Back to Index

Lamp
Spinning Lamp Totan

Sim Tech Sample
After spending about 6 hours straight working on it, I figured out a reasonably code-simple way to simulate the spinning lamp in TOTAN. I've attached a VPT that contains a mockup of the lamp object and the simulator I designed for it.

This thing is far from perfect - the object can be fooled into turning the wrong direction; it can't tell the difference between a dead-on hit and a glancing blow (one where the ball is deflected off at an angle); and I don't have the rotation down very well - friction is poorly simulated.

But, this is a start, and it *might* be the best we can do until we're able to get the ball's direction and speed values from (or during) a hit event.

Unlike captive balls, I think getting the free-spinning lamp in TOTAN to work satisfactorily with the current VisPin release is just a bit out of reach. If the person working on TOTAN is satisfied with this, though, or has a better solution, it'll still make the machine plenty playable. =)

Feel free to play around with this thing, guys - I'd be interested in seeing if what I started evolves. =)

WARNING: Heavy math involved! If you want to piece through the math in the script, I included some notes, but there's some complicated trig stuff to sort through. Be careful if you modify the math routines.

Back to Index

Pinmame

How do i move the display and make it stay there next time i play a table?

The last position is saved but the script can override it. Look for a line in the script like

.SetDisplayPosition ...
or
Controller.SetDisplayPosition ...

and delete it or comment it out.

How do I reduce the size of the DMD on games like Baywatch?

Hit F1 while vpinmame is running. Make sure to use the COMPACT DMD option. Also, remove the Double Size option if it's selected.

How do i move the display and make it stay there next time i play a table?

The last position is saved but the script can override it. Look for a line in the script like

.SetDisplayPosition ...
or
Controller.SetDisplayPosition ...

and delete it or comment it out.

Is there any way to make the vPinMAME games sound better?

Some people report success by altering the DirectX sound configuration.

Click the start menu
Click the run button
Type DxDiag and press enter
On the sound page (middle tab), select "basic acceleration" by sliding the slider to the left hand side.
Press Exit.

Are there any bugs/issues with vPinMAME v1.10?

#1) Some of the BSMT games have volume issues. Either too loud or too soft. Nothing much can be done to fix it for now. You should be able to use the 8,9 keys to adjust volume for the SEGA games. If not, we might just need a new sega.vbs to handle support for it, so sit tight..

#2) The BSMT uses COMPRESSED audio samples on most games. We DO NOT yet know how to make it sound good when played back in vpm. It will sound loud, scratchy, and generally bad!! This is a known issue, PLEASE DO NOT post about it, we already know..

#3) In some cases, lower end setups *might* experience more static for ALL games with sound in the new release. Please read the whatsnew.txt for a brief description on this.. We're trying to work out something better, but just be aware, that's the way it is for now.

#4) The VPinMAMETest.exe is NO LONGER USED! Use the "TEST" button from the setup/installation program (setup.exe) instead to check out your vpinmame installation.

There are some issues with the Mech Handler code, which means that certain tables may have broken playfield toys/diverters or generally exhibit strange behavior.

Back to Index

Script

Just a couple of questions about script....
1) I am having problems defining variables:sometimes
they dont seem to register at all! do you have to DIM them all?.....and what about the program
order?....does it make a difference
2) My version of VPinball seems ill, it crashes left
right and centre
550p3,25M6,16MrivaTNT,Norton,Network card,USB
MIDI interface,Win98,SoundBlaster1024
sometimes use soft FSB, PC Accellerator to overclock.?any ideas?
3)What is the routine to determine if a
has run out?....ie I am in a subroutine for a skillshot, if the
timer runs out then skillshot has finished....but ive already done loads of code....do i have to repeat it
under the SkillshotTimer_Timer() routine?
4)Is there an endsound command?
5)Anyone got any VB script info or tutorials yet?

Any variable should be DIMed whith the exception of local variables used to pass parameters to a subroutine.

Their order doesn't matter (I like to put single value variables first, followed by single dimension and multi-dimension arrays after, but that's a personal preference). Best place though is in a place that won't be read more than once...like the top opf your script

A temporary variable (used for stuff like for/next loops) can be placed within the sub routine that uses it, but remember that everytime the Dim statement is read, the variable is going to be re-set.

As far as a skill-shot timer is concerned, you can have it start as soon as the ball is shot (or when it hit's a launch alley gate). Set the time for the correct duration (1000=1 second), and simply include a x.timerenabled = false (where x is the timer's name) within the timer subroutine. If it is an actual timer object you're using, then it would just be x.enabled =false. If you include the "=false" within the timer routine itself, the timer will end as soon as the duration has passed.

So for your skillshot it would be something like...

```
sub start()
Light_ss.state=lightstateon
x.timerenabled=true
end sub

sub x_timer()
x.timerenabled=false
light_ss.state=lightstateoff
end sub
```

You can check if the timer is still active with a simple "if x.timerenabled" statement ("if" will always return a 0 for false and a 1 for true, so you don't need any "= false" or "=true"). One suggestion, rather than using a gazillion timer objects, use some of your table elements instead. Every object has it's own timer that you can use. So for your skillshot, if you use an alley gate as the hit event that starts the time, why not use the gate's timer. ie...

```
sub gate_hit()
gate.timerenabled=true
end sub
```

The endsound command is a good question...I have no idea.
Actually be careful of how much you follow the above...I'm self taught myself, so I could be full of shit. If no one points out any errors in my suggestions, you can try them, but don't take em to the bank!
You only have to DIM variables if you've put "option explicit" at the start of your script. (at least one person has told me that he thought "option explicit" activated some kind of adult content lock! <grin>).
However, if you skip the "option explicit", your programs will run fine, but the variables will only be scoped for the subroutine in which they're used... so if you want a variable to be globally available, you'll need to refer to it somewhere within the "initialisation" code at the top of the script (i.e. the stuff at the top of the script).... otherwise, the variable will reset to zero whenever you move out of a routine where it's scoped - if you have variables that don't seem to hold their values, then this is almost certainly the problem.
To be honest, there probably isn't any good reason to not use "option explicit" - it forces you to explicitly declare all your variables upfront, and catches out loads of typos and syntax errors that you'll otherwise miss. Which makes me wonder why I forgot to use it in dragons castle or star race..... maybe I like a challenge... (disclaimer: I've been VB coding for even less time than Pebo. And I'm so stupid that I don't even use "option explicit" in my programs. I wouldn't listen to a word I say if I was you...)

Mod's

Well, really i doesn't need to be dimmed at all, but I always have option explicit in my code so I don't have variable naming problems. But even so, I could declare i as a global, but nothing about i required global scope. It's a local variable used by subs, so I declare it every time. Of course, you're free to commit whatever travesty of coding style you want...

Mod is the modulus operator, which does an integer division and returns the remainder. So, 9 mod 7 = 2, 21 mod 7 = 0, etc. I use it because with my 4 light sequence, there are 7 possible states (1 - 2 - 3 - 4 - all off - all on - all off - 1 - etc.). That way, I can just keep adding to my counter and not have the annoying if counter >= 7 then counter = 0 code. In this case, it's just personal preference, but in some of my games the attract sequence can get quite complicated (I believe Frontier has a sequence that is 90 or 100 lights long) and different lights go at different speeds. So I can have this sequence going fast and another 10 light sequence going slower with (attractCounter \ 4) mod 10 to kind of break things up a bit (\ is the integer division operator). So that's why I use mod.

Back to Index

Random
Random #'s

How do I generate a random number?
To produce a random number, use the rnd function.

Before calling Rnd, use the Randomize statement without an argument to initialize the random-number generator (this ensures that you get a random number every time you call rnd)

To produce random integers in a given range, use this formula:

```
Int((upperbound - lowerbound + 1) * Rnd + lowerbound)
```

Here, upperbound is the highest number in the range, and lowerbound is the lowest number in the range.

So if you wanted to get a random number between 1 and 10 then you would write the following:

```
Dim x
Randomize
x = int((10 - 1 + 1) * Rnd + 1)
```

this is a part of my match section... I'm trying to make my random number not repeat itself.

just to be clear, if the last number was 10 the next random number must not be 10, but instead go back and 'reroll' untill the number has changed.

```
Sub RunMatch()
MatchResult = Int(Rnd(1)*10) * 10
If MatchResult = 0 Then
BallText.Text = "00"
Else
BallText.Text = MatchResult
End If
MatchCount = MatchCount + 1
MatchTimer.Enabled = True
```

What is the range of numbers you want to choose from?
You could use recursion, but it's a bit dangerous as your program could litraly loop itself to death.
Um try the following, this is off the top of my head so forgive any small mistakes.
This script assumes that you want a range from 1 to 10.
At the top of the table

```
Dim LastRandomNumber 'Var to hold the last number
In the Table_init routine
LastRandomNumber = 0
```

Now the routine to call to get the random number.

```
function GetRndNumber (lastnum)
dim RandomNumber
randomize (timer)
RandomNumber = int(rnd * 10) + 1
if RandomNumber = LastRandomNumber then RandomNumber = GetRndNumber(lastnum)
GetRndNumber = RandomNumber
LastRandomNumber = RandomNumber
End Function
```

To get the RandomNumber use the following

```
GetRndNumber (lastRandomNumber)
the random number I'm trying to produce is for the match,
Int(Rnd(1)*10) * 10
```

1 through 10 times ten...

your solution looks better than any of the ones I was trying but I still fail to make it work. it keeps generating 0 for some reason... it's probly me, I'll take another look at it tmrw and see if it makes any more sense then. I think I'm just burnin out for today.
I'm pretty sure that if you're using Int(Rnd *10) * 10 to generate your numbers, you're going to get numbers ranging from 0 to 90.

Rnd generates numbers in the range 0 <= Rnd < 1, so you'll never get 100. You need to adopt MrRalphMan's approach: (Int(Rnd*10)+1) * 10.
Damn, never had a chance to do much of anything today till just now, and I'm sure you got up long ago here is my whole match section of my script as it is..
If you needed/want more than that I can just send you a wip of my whole table, not a problem.
BTW Joe, I belive you're right, my code as is WILL generate from 0 - 9, I know I said 1 - 10 earlier, I was wrong. I had previously figured all that out when I first put that code together, I was being forgetfull and lazy in my last post, very tired, too much VP! Anyhow... for a match routine 0 - 9 works perfectly 0 = 00 and 9 = 90 it's all just double digits, so in this case it works perfectly

```
Sub RunMatch()
MatchResult = Int(Rnd(1)*10) * 10
If MatchResult = 0 Then
BallText.Text = "00"
Else
BallText.Text = MatchResult
End If
MatchCount = MatchCount + 1
MatchTimer.Enabled = True
End Sub

Sub MatchTimer_Timer()
MatchTimer.Enabled = False
If MatchCount = 20 Then
MatchCount = 0
EndMatch
Else
RunMatch
End If
End Sub

Sub EndMatch()
If MatchResult =(Score-Round(Score/100)*100) Then
AddCreditAward
End If
GameOver
End Sub

Sub GameOver()
GameOn = False
Attract.Start
End Sub
```

Taking Camera Shots of a real table

Well.. Ideally you want to have your camera at the highest resolution and lowest compression settings possible. You'll want to take the playfield glass off and get that pinball in a lot of light (outside is best but otherwise a lot of nice diffuse (SP?) light).

If you have enough light (only the pics will tell) Then position yourself or the camera to be directly overhead of the table and line up the camera lengthwise with the table (so cropping will be less and you'll get a better and bigger overall pic).

Then shoot the following shots until you get at least one really good one of each.

1 full Playfield shot
1 shot with 60% of the lower half of the table
1 shot with 60% of the upper half of the table.

You can also take the above sequence with flash as well and see if it might come in handy later for something. With flash you'll have the flash bounce so I'd recommend the other for less cleanup.

Take a lower light pic of the table as well with all lights on for color matching.

Then take straight on closeups (watch the focus.. you can be a distance away just try to fill the shot with the zoomed plastic) of each and every individual plastic and drop target type on the machine.

Then take a shot of a lit and unlit backglass pic. This is a tough one because of the glare that most BGs give.

That should be a good start. Check how the pics came out when viewed full size on your computer and reshoot any that were too dark or blurry or had some other problem.

Creator of Visual Pinball . If it wasn't for Randy I'd be playing Mame
http://www.randydavis.com/vp/

Shiva The best pinball site on the net
http://www.hippie.net/shivasite/UltraBoard/UltraBoard.cgi

AJ's Table Site Good selection
http://www.thefew.com/vptables/

IR Pinball , Table Downloads, Table Creators and they acquired the talented Leo
http://www.hippie.net/shivasite/irpinball//nonirptables.html

Thanks to Randy, Shiva, (Shiva Site Maintainers) for getting it going again and everyone who made a table.
My next release of Knock My Fn Teeth out will be the Final…….when? soon as its finished
I'm slower with this stuff but like getting it to work, It's all greek to me.
I need brown balls for my next table "Eat Poop & Die" ☺
Keep up the good work & keep them tables coming.
Cold1

Back to Index

Appendix C
Table List

10th Inning (United 1948).vpt
1cactus.vpt
1Juggler_Beta0.vpt
2001 (Gottlieb 1971).vpt
3 Stooges V1.0.vpt
4 Square (Gottlieb 1971).vpt
5 Star Final2.vpt
5StarFinalVP5.vpt
7 of 9 breakout.vpt
77.vpt
777.vpt
777U.vpt
77U.vpt
acdc.vpt
Addams Family 4.5 OCX.vpt
Addams Family 4.7.vpt
Addams Family Gold 4.4.vpt
Adventure RC1.vpt
Adventure RC2.vpt
adventure.vpt
Age of Chivalry_b2.vpt
Air Aces (Bally 1974).vpt
Aladdin's Castle.vpt
ALASKA (Interflip of Spain 1978)Reel.vpt
Alaska.vpt
Alien Invasion Final.vpt
Alien Poker 1.1.vpt
Alien Reactor.vpt
Alien Table.vpt
ALIENDETOX.vpt
AlienReactor.vpt
Aliens 2.0.vpt
alphaone.vpt
AlpineClub_RC1df5.vpt
AMD against the Dark Empire - FINAL.vpt
AMD Against the Dark Empire.vpt
Anna's Colours V1.0 (sphere 2001) .vpt
Anna's Colours V1.1 (sphere 2001) .vpt
Another World Two V1.0 (sphere 2001) .vpt
Apetite For Pinball (GNR)beta4.vpt
Apollo (Williams 1967).vpt
Apollo_RC2.vpt
apotopongF2.vpt
AquAdv.vpt
Area51.vpt
Arkanoid 0.7.vpt
ATARI DIG DUG V1.0.vpt
AtlanticCity.vpt
AToilet.vpt
AToiletscript.vpt
Attack From Mars RC1.1xx2.vpt
Attack From Mars RC2.vpt
Aztec_RC1a.vpt
Aztec_RC2.vpt
BabyContactV2.vpt
BABYDRAGV1.0.vpt
babylove11.vpt
Back to the Future 1.0.vpt
Bad Cats 2.0.vpt
Bad Cats Wide Mod.vpt
Baffle Ball Sr (Gottlieb 1932) v2.vpt
BallDropper.vpt
Balls A Poppin' (Bally 1956).vpt
Balls A Poppin' (Bally 1956)Reel.vpt
Ballyhoo2.vpt
BallyhooVP5.vpt
BallyTribute1_0.vpt
BallyTribute1_1.vpt
Bank A Ball (Gottlieb 1965).vpt
BankABall_RC2.vpt
barracora1.0.vpt
baseball.vpt
BATMAN Home made 2V1.0.vpt
Batman R7.vpt
batmanR4.vpt
BayWatchV2.0.vpt
BaywatchV3.0.vpt
Bear Creek 4volt revolution.vpt
Beat Box b0.9.vpt
Beatles Help.vpt
BEATLES.vpt
Beavisb3.vpt
Berzerk FNL.vpt
big brave1-1.vpt
Big Daddy (Williams 1963)8-1-01.vpt
Big Indian (Gottlieb 1974)Reel.vpt
big indian1-2.vpt
BIG SCORE V1.0.vpt
Big Top (Gottlieb 1964) reel.vpt
Big Top (Gottlieb 1964).vpt
BigDaddy_RC3.vpt
bigleague13.vpt
Black Hole, The (Gottlieb 1981).vpt
Black Jack.vpt
black rose cannon3.vpt
Black Rose.vpt
Black Sheep Squadron (Astro, 1979).vpt
Black Sheep Squadron-beta1.vpt
blackhole_011.vpt
BlackJack_RC2.vpt
BlackKnight Beta vp R2.0.vpt
BlackKnight_vp 1.1.vpt
BlackKnight2000_Beta4.vpt
BlackKnight2000_Beta5.vpt
Blackout 1.2.vpt
Blasteroids 3d.vpt

Blue Ribbon (Bally 1965).vpt
BlueRibbon_RC2.vpt
bo_Script.vpt
BombSquadBeta01.vpt
Bon Voyage (Bally 1974).vpt
Boomerang v1.4 - aProgrammingExample.vpt
Boris 04 8bit.vpt
Bottom of 9th baseball b2.vpt
Bow and Arrow (Bally 1974).vpt
Brain Salad Surgery V1.1 (sphere 2001).vpt
break shot b3.vpt
Broadway (Bally 1951 Bingo).vpt
BruceLee.vpt
Bubble Bobble015.vpt
buccaneerA.vpt
Buckaroo (Gottlieb 1965).vpt
BUCKAROO (Gottlieb 1965)Reel.vpt
Buckaroo-novelty.vpt
Budget Pinball.vpt
bugsbunny.vpt
bumpnjump.vpt
Butterfly10.vpt
ButtonsandBows(Gottlieb 1949).vpt
Cactus Canyon 1x.a.vpt
Camelot (Bally 1969).vpt
Canasta86 (Inder 1986).vpt
Captain America beta Final3.vpt
Captain Garrison V0.991.vpt
Captain Garrison1-2.vpt
CaptFantastic.vpt
Card Sharks Final.vpt
Casino RC1.vpt
Centaur (Bally 1981)1.1.vpt
centaurVPM.vpt
Champ (Bally 1973)1.1.vpt
Chinatown (Gottlieb 1952).vpt
Circusworld2.0.vpt
Cirqus Voltaire Beta 2.7.vpt
Citadel-V1.2 (sphere 2001).vpt
Classy Bowler (Gottlieb 1956).vpt
CleopatraBeta2.vpt
CloseEncountersV2_3.vpt
combat.vpt
Comet 1.0.vpt
Commando.vpt
Coney Island (Bally 1951 Bingo).vpt
Congo 1.0.vpt
Congo Mod.vpt
Corral (Gottlieb 1961).vpt
Corral (Gottlieb 1961)-tb5.vpt
Corvette 1.1.vpt
cosmic gunfightrc1.vpt
CoverGirl_RC1.vpt
CoverGirl_RC2.vpt
Creature from the Black Lagoon 1.5.vpt
Crescendo (Gottlieb 1970).vpt
Cyclone 0.9.vpt
Cyclone 0.92b.vpt
Cyclone Mod.vpt
Dads Table.vpt
Dark Side Of The Moon V1.0 (sphere 2001).vpt
DeepBlueSea_1.0.vpt
Defender 1.1.vpt
Demolition_Man_Wip_3.Vpt
Demons & Wizards V1.0 (sphere 2001).vpt
DeuceWild.vpt
devils Dare (Gottlieb)r01.vpt
devils dare_r012.vpt
devilsdare097.vpt
Diamond Jack11.vpt
DiamondJack_RC3.vpt
Diner 1.2.vpt
Diner 1.5.vpt
Ding Dong (Williams 1968).vpt
Dirty Harry.VPT
Disco Fever beta 1.vpt
discof08.vpt
dkong 1 different.vpt
dkong Final.vpt
Do run run.vpt
Doctor Who 1.2.Fast.vpt
Doctor Who 1.2.Slow.vpt
Dolly Parton 1.5.vpt
Dolly Parton No Rom 01.vpt
DollyParton wide MOD.vpt
DollyParton V12.vpt
Domino (Gottlieb 1968).vpt
Doom 122.vpt
Doom 2 v1.1.vpt
Dord.vpt
Double Vision - v1.9.vpt
dracula_rc1.vpt
Dragon (Interflip of Spain 1977).vpt
DRAGON (Interflip of Spain 1977)Reel.vpt
Dragon Ball Pinball.vpt
Dragon Master0.1.vpt
Dragon V1.0 (sphere 2002) .vpt
dragon's castle.vpt
Dragons Quest.vpt
DRAKOR.vpt
Eager Beaver (Williams 1965).vpt
earthshaker_mod095b.vpt
Eat Poop n Die.vpt
Egg Head (Gottlieb 1961).vpt
Egypt v1.0.vpt
Eight Ball 1.2.vpt
Eight Ball Deluxe 1x.vpt
EightBallChampv2_1.vpt
Eldorado.vpt
elektra_wip1.vpt
Elvira and the Party Monsters 0.99.vpt
ELVIS PRESLEY V1.0.vpt
Embryon v1.1-4sloPC.vpt
EmbryonV1.2.vpt
Emerald Altar V1.1 (sphere 2001) .vpt
Emu Christ beta 2.5.vpt
Escape From NY b3.vpt
EscherWaterfall_v0.1.vpt

Evel Knievel 1_0.vpt
Excalibur b4.vpt
Eye Of The Tiger.vpt
F-14 TomCat 1-7.vpt
Falstaff (Gottlieb 1957).vpt
Fathom 1.0.vpt
Fathom VP no roms.vpt
fathom vp2.vpt
fathom2.vpt
Fight Night 0.4.vpt
FinalFantasy1.vpt
Fire Power2 b10.vpt
Fire Rescue V1.1FIX.vpt
FireBall-SpinnerOff (Bally 1972).vpt
Fireball-SpinnerOn (Bally 1972).vpt
FirePower 2 1.1.vpt
FirePower V11.vpt
Fish Frenzy.vpt
Fish Tales 3.1.vpt
flash no romsv1.vpt
Flash Gordon (Bally 1981)newestx.vpt
Flash Gordon v1.2.vpt
flash_vpm 1.2.vpt
Flight2000 99.vpt
Flintstones 1.0dx.vpt
Flintstones 1.0dx3.vpt
Flip Flop (Bally 1974).vpt
Flip Flop v1.0.vpt
Flip-A-Card (Gottlieb 1970).vpt
Flipper (Gottlieb 1960).vpt
Flipper Clown (Gottlieb 1962)Reel.vpt
Flipper Clown1.vpt
Flipper Cowboy (Gottlieb 1962).vpt
Flipper Pool (Gottlieb 1965).vpt
Flop House pinball034.vpt
Flying Carpet (Gottlieb 1972).vpt
Football.vpt
Force Majeure V1.0 (sphere 2001).vpt
Frankenstein 0.9.vpt
Frankenstein 1.1.vpt
Frankenstein.vpt
Freedom(Bally 1975)Eastern b1.vpt
Freedom(Bally 1975)Western b2.vpt
Freedomvpb1.vpt
Frontier (Bally 1980).vpt
Frosty the Snowman.vpt
Fun Cruise (Bally 1965).vpt
Fun House 1.0.vpt
Fun Park (Gottlieb 1968).vpt
Fun Park (Gottlieb 1968)reel.vpt
Future Spa.vpt
FutureSpaV2_0.vpt
FutureSpaV2_1.vpt
galaxian10.vpt
galaxian2 small table.vpt
Galaxy 1.1.vpt
Galaxy At War.vpt
Galaxy V1.2 no roms.vpt
GameShow V11.vpt
Gator Eyes aBC.vpt
genie v1.8.vpt
Getaway - High Speed 2 1-8.vpt
Ghost Busters old.vpt
ghostbusters 8-17-01.vpt
ghostbustersmid.vpt
Ghouls Ghosts and Goblins V100.vpt
Ghouls Ghosts and Goblins V120.vpt
Ghouls Ghosts and Goblins V121.vpt
Gigi (Gottlieb 1963)reel.vpt
Gigi B1.vpt
Goldeneye b22.vpt
Goofy.vpt
goonies mid.vpt
goonies.vpt
Gorgar 0.5.vpt
GrandLizard.vpt
GrandLizard_NoBitmap.vpt
Great Rock N Roll Swindle10.vpt
GunsNRosesv1.1.vpt
GunsNRosesversion1.vpt
Halloween53 B1.vpt
Harlem Globetrotters V1.1.vpt
Harley Davidson (Bally) 1.1.vpt
Harley Davidson (Small Table).vpt
Harly Davidson ballyv1.0 (Biger Table).vpt
Harry Potter.vpt
haunted_r01.vpt
haunted_r031.vpt
Haunted3.vpt
Hawkman.vpt
Heat Wave (Williams 1964) reel.vpt
Heat Wave.vpt
Heat Wave-voice (Williams 1964) reel.vpt
HeavyHitter.vpt
HeavyHitterVP5.vpt
High Speed b1.vpt
HillbillyHellcats v0.1a.vpt
Hi-Lo Ace (Bally 1973)1.1.vpt
Hokus Pokus (Bally 1975).vpt
HOKUS POKUS (Bally 1975)Reel.vpt
Home Run (Gottlieb 1971).vpt
Hotdoggin v1.0 (Bally 1979).vpt
HotTipv1.0.vpt
Hurricane b2.vpt
I Am Santa Claus Beta 2.vpt
I Am Santa Clause Final.vpt
ID4 0.7.vpt
Ignacio4.vpt
Ignition 10.vpt
illusionsquared5.vpt
Incredible Hulk, The (Gottlieb 1979).vpt
Indiana Jones (Redux) 1.0.vpt
Indy 500 1.40.vpt
init.vpt
Jack in the Box (Gottlieb 1973)8-8-01.vpt
Jackbot 1.1.vpt
JBWiz.vpt
Jetson203.vpt

jetsons 617.vpt
jetsons 626.vpt
jl0.75.vpt
jmj_beta1.vpt
joe.table.vpt
johns table.vpt
Joker Poker (Gottlieb 1978).vpt
Jokerz! 0.9.vpt
JP2lostworldb1.vpt
Judge Dredd v0.9.vpt
Jungle Lord 90.vpt
Junk Yard 0.97.vpt
Jurassic Park b1.2.vpt
Jurassic Park b2.vpt
Jurassic Park B6.vpt
Jurassic Park B6a.vpt
Kewpie Doll(Gottlieb 1960).vpt
kickertimerdemo.vpt
Kings & Queens (Gottlieb 1965).vpt
Kings & Queens.vpt
KINGS (Williams 1957).vpt
King's Checkers (Genco 1934).vpt
Kiss sept1 01.vpt
kmfto New Final11 2002.vpt
kmfto New Final11.vpt
Laserball (Williams 1979)v1.2.vpt
Lasercue V1.08-19-01.vpt
Last Action HeroBeta0.8.vpt
Lawman.vpt
Lethal_Weapon_3_B_S1.Vpt
LethalWeaponB3.5.vpt
lhexagone10.vpt
LinuxV0.95beta.vpt
Lite-A-Line (Keeney-Bingo-1952).vpt
LiteLineVP5.vpt
LittleJoe.vpt
Live Power (Exhibit Supply 1942).vpt
Lochness.vpt
LooneyAV.vpt
LooneyCE.vpt
Lost In Space.vpt
Lost World (Bally 1977)1.1.vpt
lostinthewoods2.vpt
LostWorld v1_2 (Bally 1977).vpt
lotto.vpt
Magic Circle b0.8.vpt
Magic City (Williams 1967) high score.vpt
Magic City (Williams 1967)redone6-30-0`.vpt
Magic City.vpt
magic g2vpm.vpt
Magic Town.vpt
Magnification V1.1 (sphere 2001).vpt
Magnotron11.vpt
Majorettes (Gottlieb 1964).vpt
mammoth.vpt
mammothVP5.vpt
Mariner (Bally 1971) RC1.vpt
Mario tshotwip5.vpt
mario.vpt
mars God of War 091.vpt
Mata Hari (Bally 1977) No Roms Final.vpt
Mata Hari 1.0.vpt
MaxPayne.vpt
mcpwalls34tb4.vpt
Medieval Madness.vpt
meteor.vpt
MHariBeta1.vpt
Mibs.vpt
Middle_Earth_b0.7 (Atari 1978).vpt
Midevil Madness_RC3.vpt
Midget Hi-Ball (Peo 1932).vpt
MidgetVP5.vpt
MIDI Sample Table.vpt
MidnightMagic.vpt
Millionaire 0.5.vpt
mini golf.vpt
Miss Universe V1.0 (sphere 2001) .vpt
Monster Bash Beta 4.vpt
Monte Carlo (Bally 1972).vpt
monty python beta.8.vpt
Mousin' Around v1.6.vpt
Mr & Mrs Pacman 3.0 VPM.vpt
Mr Mrs Pacman1_7.vpt
mr ms pacman 01.vpt
Mrdo_ver1_3.vpt
Mystic 1.0.vpt
N.N. Acoustic V1.0 (sphere 2001) .vpt
Nags (Williams 1960).vpt
NAGS (Williams 1960)Reel.vpt
NBA Fastbreak v0.5.vpt
Necromundia V1.1.vpt
Neptune.vpt
Night Rider-Beta3.vpt
Nitro Ground 1.2.vpt
No Fear Final.vpt
No Good Gofers v11.vpt
NorthStar_RC1.vpt
Noyze v1.0.vpt
nwteacherspet.vpt
OCEAN LIFE V1.0.vpt
OddsAndEvens-Beta2fixed.vpt
Office.vpt
OfficeVP5 .vpt
Old Chicago.vpt
Old Faithful (Gottlieb 1949)1.1.vpt
OXO (Williams 1973).vpt
PACMANS CASTLEV1.0.vpt
paganpinball_beta01.vpt
paragon_beta01.vpt
Party Landb5.vpt
Party Zone 1.3.vpt
Party Zone V12.vpt
PettingEcho-v0.1.vpt
Phaeton v1.0 (sphere 2001).vpt
PHOENIX.VPT
PikaPinball_V1.vpt
Pinball (Stern 1977).vpt
Pinbot 0.91.vpt

plan9-v1.vpt
Planet Probe 2.vpt
Play Mates (Gottlieb 1968).vpt
Playboy 1.5.vpt
Playboy 2.0 ocx.vpt
Playboy Bv2.1.vpt
Playboy mod Gerhard.vpt
Playboy wide Mod1.1.vpt
Playground.vpt
PM Scared Stiff FinalRev.vpt
PM STTNG V2.vpt
Police Force v1.5.vpt
Popeye 2.0.vpt
PopeyeV0.9.vpt
PotHead.vpt
Power Play 2.0 (Bally 1977).vpt
power play beta.vpt
power play_rc1.vpt
Power Puff Girls Beta1.vpt
Power Puff Girls Beta3 - 17 cleanin.vpt
Power puff mojo.vpt
powerpuff townsville.vpt
Pro Football (Gottlieb 1973).vpt
ProPool.vpt
Psycho 1.3.vpt
Psychomim.vpt
Punch Out.vpt
Queen of Hearts (Gottlieb 1952).vpt
Queen of Hearts (Gottlieb 1952)F Mod.vpt
reactor.vpt
ReelExampleTable.vpt
Revenge Of The Robot Warriors.vpt
RGB v2.vpt
Riverboat Gambler 1.0.vpt
Riverboat Gambler Mod.vpt
RoadShowB4.0.vpt
Robocod.vpt
Rocket Ship (Gottlieb 1958)F3.vpt
RockFantasyV111.vpt
Rocky And Bullwinkle And Friends - Release Beta One.vpt
Rocky06 VP.vpt
rockyvpm091.vpt
Rolling Stones B1.vpt
Royal Flush (Gottlieb 1976)Reel.vpt
RoyalFlush11.vpt
RumbleBeta.vpt
Safari.vpt
safe cracker 1-0.vpt
santa.vpt
SatanicV7.vpt
Saturn 5 V1.0 (sphere 2001) .vpt
Scared Stiff 2.0.vpt
Scared_Stiff_Table Final no roms.vpt
ScoobyDoo.vpt
scorpion1.vpt
Scramball Novelty Game (Keeney 1958).vpt
Sea Wolf (Williams 1959).vpt
Search4.vpt
Sexy Girl.vpt
Shadow_beta13.vpt
Shangri-La (Williams 1967).vpt
Shangri-La (Williams 1967)-tb5-1.vpt
Shenmue2.vpt
Sheriff10.vpt
shivaengine1_6 scripted.vpt
shivaengine1_6.vpt
Shooting Gallery.vpt
shootingstar.vpt
Silver Ball Mania 02e.vpt
Silver Ball Mania v1.0001.vpt
Silver Ball Mania vpm1.vpt
Simpsons v1.0.vpt
Simpsons v1.1c.vpt
Simpsons v1.1eXQS.vpt
simpsons2v1.0.vpt
Sinbad EM.vpt
Sinbad1.1final.vpt
Sing Along (Gottlieb 1968)1-1.vpt
Sing Along (Gottlieb 1968)-df5 reel.vpt
Six Million Dollar Man, The (Bally 1978).vpt
skateball 1.0final.vpt
skateball_rev2.0.vpt
skateball_vpm2.0.vpt
Skateboard Park-V1.0.vpt
Skateordie b1.vpt
Skykings.vpt
skyrocket_final.vpt
skyrocket_rc1.vpt
SlickChick V10.vpt
Solar Ride (Gottlieb 1979).vpt
Sonic pinball.vpt
soup.vpt
South Park 0.9.vpt
space Invaders 1.0.vpt
Space Invaders V1.1 no roms.vpt
Space Is Deep V1.1 (sphere 2001) .vpt
Space Shuttle RC2a.vpt
SpaceInV1.2.vpt
SpaceTime.vpt
Spanish Eyes (Williams 1972).vpt
Spanish Eyes (Williams 1972)8-19-01.vpt
SPANISH EYES (Williams 1972)Reel.vpt
Spanish Eyes-xtraball.vpt
SpanishBoxV1.1.vpt
Spectrum (Bally 1982).vpt
speed Racer.vpt
Sphere's Dream v1.0 (BJ 2002).vpt
Sphinx.vpt
Spider Man.vpt
SpiderMan II (2.1) Rob Grant.vpt
spiderman.vpt
SpidermanMOD.vpt
SpongeBobSquarePants1.0.vpt
Sportcham.vpt
Spring Break (Gottlieb 1987).vpt
Square (Egg) Head (Gottlieb 1963).vpt
SQUARE HEAD (Gottlieb 1963)upd.vpt

Stallturn.vpt
Star Race (Gottlieb 1980).vpt
Star Trek - TNG 2.3.vpt
Star Trek - TNG 2.5.vpt
Star Trek (VPM) beta2.vpt
Star Trek 25th 0.9.vpt
Star Trek B1.vpt
Star Trek Mod other.vpt
Star Trek Mod.vpt
Star Trek TNG 2.6.vpt
Star Wars RC1.2.vpt
Star_Trek_The_Next_Generation Final.vpt
StarJet_RC1.vpt
Stars Bingo (United 1952).vpt
StarShip Troopers b1.1.vpt
StarsVP5.vpt
StarTrek25thB1.vpt
Stingray.10.vpt
Strange Days V1.0 (sphere 2001) .vpt
Strike Power (Bowling).vpt
Strike Power 1.1 Final.vpt
Strikes And Spares 20.vpt
Super Spin (Gottlieb 1977) 1.1.vpt
Super Spin (Gottlieb 1977).vpt
Super_Android Beta 0.9.vpt
Super_Android Beta 0.971.vpt
SuperBall_BKv1.1.vpt
SupersonicV1_3WithBox.vpt
SupersonicV2_5.vpt
Sure Shot (Gottlieb 1976).vpt
Surf!bgfx.vpt
SweetHearts_RC2.vpt
T2 - Judgement Day 1.3.vpt
T2vpm.vpt
Tales From Topographic Oceans beta2.vpt
Tales of the Arabian Nights RC2.vpt
Tales_Of_Arabian_Nights Final.vpt
Target.vpt
targetdropdemo.vpt
TargetVP5.vpt
Tarzon.vpt
tastysamba101.vpt
Taxi beta2a bjw mod.vpt
teeth.vpt
tgrrs17gs.vpt
The Addams Family 4.6.vpt
The Haunting 1.0.vpt
The Machine_BOP_Final.vpt
The RhythmApparatus Of Dr.FrankensteinV3.vpt
The Simpsons.vpt
The Skull V1.0 (sphere 2002) .vpt
The Who's Tommy 2.24.vpt
The_Shadow_v1_0.vpt
Theatre of Magic RC2.vpt
TheClockV10.vpt
thehunter10.vpt
Tiger Electronic Pinball - v2.0.vpt
Time Fantasy b2.vpt
Time Warp 1.1.vpt
TimeWar.vpt
TM_test1.vpt
tmnt_-_0.9_beta.vpt
Tomb Raider 4 V1.0 (sphere 2001).vpt
Tommy (VersionD).vpt
Top Card (Gottlieb 1974).vpt
Top Card (Gottlieb 1974)-df5.vpt
Top Score (Gottlieb 1975).vpt
TopSpeed.vpt
torture07.vpt
Tower Of druaga.vpt
Tower.vpt
TransformersV8.vpt
Travel Time (Williams 1973).vpt
TREXv2.1.vpt
tricktreat-v1.vpt
Tri-Zone (Williams 1979).vpt
Twilight Zone 3.9.vpt
Twilight Zone 3-8.vpt
Twilight Zone 4-0.vpt
Vampire.vpt
verspieldwn10.vpt
VHK Village.vpt
Video Pinball.vpt
Viking (Bally 1980)RC1.vpt
Warlok V12.vpt
wave race.vpt
WAY OUT WEST.vpt
Whirlwind 0.95.vpt
Whirlwind 0.9x.vpt
White Water 1-4.vpt
Who's Goofy (Ace Novelty 1932).vpt
whosgoofyVP5.vpt
winner11.vpt
Wishing Well (Gottlieb 1955).vpt
wizard.vpt
wizard8-10-01.vpt
WOLF3D.vpt
World Champ (Gottlieb 1957) reel.vpt
World Champ (Gottlieb 1957).vpt
WOW.vpt
WOWvb5.vpt
xenon vp.vpt
xenonvpm080.vpt
XenonVPMPlumbBeta3.vpt
XmenNonSBD.vpt
YellowSubmarineTB4(Spike2001)V1.vpt
ZAPTRAP.VPT
Zeux1.vpt
Zombie1_2.vpt

Back to Top