18. Optimizing Performance

Virtual pinball is in essence a video game, and video games place special demands on PC performance. The first prerequisite for adequate performance is adequate hardware, particularly the CPU and graphics card. You can find advice on selecting suitable components in Chapter 11, Designing the PC.

Even with fast hardware, though, you'll need to do some tuning to get the best video game performance out of your system. This chapter offers some advice on things to try.

Here's my quick list of the most fruitful optimizations with VP:

1. Minimize background tasks and other running programs (see General Windows optimizations)
2. Use a CPU affinity tool to give VP exclusive access to a group of CPU cores (see Controlling CPU affinities)

How to approach system optimization

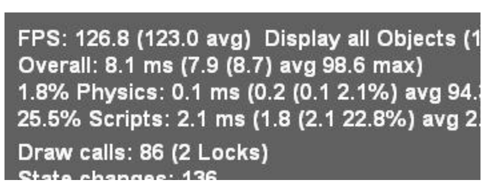
The whole point of optimizing is to make the game play better, in your subjective view. It's worth making a point to keep that in mind, because it's easy to get bogged down in the numbers, trying to make specific benchmark figures or performance metrics as perfect as possible. It's best not to get too obsessed with any one benchmark, because there's no benchmark that captures everything, and there's always a point of diminishing returns when you start focusing on making one number as high as you can get it.

That said, objective metrics are extremely helpful when making adjustments. For many of the adjustments you can make, there's no one-size-fits-all ideal setting, since the effects vary by system, so you'll have to experiment in many cases to find the right setting. You'll also have to experiment to see which types of adjustments make any sort of difference. Objective numbers can be really helpful to see if adjustments are having any effect and if they're moving the needle in the right direction. It's easy to trick yourself into seeing differences that aren't there if you don't have some kind of hard data to look at.

Measuring VP performance: frames per second

The most common performance metric in video games is the "frame rate", in Frames Per Second or FPS. This is the number of times per second that the game software can fully render the scene and update the video graphics.

VP can display the current frame rate and some other statistics on-screen while you play. Activate this display by pressing F11 while a game is running.



At first glance, it might seem strange that this isn't identical to your video card's refresh rate, usually 60 Hz (updates per second). The reason for the difference is that most video games (VP included) do their graphics rendering on their own schedule, according to how quickly they can do the computing work to produce each frame.

In order to produce smooth animation, VP's frame rate has to be at least as fast as the hardware refresh rate. If VP can't produce a new frame in time for the next hardware refresh, the TV will keep displaying the same frame as on the last refresh. This will make the picture momentarily freeze on the TV, which makes the motion appear jerky.

In practice, it's not good enough for the VP frame rate to merely be higher than the TV's refresh rate. It has to be *much* higher, by a factor of two or more. This is because the FPS rates that VP shows you is an average over many frames. The actual time it takes to produce any individual frame can vary quite a lot from one frame to the next. Some frames might take two or three times as long as the average frame to produce. If your average FPS rate is only marginally above the TV's refresh rate, all of those slower-than-average frames will take longer than one refresh cycle to generate, so you'll see a lot of repeated frames and jerky motion.

What makes a frame take longer than average? It's a matter of the complexity of the physics and graphics models that go into making up the frame. You'll typically see the frame rate slow down during multiball sequences, for example, since the physics engine has to compute the motion of the additional balls.

How to check your graphics card refresh rate

- Open the Display control panel in Windows
- Go to the Adjust Resolution section
- Click Advanced settings
- In the Adapter tab, click List All Modes

This will show you the list of screen resolutions and refresh rates that your GPU supports.

Most graphics cards and TVs use a standard 60 Hz video refresh rate. Many LCD TVs have nominal refresh rates of 120 or 240 Hz, but this isn't the video signal rate; it's just the LCD panel update rate, which uses interpolation to synthesize fake frames between the actual frames. A small number of TVs and computer monitors can accept true video signal rates higher than 60 Hz, but to take advantage of that you also need a video card that can generate such signals.

It's always best to use a graphics mode that exactly matches the native resolution of your TV: 1280x720 for a 720p TV, 1920x1080 for a 1080p TV, or 3840x2160 for a 4K TV.

How to reduce stutter

The jerky motion that happens when frames are repeated due to slow rendering is called "stutter". It's more or less impossible to make VP absolutely stutter-proof, since there are occasional oddball situations in any game where the physics computations get extremely complex and overwhelm even the fastest hardware. But it's at least possible to reduce stutter to the point where you'll hardly ever see it.

There are really two separate sources of stutter, which require separate solutions.

The first source of stutter is the raw computational time that VP spends updating the physics and rendering the graphics. The physics updates are done on the main CPU, and the rendering is mostly done on the graphics card (the GPU). So the obvious route to faster updates is faster hardware. For most people, that's something to consider in the planning stages, but it's not practical in terms of budget to update your whole PC every time a table is too slow. Barring hardware updates, the main way to increase VP's raw rendering rate is by adjusting VP's graphics options. We'll cover that below in VP video settings.

The second source of stutter is a little more subtle. Windows multi-tasks - runs many programs at once - by letting programs take turns using the CPU. Windows lets each program run for a fraction of a second, then interrupts it and lets another app take its turn. With most applications, you never notice this turn-taking, because it happens so quickly that it creates the illusion that every program is running at the same time. The illusion can start to break down with video games, though. The problem is that the turn-taking interruptions can happen at inopportune times that make VP miss its window for updating the graphics in time for a video refresh cycle, causing repeated frames and thus stutter.

One obvious way to reduce these interruptions is to minimize the number of other running programs, as outlined in General Windows optimizations below. That only goes so far, though, as you can't shut off everything else. To deal with the programs that have to keep running, there's a powerful technique known as CPU "affinity", which lets you partition your hardware and give special VP special access to parts of it. We'll talk about that later in Controlling CPU affinities.

General Windows optimizations

You can find lots of advice on the Web about general Windows tuning and gaming PC tuning. There's so much advice of that sort available that I won't try to reiterate it all in detail here, other than to mention the main points:

- Delete unused programs, especially any bloatware pre-installed by the PC vendor
- Disable unnecessary startup programs
- Remove or disable unnecessary background programs and Windows "services"
- Disable programs running in the "system tray" (the little icon area in the Windows task bar near the clock)
- If you're using Windows 10, turn off all of the cloud features in the system "Privacy" settings, to minimize background network access.
- Remove third-party antivirus/antimalware software

The last one (removing third-party antimalware programs) might make you uncomfortable. It's up to you, obviously, but I think it's worthwhile for a machine that you use as a dedicated pin cab PC and not as your main PC. There are two reasons I think you can go without. First, Windows 7 and later have pretty good protection built-in, in the form of Windows Defender. That's positioned for marketing purposes as "basic" protection only, to give you the impression you need something stronger, but it actually scores fairly high in most independent testing. Second, you can greatly reduce your exposure to malware by using the pin cab PC exclusively as a pin cab PC: don't store sensitive personal files on it, don't use it for email, don't use it for random Web browsing (limit Web use to trusted sites), and don't download random files (only download from trusted sites). Email and random Web browsing are the primary vectors for malware, so you can minimize exposure to avoiding those vectors as much as possible.

Controlling CPU affinities

The most powerful tool I've found for reducing stutter is CPU affinity. This a mechanism inside Windows for assigning each running program to a preferred group of CPU cores.

A "core" is a CPU within your CPU. The processor chips used in modern PCs, such as Intel i5 or i7 chips, are actually made up of multiple CPUs packed onto one piece of silicon. For example, an i5-8250 chip contains four complete CPUs. The term "core" is used to distinguish these CPU sub-units from the chip as a whole, which is also commonly called a CPU.

Windows has built-in support for multi-core chips. It automatically spreads work across the cores to optimize overall system throughput, and for most purposes you don't even have to think about it. As usual, though, video gaming doesn't exactly fit the typical program profile that the Windows default settings are designed for. The core affinity feature in Windows lets you override the defaults to optimize performance for special cases like games.

CPU affinity goals

The basic idea is to partition your CPU's cores into two groups: VP, and everything else. When you're running a table in VP, the game itself is the only performance-critical task in the whole system; everything else can take a back seat and wait its turn. So we're going to give the lion's share of your PC's computing power to the game, and give all other running programs the leftovers. For CPU affinity settings, the smallest unit we can work with when dividing things up is one CPU core, so if your CPU has N cores, we're going to allocate N-1 of the cores to the game, and give the one remaining core to everything else.

The point of this partitioning is to give the pinball software the most exclusive access we can to a set of CPU cores. This reduces the chances that another program running in the system will be able to interrupt VP or its components in the middle of some time-critical tasks. (And virtually everything VP does is time-critical, since it's a real-time physics simulation.)

In practice, I find that this makes a night-and-day difference in stutter, reducing it from noticeable to practically never on my pin cab.

CPU affinity tools

PinAffinity: This is a simple CPU affinity setter I wrote specifically for pin cabs. It's designed to be extremely simple to set up and completely automatic once configured, and it's free and open-source. You can find it here: PinAffinity.

Instructions for basic pin cab setup are included in the download, but here's a quick overview:

- Download the "bit" version that matches your copy of Windows (32-bit or 64-bit)
- Unzip the files into a folder on your hard disk
- Run PinAffinity.exe
- Use the "Add Program" menu to add the .EXE file for each pinball player program on your system to the designed Pinball program list
- Minimize the PinAffinity window and leave it running in the background while you play. It automatically sets CPU affinities for new processes as they're created.
- If you wish, you can create a shortcut to PinAffinity.exe in your Start Menu "Startup" folder so that the program automatically launches each time you boot

Other tools: On my own cab, I used to use a freeware program called PriFinitty. Unfortunately, it's no longer available; the developer abandoned the project a long time ago and never released the source code.

Another option is Process Hacker 3, available here: wj32.org/processhacker/. Process Hacker is a full Task Manager replacement, so it's not specifically designed for the pin cab use case, but it has the basic function we need (the ability to set CPU affinities persistently on a per-program basis). Note that you'll need a "nightly build" version of Process Hacker 3. The current public release version, Process Hacker 2, can control CPU affinities for live processes but can't save them or apply them automatically to new processes.

Recommended CPU affinity configuration

I recommend the following basic configuration:

- Assign three cores to VP (and any other pinball software you use)
- Assign the remaining cores to everything else

PinAffinity uses those settings by default.

Why three cores for VP? You might have read that Visual Pinball is single-threaded, so it might not seem like it would benefit from more than one core. It's true that VP's core physics and graphics run on a single thread, but if you look at a VP process with a tool like Process Explorer, you'll see that it has about 20 threads running. Where are they all coming from if VP is single-threaded? Mostly from the external subsystems that VP uses. VPinMAME runs on its own thread; DirectInput and DirectSound create multiple threads to service I/O events; DOF creates a thread for each output controller it accesses; and most video card graphics drivers create several additional threads.

The main VP physics/rendering thread and the VPinMAME thread each consume significant CPU time; the rest of the threads do little actual computing work, as they only exist to service I/O events and timed events as they occur. The VP and VPM threads probably don't come close to saturating the CPU on your machine; you'll probably see only 10% to 20% CPU usage on these threads. But even so, they both benefit from having a free core available because they both need "real time" responsiveness to keep up with external events. In the case of VPinMAME, it has to respond to game events immediately when they happen, and has to maintain precise time sync with the audio playback to prevent audible glitches in the soundtrack. In the case of VP, the physics/rendering thread has to keep in precise sync with the video refresh cycle; any lag in rendering is visible as stutter. It also has to respond quickly to input events to avoid perceptible latency (e.g., so that the flippers don't feel sluggish when you press the buttons).

That's why three cores seems to be the sweet spot for VP performance. We have two threads that run more or less continuously (the main VP physics/rendering thread, and the VPinMAME thread), and a bunch of other threads that need to respond quickly to events but do little work. If you give this collection of threads three cores to work with, Windows will be able to balance the load so that everything has near-real-time CPU access.

One video card or two?

One of the frequently asked questions on the forums is whether it's better to use a single video card that can support multiple monitors, or a separate video card for each monitor. (Most pin cabs have either two or three monitors: the main playfield TV, the backglass TV, and possibly a third monitor for the score display or "DMD" - the dot matrix display.)

This question actually contains two components, so let's unpack it. The first part is: can my video card handle multiple monitors? The second part is: would I get better performance by adding an extra video card for the second and third monitors?

The answer to the first part is basically always yes. All modern gaming cards have support for multiple monitors and provide built-in ports for connecting two to four monitors. And Windows has excellent, fully automatic support for multiple monitors built in. When you're setting up your system, there shouldn't be anything special you have to do with either your video card or Windows to configure multiple monitors; you shouldn't have to do anything more than just plugging them all in.

What about performance, though? Intuitively, it seems like more video hardware should translate to faster performance, for the same reasons that CPUs with more cores run faster. It seems like an extra card would take some load off the main graphics card. In practice, though, adding a second card makes most systems run slower. I can only speculate about the reasons for this, but I suspect that it has to do with contention for the data bus that connects the CPU and GPU. A second video card creates bus contention that doesn't exist if there's only one video card. Whatever the reason, most system in practice run faster with a single video card handling all monitors.

"One card is faster" isn't an absolute rule, though. I have heard from people who found that adding a second card actually did improve performance on their systems. But this seems relatively uncommon; for most systems, it seems that you'll get better performance by buying one fast video card than by trying to split the load across two or more cards.

Input lag

A common performance problem in video games is input lag: a noticeable delay between pressing a button and seeing the result on-screen. On a pin cab, this is mostly noticeable with the flippers. Input lag makes it feel like the flippers are slow to respond when you push the buttons.

Latency can come from many sources, but in most cases, the culprit turns out to be the TV. That's the place to focus your efforts to fix the problem, in part because it's almost always the biggest contributor to lag by far, and in part because there aren't really any adjustments to be made anywhere else in the system. Everything else is probably already running as fast as it can.

You can find more about input lag and how to minimize it in Chapter 7, Selecting a Playfield TV. We won't repeat all of the detail here, but the main point is that you should be sure your TV is set up with as little image processing as possible, especially enhancement modes related to "motion smoothing". Look for a Game Mode setting in your TV's setup menus; that usually selects the right combination of settings to minimize lag, since this is a concern for video games in general.

Audio lag

Audio lag is a noticeable delay between visual events appearing on-screen and the playback of the corresponding audio effect. This fortunately isn't a common problem. If you experience it, though, here are a few things to try.

Check your sound card's settings. (Open the Windows "Sound" control panel, select your card, and click Properties.) Make sure that everything is turned off under the Enhancements tab.

Some sound cards have extra properties tabs that control special hardware features of the card. Check any extra you see and make sure that any processing modes, effects, or enhancements are disabled.

If you're using an add-in sound card (rather than motherboard audio), and it came with its own separate settings program, go through that and make the same kinds of checks: look for any effects or processing modes that might be slowing things down, and disable any you find.

If you're using your TV's built-in speakers through the HDMI video connection, check the setup menus to see if there's a setting specifically for audio delay or audio sync. Your TV might be intentionally delaying the audio signal so that it syncs up with delays in the video signal processing, and it might let you adjust the delay time. If so, make it as short as possible.

VP video settings

Visual Pinball has a number of options related to graphics rendering that can affect performance. To access these options, start VP without loading a game, in editor mode. On the menu, select Preferences > Video Options.

The effects on performance of the different settings vary by system, and many of them trade off between quality and speed, so you'll have to experiment to find the ideal settings for your system. Here's an overview of the main settings that affect performance:

- Anti-aliasing: this controls extra graphics processing to make edges look smoother. Disabling it does the least processing, so it will be the fastest, at the expense of rougher looking edges. The performance impact of the different anti-aliasing modes depends on your video card.
- Ball reflections, ambient occlusion: these control extra rendering to create more realistic graphics. Turning them off results in faster rendering times.
- FPS limiter: setting this to 0 (the default) allows VP to render video frames as fast as it can. Setting it to 1 makes VP synchronize its rendering cycle with the actual video refresh rate. Some people find that synchronized rendering produces smoother graphics, so you might want to try it to see if it makes any noticeable difference for you. If not, I'd leave it at the default 0 setting.
- Exclusive full-screen mode: this makes VP take over the monitor completely when running in full-screen mode, rather than sharing it with other programs. This can improve graphics performance on some systems, but it can cause weird glitches with Windows multitasking, so I'd avoid it unless it makes a big difference for you.

19. Customizing VP Tables

One of the great things about the Windows virtual pinball environment is that so many tables are available in a format that lets you modify and customize them in any way you please. The free pinball player programs (Visual Pinball, Future Pinball) are also full pinball construction programs. You can open any table for Visual Pinball or Future Pinball in its editor and make your own modifications.

(The same can't be said of the commercial pinball games, such as Farsight's Pinball Arcade or Pinball FX/2 and FX/3. With those, you're stuck with what they sell you, which is never a perfect fit for cabinet play. That's one big reason that VP is so popular with pin cab builders.)

This chapter provides an overview of how to customize tables in Visual Pinball. We go into detail on a few of the key customizations often needed to adapt a table to cabinet play. Many of the tables available for VP were designed with regular desktop in mind, and weren't tested on a cab by their original authors, so they need some tweaking to look and play their best on a cab.

Opening a table in the VP editor

Visual Pinball is, at its core, a pinball construction program that also happens to let you play the tables. So editing a table is really the default thing that VP does.

In VP 8 and 9, the first thing you see when you start the program is a blank table editor window. Use the **File > Open** menu command to open a table in the editor.

In VP 10, they tried to make VP a little friendlier for the "average user", who just wants to play existing tables rather than creating their own. So VP 10 starts by bringing up a "Select Table File" dialog when you first start the program, and then immediately starts a game session with the table you select. If you want to edit a table instead, you have to bypass this initial dialog. Just click Cancel in the dialog, then use the **File > Open** menu to open a table in the editor, just like in VP 8/9.

Adjusting the viewing angle

When VP runs a game, the image you see on the display is a rendering of a 3D model of the table. To construct this image, VP uses an imaginary camera that views the model from a selected position in space. You can adjust this camera position to create different views of the table.

I find that most of the VP tables I download need some adjustment in their viewing angle to look their best on a pin cab display. And most pin cab builders feel the same way, because the ideal viewing angle is subjective. No one viewing angle will satisfy everybody.

VP 9: Adjusting the viewing angle in VP 9 is a bit tedious because you have to do it by typing numbers into a property sheet in the editor, and then run the game to test the results. I always have to iterate this process five or ten times before I find a satisfactory solution.

- Open the table in the VP editor
- On the left tool palette, click the **Backdrop** button
- Make sure the properties panel on the right is showing; if it's not, click the **Options** button on the left tool palette
- The properties panel should be labeled **Backdrop** at the top; if it's not, click in a background area of the editor window to select the backdrop

In the Backdrop properties window, the **Colors & Formatting** section contains all of the viewing angle options. The exact settings vary a lot from one table to the next, so there's no one-size-fits-all setting list I can give you. You'll just have to experiment with the different settings to see the effect they have. Change a setting and run the table to see the result. Change one thing at a time so that you can see each setting's individual effect.

Here's an overview of the individual settings and what they do:

- **Inclination:** The camera tilt, in degrees. 0 points the camera straight down at the table. Positive values tilt the camera upwards. For cabinet use, this should usually be close to 0.

- **Field of view:** The camera's viewing angle. This is analogous to a zoom lens on an optical camera. Zero produces an extremely flat view, like a telephoto image from a long way away; a high number (120-150) creates an exaggerated fisheye lens view. A value around 20 is usually good for cabinet use.
- **Layback:** The camera's distance from the front of the table. Higher values create a more tilted perspective. 0 creates a view from right over the center of the table. You want a value that places the camera a little ways out in front of the table, just like the normal viewing position for a player. A value of about 2/3 of the Y offset below is usually good.
- **XY Rotation:** For cabinet play, set this to 270.
- **X Scale:** This adjusts the table's size relative to the width of the monitor, which is confusingly the height of the table, when rotated 270 degrees for cabinet play. Adjust this to fit the table in the monitor across the monitor's width. This varies a lot by table, since it's a function of the playfield dimensions as well as the camera angle settings above. A value of around 1.4 works for many tables, but you'll have to fine-tune it for each table.
- **Y Scale:** This adjusts the table's size relative to the height of the monitor, which is the width of the table, when rotated 270 degrees for cabinet play. Adjust to fit. A value of around 2.0 works for many tables, but you'll have to fine-tune it for each table.
- **X Offset:** This adjusts the side-to-side position of the table, which is confusingly the vertical position on the monitor, when rotated 270 degrees for cabinet play. Adjust this so that the table is positioned properly. A value of around -450 works for most tables, but you'll have to fine-tune it for each table.
- **Y Offset:** This adjusts the top-to-bottom position of the table, which is confusingly the horizontal position on the monitor, when rotated 270 degrees for cabinet play. Adjust this so that the table is positioned properly. A value of around 50 works for most tables, but you'll have to fine-tune it.

VP 10: You can use exactly the same procedure as above with VP 10, but VP 10 also has an interactive "camera mode" that's a little easier to use. Camera mode lets you see the effect of each change immediately on the rendered table, without having to switch back and forth between editor mode and play mode repeatedly.

To activate camera mode, use the menu command **Table > Camera/Light Edit Mode**, or press F6. Follow the on-screen instructions to cycle through the settings and make adjustments. The settings listed above for VP 9 all have the same meanings here.

The camera mode controls are a little cumbersome, and it's hard to set exact values with them. You can always go back and fine-tune the values with the properties editor (using the VP 9 procedure above) to make any final adjustments.

Fake 3D table elements

One thing to note is that a lot of tables have some "fake 3D" table elements that don't respond well to viewing angle adjustments.

For example, consider Rudy's head in *Funhouse*. On the real machine, of course, Rudy is a rather large 3D chunk of plastic. But some VP versions of *Funhouse* don't use a 3D model object for Rudy; they just use a photo of Rudy pasted onto a flat surface in the VP model. That's what I mean by a "fake 3D" element: it's meant to look like it's 3D, but it's actually just a flat photo in the software.

The problem with these fake 3D objects is that the viewing angle captured in the photo will stay the same no matter how much you change the viewing angle of the table. The photo is, after all, just a photo. If you change the overall table viewing angle too far, it will become extremely obvious that the flat photo is now from the wrong perspective.

There are a few ways you can deal with this when you run into it:

- You can live with the distortion. The distortion will become more obvious the further you change the table viewing angle, so if you only need to adjust the angle a little bit, the distortion might remain tolerable.

- You can take or find a new photo from the new angle and replace the one in the table. This is tough unless you have access to the real machine, but you might get lucky and find a suitable image on the Web. You can find a lot of images on the Web for the more popular titles, after all.
- You can substitute a real 3D model (known in VP parlance as a "primitive") for the fake, flat photo. Ask on the forums to see if someone has already created one; there are 3D models of lots of pinball elements floating around (even unique ones like Rudy's head). Browse through some generic 3D model sites looking for something similar that you can adapt via Blender or SketchUp. If it's not too complex, create one yourself with one of those programs.

  Once you have a 3D object, you have to save it in the Wavefront ".obj" format. This is a common format that most 3D editors can save to. Next, create a "primitive" object in VP and import the .obj file. You'll also need a "texture" (an image that's projected onto the 3D surface to provide its coloration). The details are beyond the scope of this guide, but you should be able to get help in the forums if you're not familiar with VP primitives.

Viewing and editing the table script

Many customizations in VP are made through the table's "script". Every table has a script, which is basically a little computer program that carries out certain operations when you're playing a game with the table. It's called a "script" by way of analogy to the script for a movie or play. A movie script is a series of things the actors are supposed to say and do during the movie; a VP table script is a series of things the computer is supposed to do while the the is running.

To view a table's script:

- In VP 8/9, use the **Edit > Script** menu command
- In VP 10, use the **View > Script** menu command

That brings up a text editor window showing the script. You can simply type into this window to edit the code.

Table scripts are by their nature utterly unique, meaning there are no fixed patterns that they have to follow. However, there are certain conventions that many table authors follow, so you'll start to see patterns after you've looked at a few scripts.

VP scripts are written in the Visual Basic language. (Which makes for some confusing initials: VP scripts are VB scripts!) If you want to be more technical, VP actually uses a variant of Visual Basic called "Visual Basic Scripting" or VBS. Beware example code you find on the Web, because many Web examples of "Visual Basic" use a different variant known as "Visual Basic for Applications" or VBA. VBA is much more powerful, so unfortunately, many generic Visual Basic examples on the Web just won't work in VP's simpler version of the language.

Option variables

As mentioned above, many VP table authors follow common conventions and patterns for how scripts are arranged. One of these common patterns you'll often see is a set of "option variables" defined near the top of the script, that let you select some pre-programmed variations on the table's behavior. It's always a good idea to scan through the script for a new table you've installed to see if it has any option settings and customize them to your liking.

To see if a table has any option variables, read through the comments near the top of the script. A comment in VP starts with an apostrophe ('), and the VP editor usually shows it as green text:

```
' Funhouse / IPD No. 966 / Williams, November, 1990 / 4 Players
' VP9 12.0 by JPSalas 2009
```

Script options are typically defined as Visual Basic variable assigments or `Const` (named constant) definitions. Most authors group these near the start of the script, to make them easy for people to find without having to read through the whole of the script, and prominently label them with comments so that you'll know what they're for.

For example, here are the options at the top of *Whirlwind* for VP 9:

```
' *************************************************
' OPTIONS
' *************************************************

' Controller
' 1=VPinMAME, 2=UVP backglass server, 3=B2S backglass server
Const cController            = 3

' DMD rotation

Const cDMDRotation           = 0

' VPinMAME ROM name
' enter string of valid ROM
Const cGameName              = "whirl_l3"

' flasher and GI on or off option

' 0 or 1 to disable or enable the flashers
Const Flashers_ON            = 1

' 0 or 1 to disable or enable GI
Const GI_ON                  = 1

' some cabinet sound options

' 0 or 1 to disable or enable the flipper sounds
Const Flippers_Sound_ON      = 1

' 0 or 1 to disable or enable the slingshot sound
Const SlingShot_Sound_ON     = 1

' 0 or 1 to disable or enable the bumper sound
Const Bumpers_Sound_ON       = 1

' some more table options

' 0, 1 or 2 to set 'storm sound': 0 is off, 1 is fan, 2 is storm
Const StormMode              = 1

' 0 or 1 to disable or enable the "fan rotated" Williams W
Const RotatingWilliamsW_ON   = 1

' 0 or 1 to choose the standard or blue colored apron
Const BlueApron_ON           = 1

' 0 or 1 to aim the plunger outlane: 0 up the inner orbit or 1 up the ramp
Const Plunger2Ramp_ON        = 1
```

You don't have to be much of a programmer to know what to do with these: just change the number after the "=" in any line where you want to change to a different setting.

How to fix up tables for a real plunger

Many VP 9 tables require some scripting changes before they'll work properly with a plunger device. Most VP 10 tables work with plungers automatically, but you might run into a few that need the same kind of fixup as is often needed for VP 9. The changes are sometimes fairly complex, so we cover this topic in a separate chapter: Chapter 39, Fixing VP Plungers.

How to enable B2S backglasses

Most VP 10 tables will work with B2S without any modification, as will some later (2016+) VP 9 tables. Earlier VP 9 tables often require some slight modifications to the table script to enable backglass art, though. See Chapter 17, Backglass Software Setup for details.

How to play table sound effects through the backbox speakers

If you have a separate set of playfield effects speakers inside your cabinet, VP decides whether to use your main backbox speakers or your playfield effects speakers as follows:

- If the sound comes from the game's ROM (the original game's software, being emulated in VPinMAME), it's played through the backbox speakers
- Otherwise, it's played through the playfield effects speakers

If you don't have a separate set of playfield effects speakers, all sounds are played through your main speakers. See "Playfield effects speakers" in Chapter 41, Audio Systems for more about setting up the extra speakers.

Assuming you do have playfield effects speakers, you might want to override the rule about playing all of the non-ROM sounds through the playfield effects speakers. VP lets you override it on an effect-by-effect basis.

First, let's think about why the rule is set up this way in the first place. The ROM soundtrack is the game's original soundtrack from the arcade game, so on the *real* version of the machine, all of the sounds from the ROM were played back through the real machine's backbox speakers. So it makes sense that we'd want to do the same thing in a virtual cab. What about the "non-ROM" sounds? Those are sound effects that the VP table author added into the simulation of the table. These are almost all meant to simulate the sound made by something mechanical on the playfield, like the ball rolling around and bumping into things, bumpers bumping, etc. So in almost all cases, you want these to sound like they're coming from the playfield area rather than from the backbox.

Now let's think about why you might want to override this for some sounds. Occasionally, you might have a mechanical sound that actually would have come from the backbox on the original real machine. For example, some EM-era machines had scoring bells situated in the backbox. Likewise, any simulated score reel sounds ought to come from the backbox area. In addition, some tables might have the occasional added voice or music effect that supplements the game's original ROM soundtrack, so you might want these to play through the backbox speakers as though they were part of the ROM soundtrack.

In VP, table sound effects are tied to one or the other set of speakers (playfield or backbox) on an effect-by-effect basis. All of the sounds are initially set to play through the playfield effects speakers. To change an effect to play through the backbox speakers instead, here's the procedure:

- Launch VP
- Open the game in the VP editor (don't run it)
- On the menu, select **Table** > **Sound Manager**
- Find the sound you want to redirect to the backbox speakers and select it in the list; you can use the Play button to listen to each sound if you're not sure it's the one you're looking for
- Check its current speaker assignment:
  - In VP 9, if the "Import path" looks like a regular file name, it's assigned to the playfield effects speakers; if it says ***Backglass Output***, it's assigned to the backbox speakers
  - In VP 10, the "Output" column will say either **Table** (plays through the playfield effects speakers) or **Backglass** (plays through the backbox speakers)
- If it's not already on the backbox speakers, click **Toggle BG Out** (VP 10) or **To BG Out** (VP 9)

If you want to go the other direction - change a sound that's already on the backbox speakers to use the playfield effects speakers instead - the process is exactly the same with VP 10. Just select the sound in the list and click **Toggle BG Out** to switch it back to **Table** mode. The process in VP 9 is rather ugly: you have to export the sound effect to a WAV file and re-import it. What's more, some VP versions have a bug that won't let you export a sound that's been set to the backglass output, so you're kind of stuck. The best workaround would be to download a fresh copy of the table, export the sound from that fresh copy, and import the sound into your modified version of the game.

What about changing some of the ROM sounds to play back through the table effects speakers? Sorry; it can't be done. All of the ROM sounds are handled by VPinMAME, which doesn't have any options for changing the speakers for a specific sound. Remember that the ROM software is more of a "black box" than a VP table, since it's emulating an old arcade machine that didn't work like a PC with modern abstractions like WAV files. VPinMAME doesn't have any way to make a simple list of the sounds in a ROM that you could use to choose speakers like you can with the table sounds in VP.

How to enable DOF

DOF support is similar to B2S support: for most VP tables and some later VP 9 tables, DOF support is automatic, whereas earlier VP 9 tables usually require some script modifications. See Chapter 46, DOF Setup for details.

Removing sound effects for DOF play

If you have DOF mechanical feedback devices (solenoids, gear motors), you'll usually want to disable the digitized sound effects that tables play back to simulate the same events, since the digitized sounds tend to sound fake (not to mention redundant) when real mechanical devices are firing at the same time. Chapter 46, DOF Setup describes how to remove the unwanted sound effects.

How to fix EM tables that use the wrong coin keys

Some re-creations of EM (electro-mechanical) tables use the "wrong" keyboard keys for some functions, especially the coin-in buttons. If you're having problems with an EM game where it won't respond to your pin cab's coin buttons, this might be the cause.

The reason you see this in EM tables in particular (as opposed to more modern "solid state" games - the type with electronic displays of some kind) has to do with VPinMAME. VPinMAME is the part of the Visual Pinball system that normally handles most of the keyboard functions, including coin handling. The thing is that EM re-creations don't typically use VPinMAME, because VPM's function is to emulate the original ROM software from an electronic game. Part of the definition of "EM" is that it doesn't have any software, ergo no VPM involvement. And without VPM, it's completely up to the table script to handle all of the keyboard interaction, including the coin keys. EM table authors often hard-code the coin function to a specific keyboard key, which might not match your pin cab's button setup.

Fortunately, it's not too hard to fix these when you find them. The procedure is to find the place in the table's script where the coin key is handled, and change the script to test for the correct key.

- Open the table in the VP editor
- Open the table's script
- Search for the string "_KeyDown". This should take you to a line that looks like this:

  ```
  Sub Table_KeyDown(ByVal keycode)
  ```

- Note that the "Table_" prefix might be different in the actual table, but the rest should be the same. This is the start of the key handler subroutine. The code we're looking for now is somewhere in this subroutine, which is all of the code up until the next line like this:

  ```
  End Sub
  ```

- Most people indent the code in this section to make it easier to see that all of the code up to the `End Sub` goes together.
- At this point, you'll have to read through the code to find the section that handles the coin input. Hopefully, the table author will have put in a comment, or at least used well-named variables. Look for the words "coin", "credit", or maybe something like this:

  ```
  Credits = Credits + 1
  ```

- If you can find the right section, it should be preceded by a test for the key code. That will usually look like one of the following:

```
Case 6:
If KeyCode = 6 Then
```

- The number after "Case" or "Keycode=" might be different. It's usually 6, which is the scan code for the "5" key on the keyboard (confusingly!), since that's what most desktop users expect for the coin-in key. It might also be 4 (the scan code for the "3" key), since that's another common coin-in assignment.
- If you find that line, change the number to the word `AddCreditKey`
- Close the script and save the table

The special symbol `AddCreditKey` is VP's way of referring to the key assigned to the coin function in the VP option settings. If the script was using a hard-coded scan code, this change should make the table use the correct key as set in the options.

20. Cabinet Building Tools

It's hard to overstate the importance of using the right tool for a given job. Good tools can make a seemingly difficult task easy, and can let an amateur produce professional-looking results. Here are some recommendations for the tools needed to build a virtual cab.

Basic hand tools

These core tools are needed for the most basic DIY projects. You'll probably already have most of them on hand for routine home maintenance needs. You'll probably need most of these even if you're starting with a pre-assembled cabinet, and you'll certainly need them if you're assembling a cabinet from a flat-pack kit or from scratch.

- Screwdrivers: a basic set of Phillips and flat-head screwdrivers in assorted sizes
- Hex nut driver set with assorted English and metric sizes
- Hammer
- Pliers
- Needle-nose pliers
- Sheet-metal shears
- Assorted wood and metal files

Basic power tools

Some basic wood-working power tools are good to have on hand even if you're starting with a pre-assembled cabinet, to facilitate finishing work and simple customizations.

- Drill, with assorted wood bits from ⅛" to ½"
- Power screwdriver (completely optional, but it can be a huge labor saver)
- Power sander (essential for surface preparation if you're going to apply paint or decals)

More power tools

If you're building from a kit or from scratch, you might also want:

- Jigsaw
- Circular saw

Router

If you're building a cabinet from a flat-pack kit or from scratch, it's good to have a router on hand. A router is a versatile power tool with a high-speed rotating bit that can move over a piece of wood to cut grooves, holes, and edges. Some of the things a router can help you accomplish:

- Cutting custom-shaped holes
- Forming joinery edges (bevels, miters, rabbets, dados)
- Cutting grooves
- Routing out depressions or hollows

If you're starting with a flat-pack kit, most of the cuts and joinery edges should be pre-cut, but a router is still useful for a few tasks that the kits usually leave for you to do. In particular, a router is required to cut the edge grooves needed to install the plastic holders for the playfield glass, and you can also use it to cut custom holes for speakers, fans, and buttons. For this type of light usage, a hand-held router is adequate; good options are available for under $100.

If you're building a cabinet from scratch, a router is required for creating cabinetry-style joins at the corners. For joinery work, you'll probably want a table router. You might want to buy an inexpensive hand router as well, since they're easier to use for some types of jobs (such as cutting small openings).

Recommended bits:

- For general hole-cutting and dados, a basic set of straight bits in assorted sizes.
- For installing the playfield glass guides, a slot cutter bit with a 3/32" groove width. Freude makes a suitable groove cutter with a 9/16" groove depth and 3/32" grove width (part number 63-106).
- If you're building from scratch, you can buy bits for the types of joins you plan to use.

Electronics

- Soldering station. If you're doing even simple electronics work, it's worth investing in a decent soldering station. A soldering station is different from a basic soldering iron in that a station has a thermostat that controls the tip temperature, which maintains consistent soldering conditions. Stations also heat up much more quickly and have much better tips than cheap soldering irons. I'm very happy with my Hakko FX88D (available for under $100). If you've been frustrated in the past trying to do soldering work with a cheap iron, and you think it's because you don't have the right skills, you'll be amazed at your overnight transformation into a soldering genius when you switch to a proper soldering station.
- Solder. Another thing that will amaze you by improving you soldering skills overnight is to switch to a good solder. The stuff they sell at Home Depot might be okay for plumbing and other rough work, but it's not very good for electronics. The type I like is Kesler 44 63/37 Sn/Pb rosin core solder.
- Digital multimeter. An essential tool for troubleshooting electronics. The main functions I use regularly are continuity testing, voltage, resistance, and current. Virtually ever meter available will have these basic functions.

  One feature you should definitely look for is "auto-ranging". That means that the meter automatically senses the order of magnitude of the reading for each input type (rather than requiring you to select the range with the dial). The cheapest meters (in the $10 range) lack auto-ranging. It's worth a few extra dollars to get this feature.

  I don't have any specific brand recommendations. My professional electrical engineer friends have always sworn by their Fluke meters, and I don't doubt they're the best, but they're quite pricey. You can find less prestigious brands with all of the needed features for as little as $20. I think that even the cheap meters are mostly pretty good at this point, thanks to the relentless march of progress on digital electronics. I'd buy based on price and user reviews.

21. Cabinet Body

We turn now to what is arguably the essential element of our project, the thing that makes it special and different from video-game pinball on a PC: the pinball cabinet.

This section presents detailed plans for building, from scratch, a replica of a 1990s-era pinball Williams pinball cabinet, with some small changes to accommodate the virtual PC-and-monitor setup instead of a mechanical pinball table. Before we get to the building plans, though, we'll discuss the reasons behind the design of the cabinet, and some important things you should consider before finalizing your own design. If you want to jump straight to the measurements for cutting up the plywood, skip ahead to A plywood cutting plan.)

The goal for most of us is to replicate as closely as possible the exterior appearance of a real pinball machine. That's what I set out to do with my own virtual cab, and this section is based on the idea that you're doing the same thing. This section is therefore basically a guide to building a replica of a Williams pinball machine cabinet from the 1990s, using the same materials and parts. The plan here isn't an *exact* replica, because a few customizations specific to virtual pinball are needed, but most of the differences are small. For the most part, you could use the plans here to build a replacement cabinet for a real machine (and just in case you really want to do that, I've tried to point out where my plans diverge from the WPC design).

When I was building my own virtual machine, I discovered that the design of the 1990s pinball cabinets is something of a secret art. It's not secret by any intention or conspiracy, exactly; it's just that there's a lot of knowledge and experience that went into the design that no one wrote down anywhere. Not anywhere we can look at, at least; I'm guessing there actually are a lot of old engineering diagrams and blueprints locked in a basement file cabinet somewhere, but those documents aren't available to hobbyists.

Owners of the real machines can absorb a lot of the design details of the original cabinets through observation, but if you don't have a real machine to take apart and examine, you pretty much have to guess. I'm fortunate to have some real machines at home, and that turned out to be a huge help for building a virtual cab. Whenever I was unclear on something, I could look at a couple of actual examples to see how they did it, and see some of the variations in the evolving design. I took advantage of that many times. So my goal in this section is to pass along as much of that otherwise undocumented knowledge as I can, in this one place, in an order that essentially provides a recipe for building one of these cabinets.

I know that not everyone wants their cab to look exactly like a 1990s machine. You might want to base your cabinet on the pinball machines of some earlier era, for example, or you might want to create something completely novel. Even if you're aiming for a different look, though, you might find it helpful to understand how the 1990s WPC cabinet design works. Many of the elements represent decades of refinement of solutions to the mechanical and spatial constraints common to most pinball machines. You can always look to the WPC cabinet for ideas on how to solve those sorts of constraints in your own design.

Build, buy, or convert?

There are three main options for creating the body of your cabinet:

- Build it yourself from scratch
- Buy a new empty pinball cabinet, pre-built or as a kit
- Convert an old real pinball machine into a virtual cabinet

If you're new to virtual pinball, the approach that might seem most appealing at first glance is to convert an old real machine. Indeed, this was the most popular approach in the early days of virtual cabs. It has the advantage that it's already a perfect match to the real thing from the very outset. It also saves you the trouble of coming up with the cabinet design from scratch - figuring dimensions, identifying and source parts, etc.

The conversion approach was popular for a while, but it's become a lot less appealing lately because of increasing prices. Pinball has become a collector's item, so even a beat-up old machine can command a pretty high price from someone looking for a restoration project. If you're lucky enough to find a donor machine that no one wants to restore, it'll probably be in such bad cosmetic condition that it might actually be cheaper and easier to start from scratch. (You should also be prepared for some negative comments on the forums, since there's a growing preservationist sentiment, even among the virtual pinball crowd. Many people see pinballs from past decades as works of historical significance that can't be fully or genuinely re-created once lost.)

Happily, the "build" and "buy new" alternatives are both quite practical, and I consider both of them superior to conversion, even without the price considerations. You can buy high-quality reproduction cabinets that look exactly like the real thing, and you can get these in kit form or fully assembled. Or, if you have some basic woodworking skills, you can build an excellent reproduction cabinet yourself. The real machines use fairly simple designs that you don't have to be a master carpenter to reproduce. This section provides detailed plans for building a faithful reproduction of the late-model Williams cabinet design.

What's more, it's easy to obtain all of the genuine cabinet hardware (metal rails, legs, etc) needed to fit out a custom-built cabinet and make it look exactly like a real machine. All of the hardware is very standardized across machines, and several big online pinball suppliers sell the parts. You can thank the collectors for that - they buy these parts to repair and restore their machines, so there's a healthy market in the parts that keeps them readily available.

Here are my recommendations:

- For most people, I recommend using a VirtuaPin flat-pack kit. I used this approach myself, and I'm very happy with the results. It's not the cheapest option, but it's reasonably priced, and it yields an excellent finished product without requiring any real woodworking skills. With a little care, the result will be essentially indistinguishable from a brand new real pinball machine cabinet.
- If you enjoy wood-working and have a well-equipped workshop, consider a scratch build. Doing it yourself is cheaper than buying a VirtuaPin flat pack, assuming you already have the tools needed, and you should be able to get equally professional results with a little care. Pinball cabinets are relatively simple as carpentry projects go; they're built out of ordinary plywood, and the woodworking involves only straight cuts and some basic joinery work. But even though the wood-working is pretty simple, you'll need to execute it with precision, so this is best if you already have at least a modest amount of wood-working experience.
- If you don't want to do any woodworking, and you don't even want to assemble a kit, you can order a fully built cabinet from VirtuaPin or from a number of other vendors. VirtuaPin's pre-assembled product is the best one I've seen, and it's the only one I know of that faithfully reproduces the design of the real pinball machines. (In fact, it's such a good replica of the Williams 1990s design that some collectors restoring real machines buy them as replacement cabinets.) The products I've seen from other vendors use idiosyncratic designs and non-standard trim hardware, so they don't look to me quite like the real thing. If that's important to you, take a close look before buying to make sure you like the design.
- I generally don't recommend trying to re-purpose an old real pinball as a virtual cab, in part because old cabinets tend to be so beat up that restoration would be more labor-intensive than building a new one, and in part because the economics rarely pan out. More on that below.

Economics of new vs used

If you're set on the idea of re-purposing a used cabinet, I'd suggest doing a little research first to make sure you don't overpay. The question you want to ask is: would I actually save money buying the used cab, or would it be cheaper to buy the same parts new?

To answer this, ask the seller for a list of all the hardware parts that the used cabinet includes - the legs, side rails, lockbar, etc. Don't assume that everything is included, because a lot of eBay sellers strip all of the parts out and sell them separately; a used cabinet might not come with anything beyond the wood box. And only consider the hardware that you'll actually use on your virtual cab, since those are the only parts you'd have to buy if you were starting from scratch. Only count parts that you actually need in the virtual cab (e.g., don't count the playfield, bumper caps, etc), and only count used parts if they're in usable condition.

To help you get started here's a list of the main parts that real machines and virtual cabs have in common. See Chapter 10, Cabinet Parts List for a more detailed parts list with descriptions. We left the price column blank, since prices obviously vary over time and from one vendor to the next, so you'll have to fill that in by checking current prices at your preferred vendor(s) (such as VirtuaPin, PinballLife, or Marco Specialities). For the "wood body" line item, you can use VirtuaPin's flat pack or unfinished cab body offerings for comparison. Remember, only include the parts that the seller is including with the new cab, since you want to compare new-vs-used for what you're actually getting from the seller.

Add up the prices of the new parts, and compare the result to the seller's asking price for the used cab. If the asking price for the used cab is cheaper than the new parts, and everything's in good enough shape that you can actually use it, you've found a good bargain. If the seller is asking more for a beat-up used cab than what you'd pay new, I'd pass on the deal.

☑	Description	Price New
☐	Main cabinet wood body	$
☐	Backbox wood body	$
☐	Legs (qty 4)	$
☐	Leg levelers ("feet") (qty 4)	$
☐	Leg brackets (qty 4)	$
☐	#8 x 5/8" wood screws for leg brackets (Williams ref 4108-01219-11, 4608-01081-11), or #10 screws if preferred (qty 32)	$
☐	Leg bolts (⅜"-16 x 2¾" or 2½") (qty 8)	$
☐	Side rails (qty 2)	$
☐	Lockdown bar	$
☐	Lockdown bar receiver	$
☐	Coin door	$
☐	Coin acceptors ("coin mechs")	$
☐	Cashbox tray (Williams ref 03-7626)	$
☐	Cashbox lid (Williams ref 01-10020)	$
☐	Cashbox nest bracket (Williams ref 01-6389-01)	$
☐	Cashbox lock bracket (Williams ref 01-10030)	$
☐	Carriage bolts, black, ¼"-20 x 1¼" (qty 6: 4 for coin door + 2 for lock bar)	$
☐	Flange locknuts, ¼"-20 (qty 6: 4 for coin door + 2 for lock bar)	$
☐	Top glass	$
☐	Rear plastic channel for glass	$
☐	Side rail plastic channels for glass (qty 2)	$
☐	Plunger (ball shooter) assembly	$
☐	Ball shooter mounting plate (Wiliams ref 01-3535)	$
☐	#10-32 x ¾" bolts for mounting plunger assembly (qty 3)	$
☐	Backbox hinges (qty 2)	$
☐	Backbox hinge backing plates (qty 2)	$
☐	Carriage bolts, ¼"-20 x 1¼" (qty 6, for backbox hinges)	$

☐	Flange locknuts, ¼"-20 (qty 6, for backbox hinges)	$
☐	Pivot bushing carriage bolts (qty 2)	$
☐	Hex pivot bushings (qty 2)	$
☐	Backbox latch	$
☐	Backbox latch bracket	$
☐	Backbox lock plate assembly	$
☐	U-channel, metal, ⅝" x ⅝" x 27⅛" (backbox speaker panel holder)	$

Where to find used machines

Your best bet for finding a used machine at a good price will be local sellers. Search your local craigslist and local newspaper classified ads. A particularly good place to find a deal is at an estate sale. Heirs often want to clear out the house quickly and won't have any sentimental attachment to an old pinball.

If you can't find anything locally, eBay will give you access to sellers nationally (and even internationally). But I wouldn't get my hopes up; it's hard to find a good deal on a used cab on eBay these days. For one thing, shipping a cabinet is expensive due to the size and weight; shipping can add about $400 to the price. For another, eBay sellers know they can get top dollar for used cabs, so the base price is unlikely to be a bargain. Most eBay sellers are also savvy enough to strip a machine of all of the parts and sell them off separately, which largely defeats the purpose of reusing an old machine.

Kits and pre-built cabinets

If you can't find or don't want to use a salvage machine, but you also don't want to build everything from scratch, there are several companies that will sell you a brand new cabinet. For options, search the web for "new pinball cabinet" or "pinball cabinet restoration". One vendor I can recommend from personal experience is VirtuaPin. They sell cabinets in both kit form and fully assembled, and can customize them to your specifications.

You should be able to find options ranging from "flat pack" kits that you assemble yourself, to fully assembled cabinets with all of the hardware and artwork installed.

The cheapest and most DIY option is a flat pack kit. This is like an Ikea bookshelf: it consists of the wood parts, pre-cut and pre-drilled, ready for you to assemble. This is the cheapest kit option, since you provide the assembly labor, and because shipping is cheaper than for a bulky assembled cab. The degree of difficulty is slightly higher than for assembling Ikea furniture, but only slightly; no real woodworking skills are required, and you'll just need basic tools like screwdrivers and hammers. You might also have to do some sanding to even out corners and edges. And you'll have to do your own finish work (painting, staining, or applying decals). I used the VirtuaPin flat-pack kit for my own cabinet, and I highly recommend it. They use a good furniture-grade plywood, and the design is an extremely faithful reproduction of the WPC cabinets that Williams shipped in the 1990s, so the result is exactly like a brand new real machine.

The next step up in price and completeness is an assembled cabinet shell. This is just the wood shell, typically unfinished (ready for you to paint, stain, or apply decals), and without any of the cabinet hardware accessories installed. This eliminates the assembly work required for a flat pack. It's considerably more expensive to ship because it's so bulky.

At the high end price-wise, you can buy a fully assembled cabinet with all of the hardware and graphics pre-installed. VirtuaPin sells these in addition to their flat-pack and assembled-but-unfinished products. All of the VirtuaPin options use the same materials and design as their flat pack, so they all yield excellent reproductions of the 1990s Williams machines. There are other vendors selling pre-built cabinets as well, but check their designs carefully before buying, because I've seen a couple of other vendors who use their own ad hoc designs that look a bit cheap and cheesy to my eye.

Tip: Ask about the button hole layout. If you order a kit or pre-built cabinet, ask the seller for details on the locations of button holes they pre-drill, and ask them to customize the drilling to your plans if you have something else in mind. The vendor might drill holes by default that you don't want. Pay particular attention to the placement of the plunger and flipper button holes. Many virtual cab builders choose non-standard locations for these to accommodate the playfield TV (see "The dreaded plunger space conflict" in Chapter 29, Playfield TV Mounting and "Positioning the plunger" in Chapter 37, Plunger). If you're not sure how you want to handle this at the time of your order, you can simply ask the vendor not to drill any holes for the plunger or other controls. That will give you the flexibility to drill them yourself later when you know how everything will fit together.

Scratch build

If you have the necessary wood-working skills and tools, the cheapest and possibly best option is to build the whole thing yourself from raw materials. It's easy to get all of the parts and materials needed for an extremely accurate replica.

Here are the recommended wood materials:

- ¾" (nominal) plywood, for almost all of the main cabinet and backbox. Choose a high-quality plywood that's graded for furniture or cabinetry use. The ¾" thickness is important, as many of the trim parts and controls (like flipper buttons and plunger) are designed to fit into ¾" walls. It's just barely possible to fit everything into a single 4' x 8' sheet **if** you make the cabinet floor out of something else (such as particle board). If you have the budget, I'd recommend getting two sheets of plywood and using it for the cabinet floor. This cabinet floor will be stronger, plus you'll have enough plywood left over for one or two "do-overs", which is a nice bit of insurance in case you make any mistakes or find any cosmetic flaws in the plywood.
- ¾ particle board or fiberboard (e.g., MDF), **only if** you want to use it instead of plywood for the cabinet floor. The real pinball machines did this to save on cost, since the cabinet floor isn't a cosmetic element and no one cares if it looks cheap. Even so, I'd use plywood instead, since it's lighter and stronger. Particle board is likely to sag over time, and it doesn't hold screws very well.
- ½ (nominal) plywood, for the back wall of the backbox. A 4' x 4' half sheet is sufficient.
- A length of 2x2 (nominal) wood strip, in a soft wood like pine, for some miscellaneous parts. (A nominal 2x2 is actually 1.5" on a side.) You'll only need about two feet of this for the parts we make in this section, but I ended up using a bunch of this in my own cab for improvised connectors and supports and what have you, so you might want to pick up a few strips to have on hand.
- A length of 1x2 (nominal) wood strip, in a soft wood, for a trim piece in the backbox. (A nominal 1x2 is actually .75" x 1.5".) You'll need about 3'.

Some people use MDF (fiberboard or particle board) instead of plywood for the entire cabinet. I strongly recommend using plywood instead. It's what all real pinball machine bodies are made of, and it's simply a much better material for this job. The main virtue of MDF is that it's cheap. It's also very uniform, which makes it attractive for some applications, but that's not particularly helpful for this project. The downsides of MDF are that it's very heavy, and it's not as sturdy or as durable as plywood. The weight difference is a significant factor for this project because it's a very large piece of furniture; a pinball body built from plywood is already plenty heavy.

(Despite what I just said about the evils of MDF, the original William cabinets actually did use particle board in two places: the cabinet floor, and the back wall of the backbox. I have to assume this was a cost-cutting measure, based on the idea that these surfaces don't have to look pretty because they're not visible to players. I'm normally all for meticulous adherence to the originals, but in this case I don't see any merit to it; the originals would have been better if they'd used plywood instead, as it's not uncommon in old pinballs for the MDF floor to sag in the middle.)

"Nominal" in lumber dimensions means that it's what the lumber yard calls it, but it's not the true size. Plywood is generally 1/32" thinner than the nominal thickness, and dimensional lumber, such as 1x2 or 2x2 strips, is generally 1/4" to 1/2" less in each dimension than the nominal size. So when the list above calls for 3/4" plywood, it means you should buy what the lumber yard calls 3/4" plywood, even though the true thickness will be slightly less than that.

For the woodworking, you'll need some basic power tools:

- Drill
- Wood-drilling bits, various sizes from 1/8" to 1/2"
- Hole saws or Forstner bits, 1" and 1⅛" diameter. These are special types of drill bits, for standard portable drills and drill presses, designed to drill clean holes at larger diameters. You'll want to use these for anything larger than about 1/2" diameter. Spade bits are also available in these sizes, but they're not good for plywood because they cause a lot of chipping and tear-out. A hole saw or Forstner bit will produce much cleaner results.
- Jigsaw
- Plunge router. This isn't something most casual woodworkers own, but if you do buy one, I expect that you'll find it so useful that you'll wonder how you ever lived without it. A router is pretty much required if you're going to build a cab from scratch. A plunge router is a hand-held router whose bit can move perpendicular to the work surface, so that you can position the bit at a starting point above the surface, and then "plunge" it straight down into the surface to begin cutting. Routers are good for cutting grooves, insets, and openings in arbitrary shapes. They can also cut the edges of a board into different shapes, for joining pieces at corners.
- 3/32" slot cutter bit for your hand router. This is a special type of router bit that's needed to cut slots along the top edges of the side walls, for installing the playfield glass channels. See Freude part number 63-106 for an example.
- Circle jig for your hand router. This is an attachment for the router that lets you cut circular patterns in a range of sizes. This is good for cutting large circular openings (larger than a drill bit can make). We'll need this to create openings for the subwoofer and the ventilation fans.
- Table router. This is useful for cutting long straight lines and for making joinery edges. A hand router can do these jobs, too, but some tasks are easier with a table router.
- Circular saw, table saw, or track saw, with a fine-tooth blade. This is optional, but it's what most people use for the long, straight cuts for the main pieces. You can alternatively use a jigsaw or router instead.

  The default saw blades that come with most circular saws are for coarse cuts, and they won't work well with plywood. So you'll probably also need to buy an extra blade that's specifically designed for plywood and designated as a "finish" blade. The manufacturer should list the materials that the blade is designed for in the product description. "Finish" blades are designed to make clean cuts with minimal chipping; they usually have a lot of small teeth.
- Power sander. There's enough sanding needed in a cab build that the job will be infinite if you try to do it by hand. A power sander makes much lighter work of it and produces better results.

You'll also need the cabinet hardware parts. Many of these are specialized pinball parts, which are available as replacement parts from arcade supply vendors. See Chapter 10, Cabinet Parts List for the full parts list, with part number references for standard pinball parts.

I personally prefer to use real pinball parts wherever possible, instead of trying to improvise something out of common hardware parts. It can be a bit more expensive to use the real parts, but they tend to look better, and in many cases they're easier to work with because they're purpose-built for a specialized job. That said, a fair number of the parts that go into a cab are truly generic hardware, like nuts and bolts, that you can buy anywhere; there's no reason to buy special pinball-certified #8 machine screws, for example. But even some of the generic-sounding hardware can be hard to find outside of pinball vendors. Anything that I listed with a Williams part number in the master list (Chapter 10, Cabinet Parts List) is probably one of those hard-to-find items.

Choosing a cabinet design

As far as I'm concerned, there's only one cabinet design that we need to concern ourselves with: the "WPC" cabinet. This is the cabinet that Williams (and its co-brands Bally and Midway) used for all of their machines in the 1990s. The name comes from the electronics platform used in that generation of machines, which was called the Williams Pinball Controller or WPC.

One good reason to use the WPC cabinet design is that it's by far the easiest to find parts for. Machines from this generation are still widely deployed and remain some of the most popular pinballs of all time, so there's plenty of demand for replacement parts from owners of the real machines. If you design to the WPC specs, you'll be able to use these readily available parts to outfit your machine. Using the real parts will give your machine a completely authentic look, and it's a lot easier than fabricating your own custom metal parts.

Another reason to use the WPC cabinet plan is that it's the same plan used by most real machines from the modern era. Williams used it for nearly all of their 1990s machines. Stern's 2000s machines are built to almost identical plans, as are the machines from the boutique pinball makers like Jersey Jack and Spooky Pinball. This cabinet style is what you'll see almost every time you encounter a late-model pinball in an arcade or bar. A virtual cab following the same plan will look exactly like what everyone expects a real pinball machine to look like.

For a DIY project, you're always free to come up with something completely different, either to fit your particular needs or purely to be unique. I personally place a lot of value on simulation fidelity, so I like the idea of a virtual machine looking as much as possible like a real machine. But that's just me; you might have other priorities or different taste. If you have other ideas for how your cabinet should look, you can take as much or as little from the WPC design as suits you.

We provide detailed plans for the standard WPC design later in this section. Before we get to the plans, though, there are some variations that you might want to consider, so that you can customize the plans for your project's specific goals.

Standard and Wide-body cabinets

The WPC design is what we usually call the "standard" cabinet, because Williams/Bally/Midway used this for most of the machines they shipped in the 1990s. However, they also used a variation of this plan that differed by making the main cabinet wider. This wider variation was used for seven titles in all: *The Twilight Zone* (1993), *Indiana Jones: The Pinball Adventure* (1993), *Judge Dredd* (1993), *Star Trek: The Next Generation* (1993), *Popeye Saves the Earth* (1994), *Demolition Man* (1994), and *Red & Ted's Road Show* (1994). Williams marketed these seven games as "Superpin" machines, so the wider cabinet style is sometimes called the Superpin cabinet, but everyone in the virtual pinball world calls it the "wide-body" cabinet. (Actually, everyone calls it the "widebody", unhyphenated, but my spell-checker disapproves, so I'm using the hyphen here.)

The wide-body design is identical to the standard WPC cabinet in every particular, except for the width of the main cabinet, which is 2¾ inches wider than the standard body. All of the other dimensions are exactly the same as in the standard body. The backbox size is identical in both versions. (A lot of people assume that wide-body WPC machines had extra-wide backboxes as well, but they didn't. They used the same backbox dimensions as the regular WPC machines. So there's really no such thing as a "wide-body backbox", at least as far as the original WPC machines go.)

The wide-body machines obviously require wider variations for some of the hardware trim parts, to match the wider cabinet dimensions. If you build your own wide-body design, you'll need to get the wide-body versions of the affected parts. Be aware that the wide-body trim parts can be a little harder to find and a bit more expensive than the standard-body parts. Fortunately, you don't actually need very many wide-body-specific trim parts for a wide-body cabinet, thanks to the way they kept everything except the width the same as the normal cabinets. The only special wide-body trim parts required are the "lockdown bar" (the metal trim piece across the top front of the machine, also known as the front molding) and the glass cover.

The lockdown bar is mated with a second piece known as the "lockdown bar receiver", which *doesn't* require a separate wide-body version. The standard receiver works with both standard and wide-body lockbars.

Custom width

In addition to the WPC standard-body and wide-body designs, there's a third option for custom builders: you can design a cabinet with a completely custom width that doesn't match either of the official Williams designs.

The main reason to build to a custom width is to get an exact fit for your playfield TV. It's not possible to order a TV in a bespoke size; we can only buy what the TV manufacturers offer on the mass market. We can, however, build our cabinets to whatever sizes we want. So some people build their cabinets to fit their top TV pick, rather than settling for TV that only approximately fits one of the standard cabinet sizes.

Customizing the width has implications for the associated cabinet hardware, obviously, since off-the-shelf parts are only available in the standard sizes. To minimize these implications, you can use the same principle that Williams did when they designed the wide-body variation: as long as you keep all of the other dimensions the same, you can change the cabinet width to any size desired, and the only custom hardware required will be the lockdown bar and the cover glass.

The lockdown bar is the main challenge. Fortunately, there is a source for custom-width lockdown bars: VirtuaPin. They're the only source of these I know of, so hopefully they'll keep selling them. Expect to pay about double the price of the standard-size lockdown bar (which I think is a pretty reasonable premium, considering that it has to be custom-made on demand).

Note that the lockdown bar is mated with a second part known as the "lockdown bar receiver". The receiver does *not* need to be customized for different widths; the standard receiver will work with any lockbar width, as long as your cabinet is wide enough (minimum 19½ inches inside width) to accommodate it. (This is no problem if you're making a wider-than-standard cabinet, but keep the minimum size in mind if you're designing a miniature cabinet that's narrower than usual.)

The glass cover is easy to order in a custom size, and won't cost any more than a standard size. Don't bother looking at online pinball vendors; simply order from a local window glass supplier. Any glass shop should be able to fabricate a custom glass sheet for you in almost any desired size. Once you know the inside width of your cabinet, order a tempered glass sheet in the required width, by 43" length, by 3/16" thickness.

When calculating the required width of your glass, take into account the overhang beyond the inside dimensions. The easy rule of thumb is to make the glass **½" wider** than the **inside width** of your cabinet (that is, the distance between the insides of the side walls). For example, the standard body cabinet has an inside wall-to-wall width of 20½ inches, so the standard playfield glass is 21" wide.

One more tip about ordering custom glass: ask the vendor to **omit** any marking or etching that identifies the glass as tempered. Glass shops will often include a special marking on tempered glass to certify building-code compliance, in case you're planning to use it for something like a shower enclosure where tempered glass is legally required. But a marking like that can be an eyesore in a virtual cab; it might end up in a corner where it's in plain view. Tempered glass is good for a virtual cab for safety reasons, but there's no legal requirement for it, and thus no need for certification marks.

How to choose a cabinet width

If it weren't for the constraint of fitting a TV, I'd just tell you to use the standard-body plan and leave it at that. Using the standard dimensions produces a machine with exactly the right proportions to look like authentic, and lets you use readily available off-the-shelf parts for all of the hardware. It's the easiest approach and yields great-looking results.

But sadly, TV manufacturers don't always cooperate with our virtual pinball plans. TV manufacturers only make TVs in certain sizes, so we're stuck with whatever sizes are on offer. The TV you pick based on price and performance might not be available in exactly the right size for a standard cabinet.

What size TVs will the standard cabinet sizes accommodate? There's no absolute rule here, since the nominal diagonal size of a TV doesn't tell you the exact exterior dimensions - the only way to be sure is to measure the TV. (You might also be able to find the dimensions listed in the specs on the manufacturer's Web site or on a retailer site.) Generally speaking, a standard-body cab will accommodate most 39" TVs, and can handle some 40" models, and a wide-body can handle most 45" TVs.

My personal preference is to try to stick to the standard-body width if at all possible, meaning that I'd try to find a 39" TV that fits the standard cab's 20½" inside width. I prefer the proportions of the standard cabinet as a matter of aesthetics. If the TV you want to use is only available in a larger size that won't fit the standard-body design, I'd switch to the WPC wide-body, since it works with entirely off-the-shelf cabinet hardware. I'd accept the trade-off of some "dead" space in the cabinet if the TV didn't quite fill the full wide-body width, to keep the cabinet hardware standard.

I'm personally a little leery of building a cab to a custom size, not so much because of the extra cost of the custom parts (which really isn't all that huge a difference), but because it could preclude ever replacing or upgrading the TV. The odds are against any future model having exactly the same dimensions. At the very least, you won't be able to get the glove-like fit that made you choose the custom size in the first place.

Custom length

As long as we're on the subject of custom widths, we should consider lengths as well.

As with a custom width, the main reason to build to a custom length is make a TV fit exactly. As discussed in Chapter 7, Selecting a Playfield TV, a real pinball playfield is much more elongated than a 16:9 TV screen; a typical 1990s era playfield is more like 20:9. So placing a 16:9 TV in a standard-sized cabinet leaves a few inches of dead space at the front and/or rear of the cabinet. Some cab builders don't like that idea because they want to fill every square inch with TV display area.

One way to deal with the extra space is to remove it by shortening the cabinet length to exactly fit the TV.

In my opinion, it's better to stick with the standard cabinet length and accept that there will be some extra front-to-back space. The main problem with a custom length is that you won't be able to find side rails or glass guides; no one sells those in custom sizes as far as I know, and they'd be difficult to fabricate yourself unless you have some good metal-working tools. Besides, the extra space can be put to good use for features that you might want anyway:

- If you're going to install a plunger, you might need 3-4" of extra space between the front of the TV and the front of the cab to make room for the plunger mechanism.
- Even if you don't need the space for a plunger, I still like setting the TV back a few inches from the front for the sake of sight-lines, so that you don't have to look straight down to see the flippers.
- Space at the front of the TV can be used for an "apron" similar to that on a real machine, with printed instruction cards, or even small monitors that display live instruction cards, scores, etc.
- Space at the back of the TV can be used for a flasher panel or LED matrix.

Custom backbox sizes

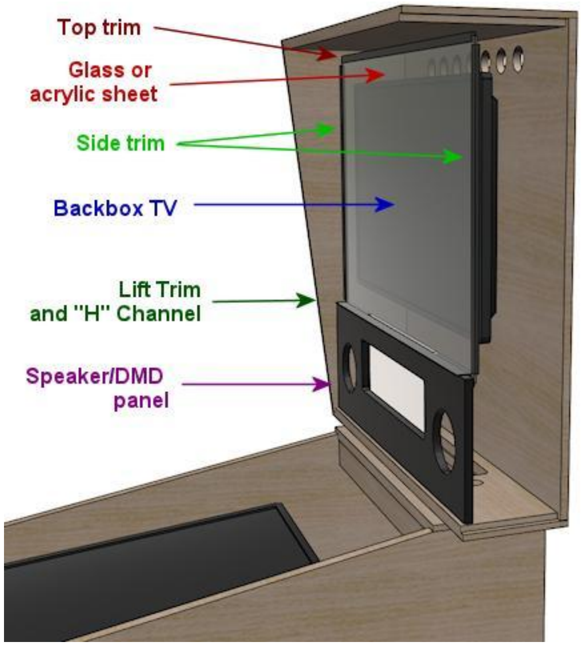
As with the main cabinet dimensions, some virtual cab builders choose to deviate from the standard backbox sizing to better fit a selected TV.

Finding a backbox TV that fits the available space can be even more vexing than finding a playfield TV. There are two big problems here. The first is proportions: modern TVs all use the 16:9 aspect ratio, but the translites in the WPC design are much squarer: approximately 13:9, closer to the old-fashioned NTSC 4:3 ratio. The second problem is that there are very few models available that are even close to the right size. 32" is an extremely popular size, but sadly, a 32" is much too wide for a standard backbox. The next size down tends to be 27" or 28". A 28" TV will leave about an inch of dead space on each side, and a few inches above and below. A 29" would be close to perfect, but at the moment those just don't exist.

You can't control what sizes the TV manufacturers make available, so you can either live with a little dead space around the perimeter, or you can resize the backbox to fit the TV. I personally think the dead space is the better compromise, in part because it lets you use standard off-the-shelf pinball parts, but mostly because the backbox shape will look "wrong" if you change it from the standard. This is a case where I think a lot of cabinet builders tend to fixate on the wrong thing during the planning stages, by focusing on the dead space rather than the overall proportions. The thing that you might not appreciate during the planning stage is that any dead space tends to disappear into the background when you're actually playing. Your eye sees what's there (the backglass graphics), not what's missing (the dead space around the edges). The proportions of the overall outline, on the other hand, are always noticeable.

If you do want to consider custom dimensions for the backbox, keep in mind that some of the associated hardware parts are sized for the standard dimensions, so you might not be able to use off-the-shelf parts for everything. Here are the parts that will be affected (see the illustration below if you're not familiar with all of these):

- The glass or plexiglass cover for the TV. This isn't a required part, but I recommend including it because it creates a more authentic appearance. There's no downside to a custom size for this part, because it doesn't come in an off-the-shelf version to begin with; you'll have to custom-order it even if you're using the standard size. You can have this made at any local window glass shop or plastics store, and they'll be able to cut it to whatever size you need.
- Trim pieces for the glass/plexi cover. Real pinball machines use black plastic trim around the edges of the translite. These are only available in the standard sizes. You can fairly easily cut them to smaller lengths if necessary, but there's no good way to make them longer, so you'll have to live with some gaps if you use a wider or taller than normal size.
- Speaker panel "H" channel and translite lift trim. These are plastic trim pieces that go at the top of the speaker panel and bottom of the translite, respectively, and they're sized to the standard width. If you use a wider-than-normal width, you can use the standard pieces, but there will be some gaps at the edges.
- Speaker panel. If you use a non-standard width, you'll need a custom speaker panel, assuming you're using the three-monitor configuration. No one sells those in custom sizes, so you'll have to fabricate one yourself. The ready-made speaker panels are made from plywood or particle board, so building your own only requires woodworking tools. Be aware that it's a fairly advanced project requiring precision work. You'll have to cut two large circular holes the speakers and a large rectangular opening for DMD. I'd recommend using a CNC machine (a computer-controlled cutting machine that cuts according to a digital plan). There are online services for this, such as SendCutSend. If you live in or near a major city, you might also be able to find a local CNC service, or a "maker" facility that lets you use their equipment for an hourly fee.



Mini cabs

A popular variation on the basic cab design is to scale things down a bit from the real machines. This can be especially attractive if you don't have a lot of space in your house, since a full-sized cab is a rather large and imposing piece of furniture. Downsizing a bit might also help gain acceptance from spouses and other family members who aren't as enamored with pinball as you are.

There's no "standard" mini-cab design, but you can find ideas from other people's builds by searching the cab forums at sites like vpforums. Many people who've built their own cabs post build logs with details of their design.

If you want to design a mini-cab from scratch, you can start with the basic WPC design, and just scale down all of the dimensions based on the playfield monitor you choose. A 32" TV makes a good core to build a mini-cab around; if you scale everything down proportionally, it yields a cab that's about 3/4 of the full size. That's enough of a reduction to fit more comfortably into a residential setting, but it's still big enough to be free-standing.

A few people on the forum have shrunk things down even further, to table-top or hand-held size, using a small computer monitor or tablet as the playfield.

For a mini-cab in the 3/4 scale range, you should be able to build it pretty much the same way that you'd build a full-size cabinet. You'll have to make the same adjustments to cabinet hardware discussed above under "custom width" and "custom length", but otherwise you should be able to use standard materials (such as ¾" plywood for the enclosure) and many of the standard hardware parts. One thing to keep in mind is that interior space will be a bit tight for the electronics, but you should be able to fit the necessary computer parts and a basic set of feedback devices.

If you reduce the scale to table-top or hand-held dimensions, you'll have to invent a lot more of the design on your own, since most of the standard hardware will be too large. That's beyond the scope of this guide, but you should be able to find one or two examples in the forums or elsewhere on the Web if you're looking for inspiration. Note also that all of the pinball software discussed in this guide is for Windows PCs, so if you're considering something else (like a tablet or Raspberry Pi) as the computer core, you'll also have to find other software to use. There are some decent commercial pinball games for tablets that could serve, but the commercial games don't tend to have any integration with cabinet features, so it might be challenging to make everything work the way you want it to.

WPC cabinet plans

We now present our WPC standard-body cabinet plans. These are based primarily on measurements taken from actual WPC pinball machines, with some additions and modifications to accommodate the peculiarities of virtual pinball. I've tried to identify all of the deviations from the real machines, for those with a special interest in accurately re-creating the originals.

Other Internet plans

There are several other pinball cabinet plans available on the Web, including other replica WPC designs. Some of the other WPC plans I've seen have slight variations from mine, so you might want to compare and contrast any others you find as a sanity check, and to see if there's anything you prefer in the variations. I've taken a great deal of care to check my plans against actual WPC machines, and I believe the version presented here is the closest to the real thing that I've seen, but of course that doesn't mean they're the ideal plans for every build, just that they're close to what Williams actually did build. You might have good reasons to deviate from that. Most of the details can be changed in small ways without much affecting the usability of the finished machine. (One usability detail that think you should avoid tinkering with unless absolutely necessary is the placement of the flipper buttons: that's very uniform on the real machines, so players are likely to notice a difference in how it feels if you change this more than very slightly.)

One set of alternative plans that I'll call out in particular is Jonas Kello's Sketchup model, available on github:

github.com/jonaskello/wpc-cabinet

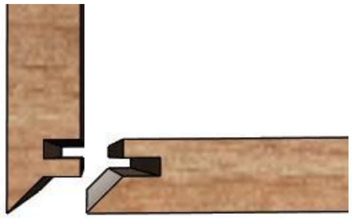
The nice thing about his 3D model is that you can look at it from different angles, which might be helpful whenever my illustrations leave something unclear about spatial relations between components. Jonas's model appears to have been prepared with excellent attention to detail. One warning, however: he explains that he took physical measurements from a wide-body WPC machine (*Star Trek: The Next Generation*) and adjusted them to the standard-body dimensions. This creates an opportunity for errors and inconsistencies to creep in, and in fact I came across a couple of errors in his model that are clearly due to this. My measurements were taken from actual standard-body machines (*Theatre of Magic* and *Medieval Madness*), so while I'm sure I have some errors and mis-readings of my own, my readings at least are free of that particular source of error. I also noticed some very slight variations (on the order of 1/16" to 1/8") between some of my measurements and Jonas's, which I'm sure can be attributed to some combination of our respective judgment calls squinting at the ruler, and variations in the original manufacturing. Williams historically used multiple subcontractors to produce their cabinets and playfields, so I imagine there had to be some variations from unit to unit for any given game, let alone different titles manufactured years apart.

Joinery

In wood-working, joinery is the art of forming joints where pieces of wood meet. There's a lot more to this than just nailing boards together; joins can involve angled edges to hide seams, and interlocking tabs and slots to add strength. Joinery is a huge subject that goes well beyond my expertise, so I won't try to offer a primer here. However, I do want to provide a quick overview of the way that the corner joints work in the real pinball machines, because you might want to adapt these - either to something simpler, if you don't have the right tools for the typical pin cab joins, or to something better, if you have other styles you prefer.

Apart from the corner joins, most of the joins we use in the plans are straightforward enough that you probably won't need to change them. Most of the joins (save the corners) are simple dado or rabbet joins that you can execute with straight router bits or a table saw.

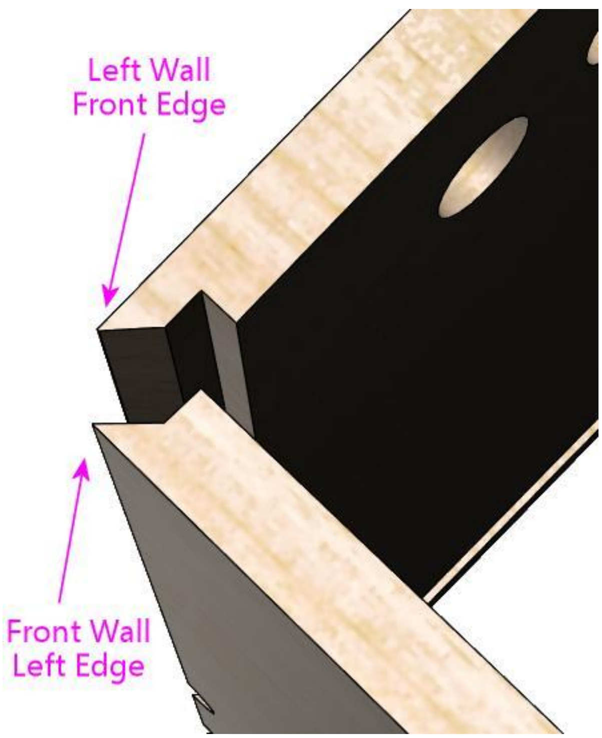
Now, on to the main cabinet corner joins, where the walls meet at right angles. There are several ways to execute these, and you'll see a few different approaches in the real machines, depending on manufacturer and era. The most sophisticated join I've seen in pin cabs is the one used in the WPC cabinets of the 1990s: a lock miter join, which looks like this (viewing the front left corner from above):



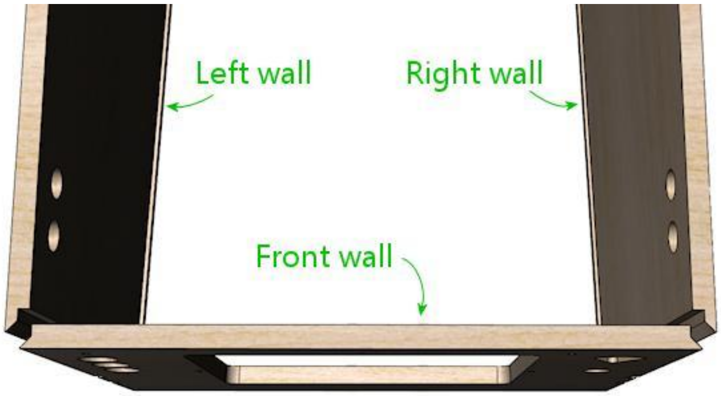
Lock miter join, at the front left corner of the main cabinet, viewed from above. This is the type of join that Williams used for the original WPC cabinets from the 1990s.

This is a really nice join aesthetically, since it makes the seam invisible by placing it exactly at the corner. It's also extremely strong, thanks the large glue surface area from the interlocking tabs. But it's a highly challenging bit of carpentry - even a lot of experienced woodworkers consider it difficult. You need a special router bit to cut it, and it's notoriously hard to get the positioning of the cuts just right. What's worse, even with the right tools and skills, you still might not be able to get it to work with some plywood, since the plies might not be strong enough for the thin cuts. The advice I've seen on most carpentry forums is that this join is great for solid wood, but not always suitable for plywood. The owner of Virtuapin, which sells reproductions of the Williams cabinets, has said that Williams was only able to make the lock miter work on their cabinets because the plywood of that era was better than today's. My own experience has been a bit better than that - I've made it work nicely with Home Depot plywood - but I can attest to how challenging it is to get the router bit set up properly.

An easier alternative is a mitered rabbet, which combines a miter joint (a 45° diagonal cut at the corner) and a rabbet (a squared recess along one edge):



Mitered rabbet join, at the corner between the front wall and left wall of the cabinet.



Top view of the front section, with a mitered rabbet at each corner.

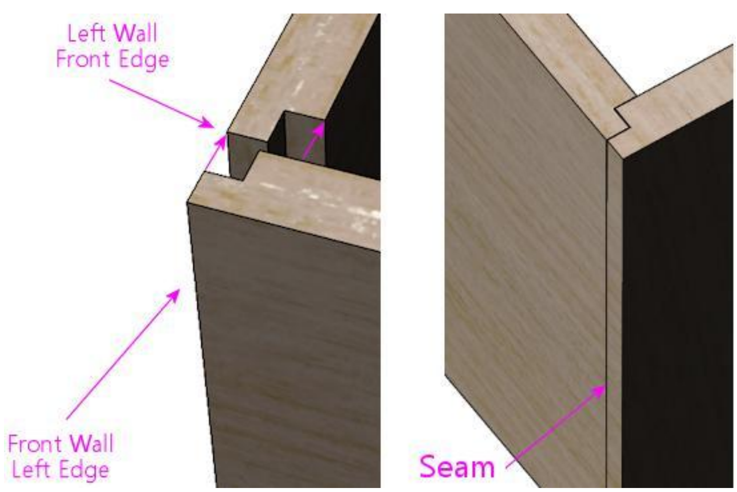
The mitered rabbet has the same aesthetic advantage of the lock miter, placing the seam exactly at the corner, but it's easier to execute, it's more plywood-friendly than the lock miter because it doesn't have such small structures.

To cut a mitered rabbet, you can buy a special router bit set just for this join, or you can cut it in multiple passes with a couple of basic bits: a 45-degree chamfer bit and a straight bit. The procedure is beyond what I can include here, but you should be able to find tutorials and instruction videos about it on the Web easily, since it's a popular joint for general woodworking.

If the degree of difficulty were no obstacle, I'd definitely go with one of the joins illustrated above (the lock miter or mitered rabbet). They both have the virtue of being cosmetically seamless, thanks to the 45° diagonal cut to the outside corner. Both are strong joins that fit precisely and self-align; assembling pieces made with these joins is like snapping together puzzle pieces.

Despite the virtues of the mitered joins, they might not be your top choice because of how challenging they are to cut. If you want something simpler, the alternative I'd use is a double rabbet.

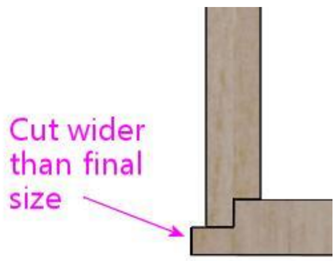
The double rabbet join dispenses with the diagonal cut out to the corner, and instead uses square interlocking notches. The big advantage is that it can be implemented with common tools (a table saw or a straight router bit). But it has a couple of drawbacks. For one, it leaves a visible seam along one of the joined faces. For another, it makes it a little trickier to translate the cabinet measurements to the wood pieces, because of the way one piece slightly extends the apparent length of the adjoining piece.



Double rabbet join, similar to the join used in Williams System 11 cabinets. This is a simpler alternative to the lock miter join used in the WPC cabinets, but it has the *drawback that it leaves a seam along one face near the corner.*

Williams used something similar to the double rabbet join in their System 11 machines in the 1980s, and VirtuaPin uses it in their reproduction cabinets and flat packs, so it certainly qualifies as a professional-grade option. And it's a good join functionally - it's strong and makes a reasonably precise fit. But a rabbet join leaves that visible seam, so I consider the mitered options to be superior, if you have the tools (and patience) for them.

If you do decide to use the double rabbet join, there are a couple of things you can do to minimize the visibility of the seam. First, choose the placement of the seam so that it's on the less visible face. The seam only affects one or the other adjoining face at each corner, so you have a choice of which wall will have the seam. The Williams System 11 cabinets placed the seams on the sides (rather than the front face), which seems like the better choice aesthetically, since the front is more visible. Second, cut the front piece a tiny bit wider (1/16", perhaps) than the final size, so that it leaves a little overhang when initially assembled, as illustrated below.



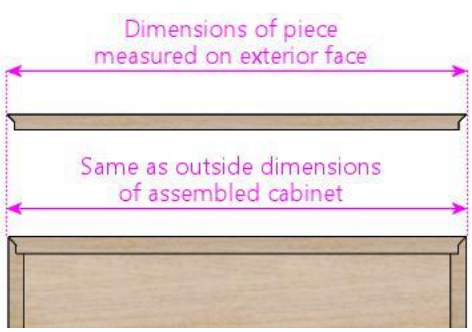
After assembly, the overhang lets you sand down the excess material until it's exactly flush with the adjoining section. It's almost impossible to get the surfaces perfectly flush in the initial cut, so your best bet is to start with a slight overhang that you can sand until flush. You can then add wood filler at the seam to further smooth it out.

Adjusting dimensions for joinery

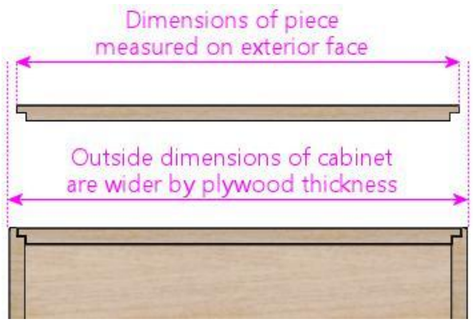
Pay close attention to the effects of your chosen corner joins on the overall dimensions.

The dimensions shown in our plans assume that you're using the joinery specified. In the case of the corner joins between the side walls and the front and back walls, this means we assume you're using one of the mitered joins we described above, either the lock miter or mitered rabbet. (Our illustrations use the mitered rabbet for the main cabinet corners, so you'll see that join in the close-ups. The lock miter is equivalent in terms of all of the measurements, so there's no need to make any adjustments if you use the lock miter instead.)

With the mitered joins, note how each piece's dimensions from corner to corner exactly match the **assembled** cabinet's outside dimensions for that section:



In contrast, with the rabbet join, note how the "inside" piece (the one forming the face with the seam) is slightly shorter than the assembled cabinet's outside dimensions. This will be shorter at each corner by the depth of the rabbet groove, which is typically half the plywood thickness, so assuming there's a join like this at each end, the overall piece will need to be cut shorter than the desired final outside dimensions by 2 × ½ × the plywood thickness = 1 × the plywood thickness:

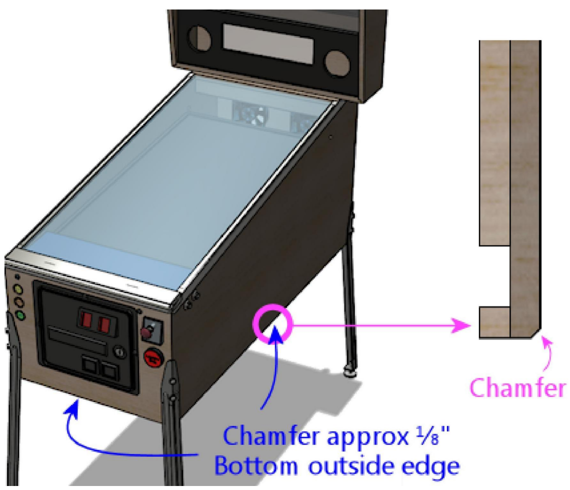


Our measurements for the main cabinet are based on using mitered joins at the visible corners, so be sure to adjust the dimensions before cutting if you're using a different join.

Edge finishes

On the original WPC cabinets, the outside bottom edges of the side and front walls are finished with a chamfer (a 45° bevel), about ⅛" wide. The point is simply to make the plywood edge less sharp and splintery. A softer edge is less likely to snag on clothing or give you a splinter. It also protects the plywood from chipping.

The chamfer is just one way to soften the edge. Alternatively, you can go over the edge with a roundover router bit to give it a curved edge, or smooth it out with a power sander. You can also just leave it square if the sharp edge doesn't bother you.



The WPC machines had square corners for the front vertical edges (at the corners between the front wall and the left and right walls). Some of the newer Stern machines round those out slightly. If you want rounded corners, you can go over the corners with a roundover router bit after assembly. I'd at least use a power sander to smooth the corners a little, but I wouldn't try to achieve visible curvature that way, since it's hard to make it uniform. Use a router if you want a pronounced curve.

Exploded view

This view shows all of the pieces making up the main cabinet body.

The triangular wood pieces at the corners are for the leg fasteners. Metal fastener plates fit over these on the inside, and two bolts go through each one at a 45° angle to the adjacent walls.

The two pieces at the top rear form a "shelf" that the backbox rests on. The rectangular routed opening in the horizontal piece is to pass power and video cables between the cabinet and backbox. The opening shown is what's used on the real machines, and it works well for a virtual cab as long as you only need to pass cables through. You might need a larger opening, though, if you plan to use a large monitor in your backbox that needs to extend into the main cabinet. This isn't an issue for a typical three-monitor setup with a laptop display for the DMD (or a real DMD device).

The smallish slat near the bottom front attaches to the floor on the real machines to form a niche to hold the cashbox. (The cashbox sits under the coin slots to collect the inserted coins.) Most virtual builds omit this piece to leave more room for the PC motherboard, which most people situate on the floor of the cab about halfway back.

A plywood cutting plan

Here's my plan for cutting up the plywood sheets. I won't claim that this is the best possible way to fit all of the pieces together, but it's a pretty efficient fit, and it doesn't require too many individual cuts.

This plan assumes that you're making the cabinet floor out of plywood (rather than particle board), so it's designed for two 4x8 sheets. If you make the floor out of particle board, though, it's just barely possible to fit everything else into a single 4x8 sheet; see Single-sheet layout below.

Before using this plan literally, please be sure to make any adjustments necessary for differences in your cabinet layout. In particular:

- Please read the notes above, under Joinery, about adjusting the dimensions of the cabinet pieces if you're planning to use rabbet joins or butt joins. The dimensions in the diagram assume miter joins for the cabinet corners, where all of the outer faces extend all the way to the corners. Don't make the same adjustment for the backbox; my backbox plan only uses rabbet join to start with.
- The dimensions shown are for a standard-body WPC cabinet. If you're building a wide-body cabinet or a custom-width cabinet, you'll need to adjust the width

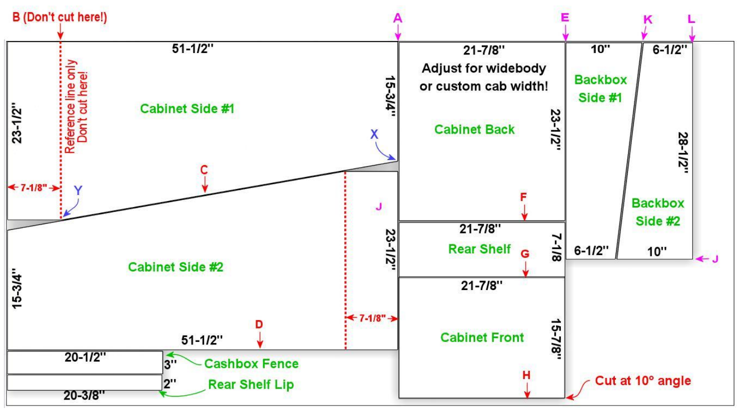
of the front panel, back panel, and rear shelf. The plan shown has room to widen those by about 5 inches, which is wide enough for a WPC wide-body (add 2¾" to the standard body width).

Each piece also requires routing for joinery and drilling for buttons and other attachments. The details are explained piece-by-piece in the sections below. Instructions for assembling the pieces into the final cabinet follow all of that. (The assembly process itself is almost easy once you have everything cut and routed, but it takes a little work to get there.)

Don't start by marking all of those cut lines on the plywood sheet. That won't produce accurate results, because your saw blade will remove some of the material between the pieces at each cut, which you can't easily account for in the initial marking. What works better is to make one cut at a time, and then measure the next cut from the newly cut edge. That ensures that the saw blade width is automatically accounted for at every stage, without any tricky calculations or estimates. The step-by-step instructions below use this approach.

Also, to make sure that your saw blade's width is accounted for in each cut, always align the saw on the "outside" of each cut, so that the blade's width is positioned in the "leftover" portion of the plywood rather than within the work piece you're cutting. Align the edge of the blade on each cut with the outside edge of the work piece.

Step-by-step procedure:

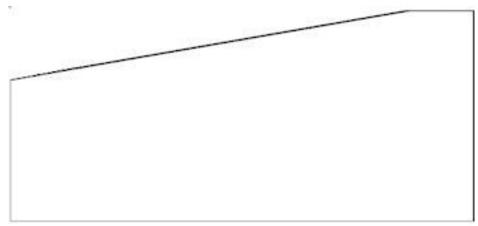


Plywood sheet #1 - 4' x 8' x 3/4" nominal thickness. Dimensions are all for the WPC standard-body design, so be sure to make appropriate adjustments if you're building a wide-body cab, a custom-width cab, or you're using a non-standard length. Click image for full-sized version.

1. Measure from one edge of the sheet to 51-1/2", to the cut line marked "A" in the diagram above. Cut across the whole sheet on this line. Set aside the smaller portion.

2. We're now going to cut the diagonal line "C" for the side panels. On the "A" cut line, measure 15-3/4" to the point labeled "X", and mark that spot. On the opposite side, pencil in the line marked "B", 7-1/8" from the edge. ("B" is just a reference point - don't cut along this line.) Then measure 23-1/2" along "B" from the edge to the point labeled "Y", and mark that spot. Draw the diagonal line "C" through points "X "and "Y". Cut all the way along "C".

3. The piece you just cut out is one side wall of the cabinet. You just need to cut out that 7-1/8"-long flat portion where the diagonal edge flattens out. Mark a line 23-1/2" from the opposite side, parallel with the opposite side, and cut along that line to make the flat portion. The finished side piece should look like this:



4. On the remaining piece, you can repeat the measurements from step 2 to mark the 51-1/2" line "D", or you can simply use the finished first side piece as a template, since both sides should be identical. Either way, mark line "D" and cut along the line to cut out the second side piece.

5. Repeat step 3 with the second side piece to cut the 7-1/8" flat portion.

6. From the remains of the 51-1/2"-wide section, cut the two small rectangular pieces shown at the bottom: the cashbox fence (20-1/2" x 3"), and the rear shelf lip (20-3/8" x 2").

7. Now we'll work on the (almost) half-sheet that was left over from the first step, after cutting along line "A". This section should be about 48" x 44". Mark cut line "E" by measuring 21-7/8" from one edge.

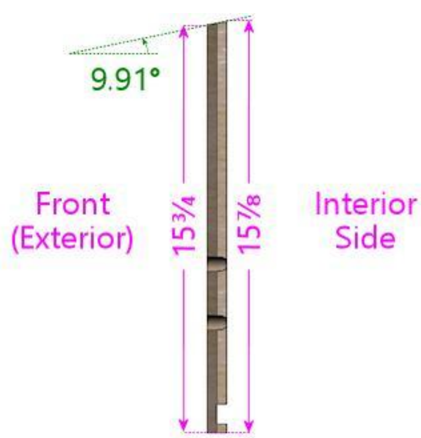
For wide-body or custom cabinets, adjust this dimension! 21-7/8" is for the WPC Standard Body design, which fits the standard Williams lockbar (Williams/Bally part number D-12615, A-18240). If you're building a wide-body that's matches the original Williams "Superpin" cabinet, to fit the off-the-shelf WPC wide-body lockdown bar (A-16055, A-17996), the width should be 24-5/8". If you're building to a custom size, you can determine the width to use here by adding 1-3/8" to your desired inside cabinet width (the distance between the inside faces of the side walls - essentially, the width available for the main TV).

Make sure the board is oriented so that "E" goes down the longer (48") dimension! Otherwise you won't have enough material for the three pieces shown.

8. Using the 21-7/8" section we just cut, mark line "F" 23-1/2" in from one edge. Cut along this line. This yields the back wall of the cabinet.

9. Continuing with the remainder of the 21-7/8" section, mark line "G" 7-1/8" from one edge. Cut along this line. This yields the rear shelf for the cabinet.

10. Still using the remainder of the 21-7/8" section, mark line "H" 15-7/8" from one edge. (This will use almost the entire rest of the piece, but there should be just enough left that you still have to trim this little bit.) **This line should be cut with the saw tilted at a 10° angle.** The saw should be tilted so that the face you cut through ends up as the **larger** face, taking the angled cut into account. The opposite face should only be 15-3/4" wide when done, due to the angled cut. If your saw only tilts in the opposite direction, so that the far face will end up being wider, you should mark line "H" at 15-3/4" to compensate. Here's the side profile of the finished piece, to give you a better idea of what we're going for:



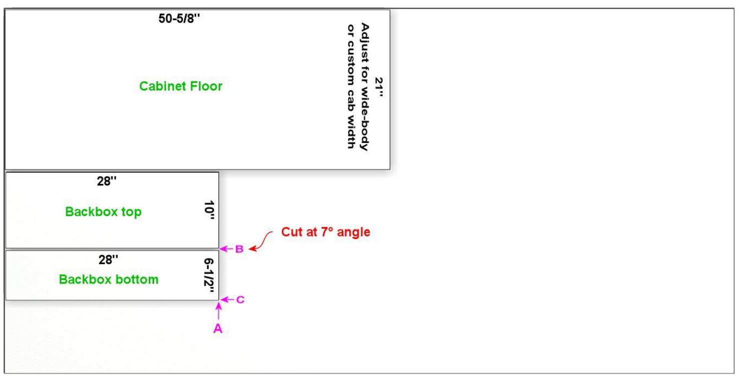
If your saw can't easily cut at an angle like this, you can get away with making a square cut. Cut the piece at the shorter 15-3/4" size in this case. The angle is to make the top edge of the piece align with the diagonal slant of the cabinet sides, so that all the edges are flush. But this isn't actually all that critical; it's only a slight aesthetic imperfection if you can't make the edges flush, and it's an imperfection you'll almost never have to look at anyway, since the whole area is covered by the lockbar! The only time you'll see it is when you take the lockbar off to get inside for service.

11. There are only two pieces left to cut out of the remainder of the first plywood sheet: the sides of the backbox. The leftover piece should be about 22" x 48" (or less than 22" if you're building a wide-body or wider-than-standard-body cabinet). Mark line "J" by measuring 28-1/2" from one edge. **Check that your saw is set back to 0° for a square cut.** Cut along line "J".

12. On the 28-1/2" piece we just cut, mark the diagonal line "K" by measuring a point 10" from a 28-1/2"-long edge at one end, and 6-1/2" from the same edge at the other end. Cut along this line. This yields the first backbox side piece.

13. On the other piece, you can either repeat the measurement to mark the square line "L", or you can use the first backbox side piece as a template to mark the cut line, since the two sides are identical. Cut along the line.

We're finished with the first plywood piece! Time to move on to the second sheet.



*Plywood sheet #2 - 4' x 8' x 3/4" nominal thickness. The dimensions are based on the WPC standard-body cabinet plan, so adjust the cabinet floor size if you're building a wide-body cabinet or using a custom width or length. Note that the backbox dimensions **don't** change for the "Superpin" widebody design - those used the same standard backbox size as regular pins. However, adjust as needed if you're building a custom backbox (e.g., building it around a particular monitor size). Click image for full-sized version.*

14. The first cut is the cabinet floor, which is a simple rectangular piece, 50-5/8" x 21".

The 21" dimension is based on the cabinet width, so adjust this for a wide-body or custom-width cabinet. If you're building to the Williams "Superpin" wide-body specs, make this 23-3/4". For a custom width, add 1/2" to the desired inside width of your cabinet (the space between the inside walls).

Note that the real pinball machines almost all used particle board for the cabinet floor instead of plywood, to reduce cost. The cabinet floor isn't visible to players, so they figured no one would care about cheap materials there. Even so, I think it's worth using plywood, since particle board tends to sag over time, and plywood is lighter.

15. The backbox top and bottom require another tilted-blade cut, so it's not quite as simple as cutting the rectangles shown. Start by cutting line "A", at 28" from one edge of the leftover from the cabinet floor.

16. Mark line "B", at 10" from a 28" edge of the 28"-wide piece we just cut. **Set your blade to a 7° tilt.**

The blade should be tilted so that the face you're cutting into will be the wider side. If your saw only tilts the other way, so that the opposite face will be the wider one after the cut, flip everything around and measure line "B" at 6-1/2" in instead of 10" in.

Once you have it set up, cut at line "B" with the 7° tilted blade.

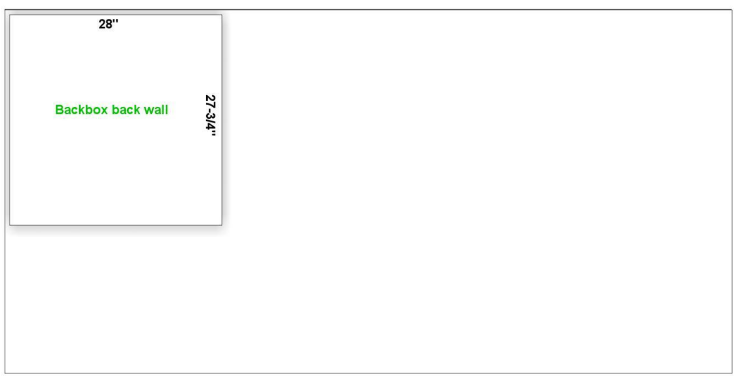
This yields the backbox top (or the backbox bottom, if you had to flip things around for the 6-1/2" cut).

Sanity check on the angled cut: The result should be 10" side on one face, and about 9-29/32" wide on the opposite face. If you did the 6-1/2" cut, the result should be 6-1/2" wide on one face. and slightly wider, about 6-19/32", on the opposite face.

17. **Set your saw back to 0° for square cuts.** Orient the remaining piece so that the angled cut with the narrow face is facing the saw blade. Measure line "C" at 6-1/2" from the angled edge (or 10" from the angled edge if you flipped around in the previous step). Cut along this line. Make the same sanity check as in the previous step for the angled cut.

18. You're now done with the 3/4" plywood! There is more one bonus step, but for this, you need a sheet of 1/2" plywood (actually, just a half sheet will do). This is for the back wall of the backbox. This is just a simple rectangle, 28" x 27-3/4".

If you prefer, you can use your remaining 3/4" plywood for this piece, but remember to adjust the depth of the groove that you route in the backbox side, top, and bottom walls accordingly. The back wall fits into a routed groove at the back of those pieces, so the groove has to match the thickness of the back wall. You'll also lose 1/4" of space inside the backbox, of course, so make sure you can spare it - some TVs are a tight fit.



Plywood sheet #3 - 4' x 8' x 1/2" nominal thickness. Or, a 4' x 4' half sheet is sufficient. Click image for full-sized version.

19. Additional bonus steps: there are a few more miscellaneous pieces that you'll need to cut out of the remaining plywood plus other wood stock. These are covered in detail in the sections below, but here's a quick summary:

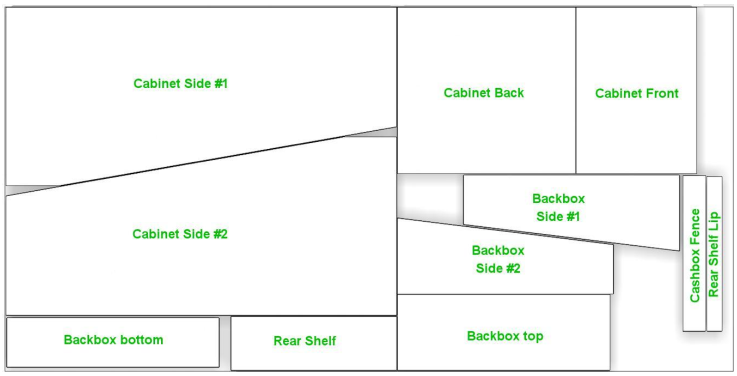
- Two 3" lengths of nominal 2x2 stock, cut in half diagonally down its length, to secure cashbox fence
- Four 6" lengths of nominal 2x2 stock, cut in half diagonally down its length, to reinforce the inside corners where the leg brackets attach
- Two 4-3/4" x 3/4" strips of 1/2" plywood, for DMD panel guides
- Two 15" x 3/4" strips of 1/2" plywood, for translite guides
- One 27-1/8" x 3/4" strips of 3/4" plywood, for a translite guide
- Two 12-3/8" x 1" strips of 3/4" plywood, for translite guides
- One 27-1/8" length of 3/4" reducer molding (or a similar shape fashioned from a nominal 1x2), for backbox trim at the top of the translite

And, of course, the woodworking on the plywood pieces isn't finished after you cut the last piece. Most of these parts require some additional work with a router, drill, and/or jigsaw. The sections below have all the details, part by part.

Single-sheet layout

It's just barely possible to make all of the 3/4" panels fit into a single 4x8 sheet, as long as you make the floor out of something else, such as particle board or MDF. The real machines of the 1990s did just that, because no one cared if the floor looked cheap. If you're on a tight budget, you can save some money by doing the same thing.

Here's a cutting plan that fits everything into one sheet. You'll have to be very careful to minimize wasted material between cuts to make this work - there's very little room for error or waste, since everything is packed so tightly together. Also, I don't think there's any way to make this work for a wide-body plan, since it just barely fits with the standard width pieces. For dimensions, see the Sheet #1 and Sheet #2 plans above.

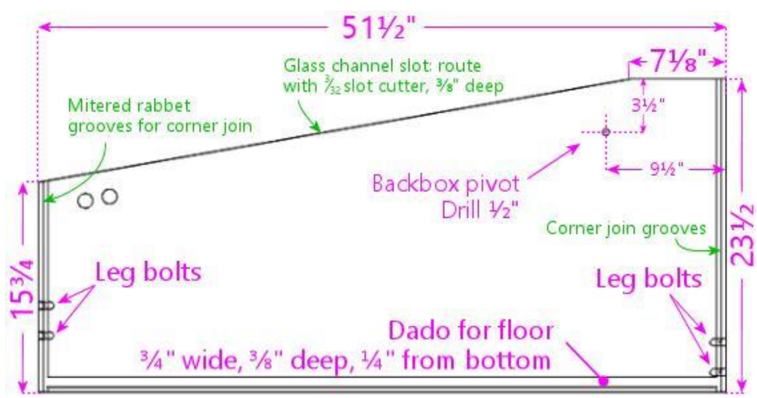


Alternative layout that fits everything into a single 4' x 8' sheet of nominal 3/4" plywood. This is based on the standard-body WPC cabinet dimensions; I don't think there's any room to expand the panels for a wide-body design. Click image for full-sized version.

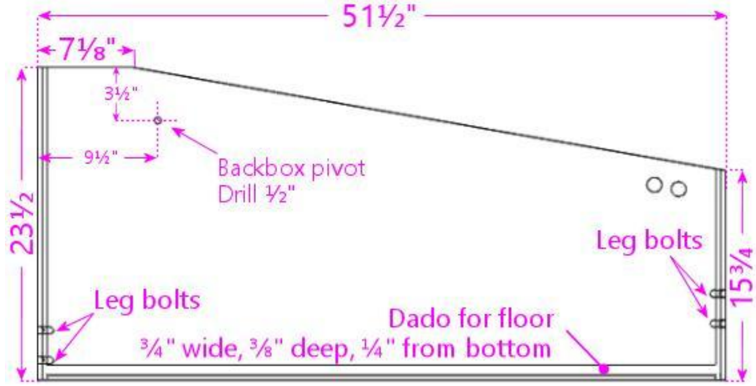
Side walls

Here are the side walls. The views are from the interior of the cabinet, to show details on the joinery routing.

(The flipper button holes and leg bolt holes are marked, but for the sake of readability, the dimensions aren't shown here. We'll provide close-up diagrams for these elements, with all of the measurement details, later in the section.)



Left side wall, viewed from the cabinet interior side



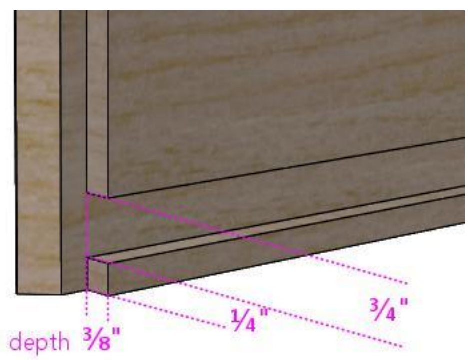
Right side wall, viewed from the cabinet interior. The right wall is a simple mirror image of the left wall.

Remember that we're measuring the dimensions of the pieces based on a mitered join (either a mitered rabbet or lock miter) at the front and rear corners, meaning that the piece's dimensions match the outside dimensions of the assembled cabinet. If you're using a different join at the corners, be sure to make any necessary adjustments. See Joinery above.

Some more views to help with visualization:

The backbox pivot is a ½" hole for attaching the WPC-style backbox hinge. If you're using different hardware to attach the backbox, omit this.

The "dado" at the bottom is a groove to join with the cabinet floor. Use a ¾" router bit to cut a straight groove ⅜" deep (that is, halfway through the thickness of the plywood). This runs parallel to the bottom edge, ¼" from the bottom edge. This is on the inside of the wall; the cabinet floor slots into this groove when assembled.



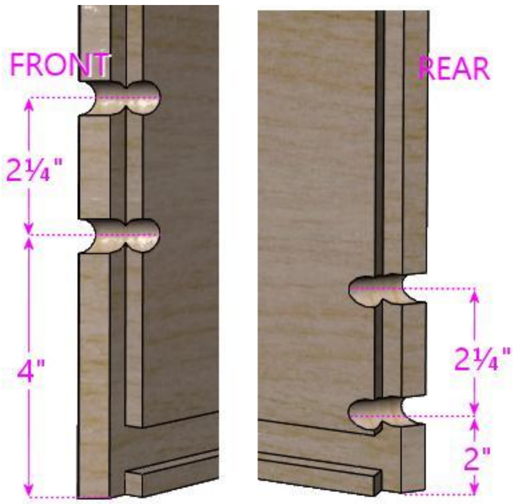
Left cabinet wall showing the dado (groove) for joining with the cabinet floor. Route the dado with a ¾" bit to ⅜" depth, ¼" from the bottom edge. This groove runs the whole length of the side wall. This is on the interior face, since it joins with the cabinet floor. The diagonal/step shape along the vertical edge at the left is the mitered rabbet cut for joining to the front wall, illustrated in more detail above.

Edge finishes

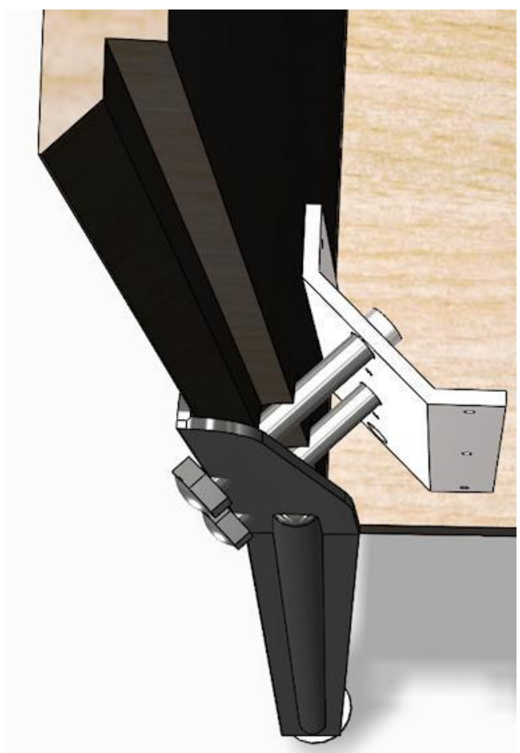
The original WPC cabinets use a slight chamfer (a 45° bevel) on the outside bottom edges of the side walls, to soften the edge and reduce splintering. This is optional, but it will reduce the chances of snagged clothes and cuts from bumping into the side. See Edge finishes above.

Leg bolts

The leg bolt holes are a little tricky. The bolts go through the corners at a 45° angle, so they bore through both adjoining walls at each corner. So, as shown in the illustration, the left and right walls only have "half a hole" for each bolt - really more of a semicircular notch.



Leg bolt holes, front (above left) and rear (above right). The distances are shown from the bottom of the cabinet. Note that the front legs are mounted higher on the wall than the rear legs. The legs themselves are the identical parts front and back, so the different mounting position is used to give the cabinet its characteristic tilt angle. The bolt shafts are ⅜" diameter.



Cutaway view (with the front wall removed) showing the leg bolts installed, to better illustrate how the bolt holes intersect the side wall. The triangular wood piece that normally fills the gap between the metal plate and the inside wall is also hidden.

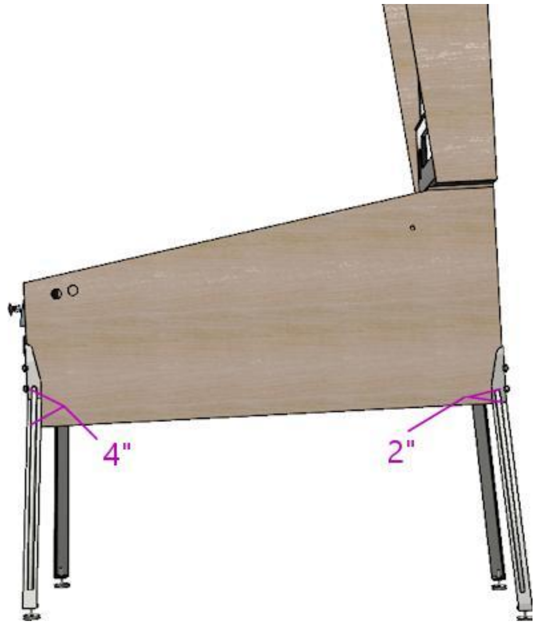


Illustration of how the leg install positions affect the cabinet slope. The legs are mounted higher on the cab in the front, which effectively raises up the back end slightly to slope the machine down toward the front. Standard pinball legs come with adjustable foot pads that you can use to make sure all four legs touch down and to fine-tune the playfield slope. The slope isn't needed for "physics" reasons on a virtual machine, but it's still desirable for an authentic appearance, and it also improves the viewing angle for the main TV.

There are two approaches to drilling the holes:

- **Before** assembly, using a router to cut the "half holes" in each panel. The bolts are 3/8" diameter, so each "half hole" needs to be 3/16" deep and 3/8" wide. You can accomplish this using a 3/8" round-nosed router bit - this type of bit has a half-dome shape for its tip, so it routes half-cylinder grooves, exactly as we need. Route to a depth of 3/16".

  The 45° angle makes it extremely difficult to cut these with a hand router, so this is only feasible if you're using a table router or a CNC machine to make the panels.

- **After** assembly, using a regular drill with a 25/64" bit to drill the holes. The 45° angle makes this a difficult job for a hand drill, but it's doable using a special jig that guides the bit at the proper angle. You can find general-purpose angle-drilling jigs at hardware stores and Amazon. Or you can build one yourself. Some people have successfully built such jigs using a couple of 2x4 blocks bolted together to make a corner. If you want something a little easier, here's a 3D-printable model for a jig that's set up with the proper spacing and angle:

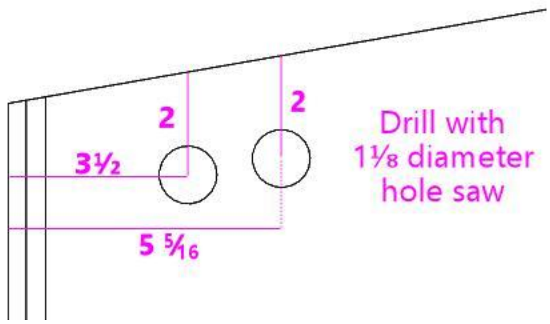
  leg-bolt-drilling-jig.zip

With either method, the holes should end up being a tight fit for the bolts. That's good, since you don't want your 250-pound cabinet standing on wobbly legs. But the holes might end up being so tight that the bolts won't fit at all at first. If that's the case, use a small round file to expand the holes slightly. Ream out a little bit at a time and test the fit frequently with a bolt, stopping as soon as it fits.

Flipper buttons

Here's the drilling plan for a set of two flipper button holes on each side. The front button in each set is the regular flipper button, and the rear button is the "MagnaSave" button, which is for the benefit of some pinball games that have extra controls beyond the regular flipper buttons. The rear buttons are optional, and not everyone likes them since they're not all that common on real machines, but I think it's good to include them because of the large number of virtual tables that make use of them. See Appendix 4, Tables with MagnaSave Buttons for more about these buttons, and a list of some of the tables that use them.

Note! Many older side rails (from before the WPC type) have pre-drilled holes for the flipper buttons. The WPC side rails don't need flipper button holes because they don't extend far enough down the side to cover the buttons, as most older rails do. If your rails cover the flipper button area, ignore our button drilling locations. Instead, use your side rails as drilling templates, since the holes in the cabinet will have to line up with the ones in the rails.



Flipper button drill hole detail for WPC-type side rails. Measurements are in inches; distances are to the center points of the holes. (Don't use these locations if you're using older side rails that cover the flipper buttons. Instead, use the pre-drilled flipper button holes in your rails as drilling templates, so that the cabinet holes line up with the holes in the rails.)

Drill the flipper button holes with a 1⅛-inch diameter hole saw or Forstner bit. The distances in the diagram are measured are from the center of the drill holes to the front and top edges of the wall, square with the front edge.

Variations:

- The rear (MagnaSave) buttons are optional. If you don't want to include them, simply don't drill the holes. The regular flipper buttons go at the same position whether or not you include the MagnaSave buttons.
- There are at least two other good ways to position the MagnaSave buttons. Some people place them directly below the flipper buttons, and some people prefer them diagonally behind and below the flipper buttons. Both of those patterns have precedents in real pinball machines that had the extra buttons (see Appendix 4, Tables with MagnaSave Buttons). The layout in my diagrams is based on the Williams MagnaSave games from the 1980s, so it's probably the most familiar look to most players, but not everyone likes the feel, due to the stretch to reach the rear buttons. The more vertical layouts are arguably easier to reach, and make it it easier to keep a finger on each button.
- Williams System 11 games (1980s) placed the flipper buttons about 1/4" higher than shown in my diagrams, which are based on the WPC games (1990s). System 11 games used broader side rails that covered the flipper buttons, so I think the slightly different positioning is purely to accommodate the different rails. I don't think it noticeably affects the feel.

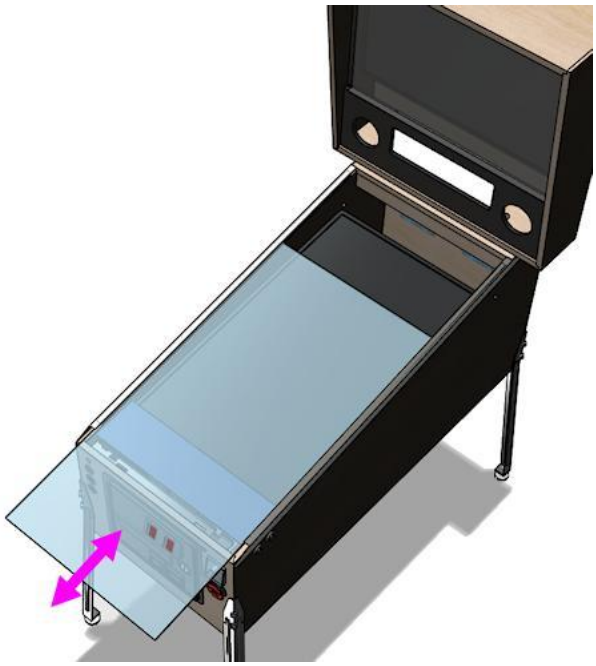
How to drill: My preference is to simply drill a hole straight through with a 1⅛-inch hole saw or Forstner bit. On the Williams machines, they drilled a more complex pattern where they drill partway through from each face with a 1⅛" bit (5/16" deep from the outside, 3/16" deep from the inside), and go the rest of the way with a ⅝" bit. This matches the shape of the flipper button, which has a large outer section and a narrower stem, and it makes for a more secure attachment. The reason I prefer a 1⅛" diameter hole all the way through is that it makes it easier to install LEDs behind the button to illuminate it from the inside.

The distances shown are from the outside front corner of the **finished cabinet**. Assuming you're using a mitered rabbet join, that's the same as the outside front edge of the work piece, so you can simply measure from the front edge of the work piece. Be sure to make any necessary adjustments to the measurements if you use a different join type, so that the position of the holes ends up at the indicated distance from the outside front corner of the cabinet when assembled.

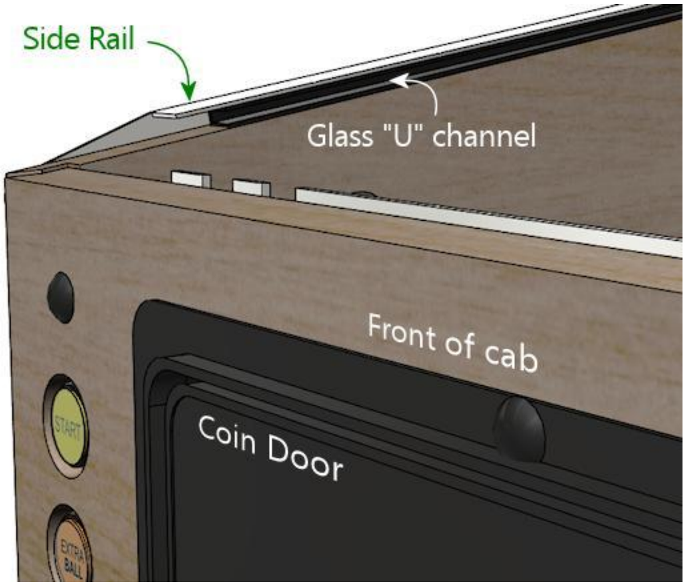
Glass channel slots

If you're going to install the standard side rails and a glass cover over the playfield, you should also install a set of "glass channels". These are plastic "U"-shaped trim pieces that fit under the side rails, along the left and right edges. These hold the glass at the sides.

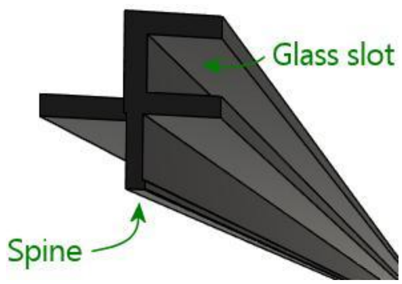
Because the glass channels are "U" slots along the length of the machine, you can slide the glass in and out of the channels through the front of the machine, after removing the lockbar. This is part of the tried-and-true design of the real machines that lets an operator easily open up the machine for maintenance access, and I think it's a great thing to replicate in a virtual cab.

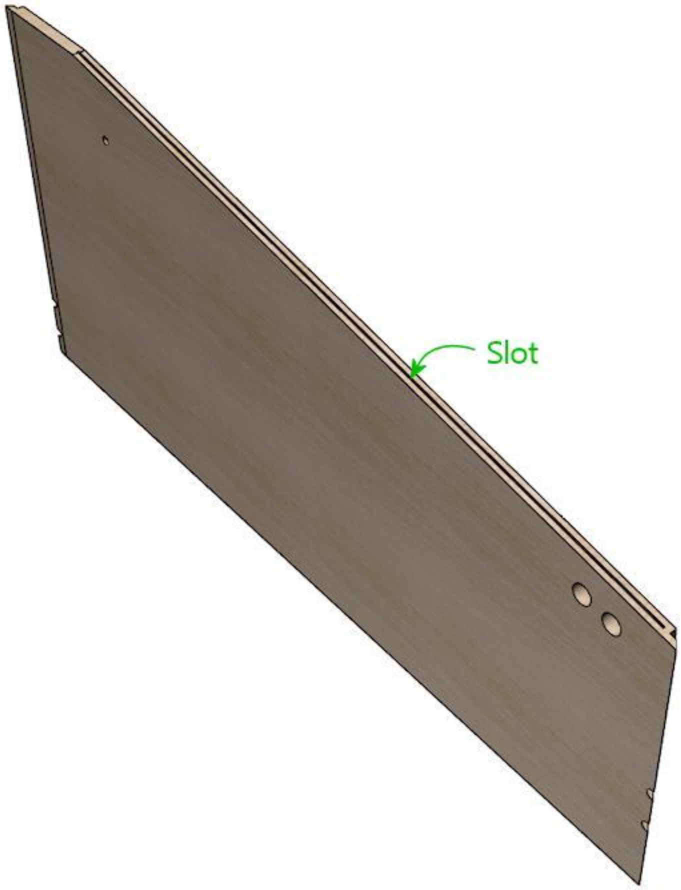


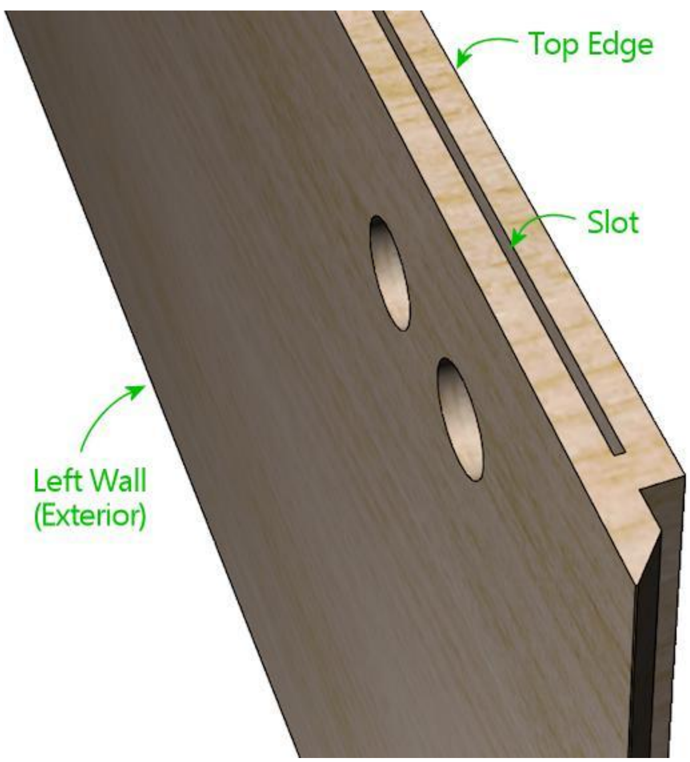
The glass channels are installed under the side rail. Here's a close-up of how they look when installed:

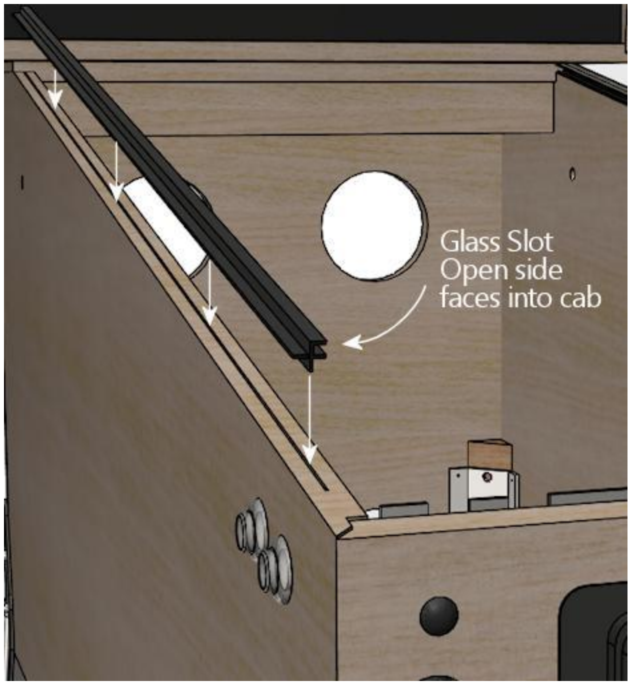


The channels attach to the side wall via a "spine" sticking out of the bottom of the plastic channel. The spine which fits into a slot in the top edge of the side wall.




Slot


Top Edge
Slot
Left Wall
(Exterior)

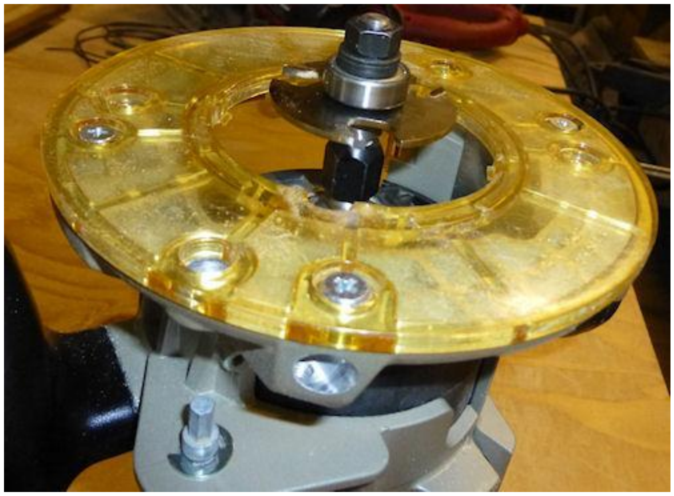


This is a neat design, in that you don't need any fasteners or adhesives. You just press the spine into the slot, and it's held there by friction. If it's ever necessary to take the channel out, you can just pull it out. On the other hand, it presents us with another little wood-working challenge: how do we cut that precise little slot?

As usual, it turns out that there's a special tool for this job, and it's really easy once you have that magic tool. What you need in this case is a special-purpose router bit called (naturally) a slot cutter. Just as a drill bit is designed to drill a hole of a specific diameter, each slot cutter bit is designed to make a slot of a specific width and depth. For this job, you need a bit with a 3/32" slot width and 3/8" slot depth. (A deeper slot, like 1/2" or 5/8", will also work if you can't find a bit for that exact depth. But the width is important - it should be exactly 3/32".) The bit I use for this is Freude part #63-106, which works perfectly.

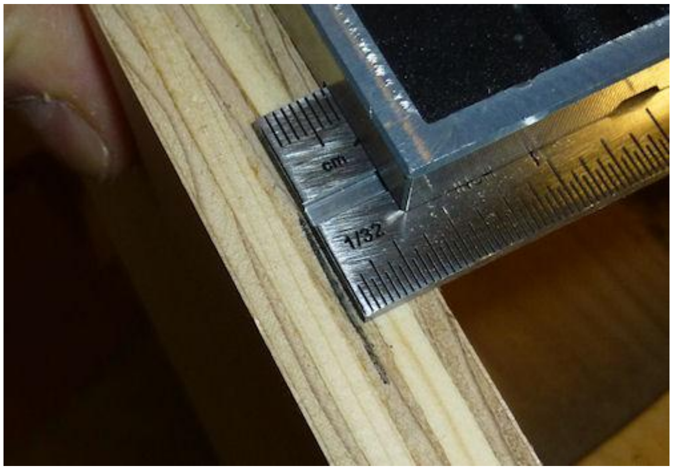
Once you have the necessary slot cutter bit, cut a slot along the top edge of the sloped portion of each side wall, centered along the edge, starting about 1½" from the front and ending at the top of the sloped section. The photos below give an overview of how you set up the bit and cut the slot.

Slot-cutter bit, 3/32" slot width, 3/8" depth (the photo shows a Freude #63-106, but other brands are available with the same specs)

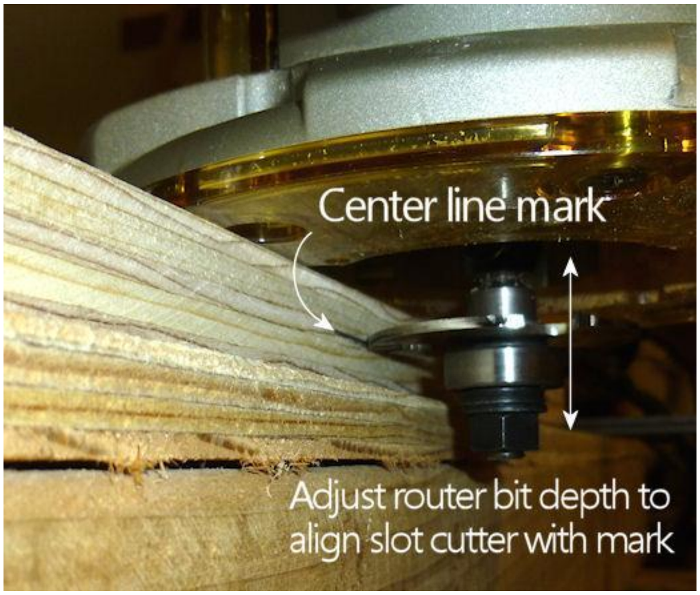


The slot-cutter bit set up in a hand router. This bit works best with a fixed-base router. A plunge router will also work - you just have to lock the depth. You can also do this with a router table, using your router table's setup for a bit with a pilot point. I'm using a hand router for these illustrations, since I've found that to be an easy way to use this bit, but the process is essentially the same with a table.

Measure the thickness of the plywood, and mark the centerpoint.



To make sure you found the exact center, flip the board over and measure the same distance from the other side. Adjust your measurement and repeat until the center mark is accurate.



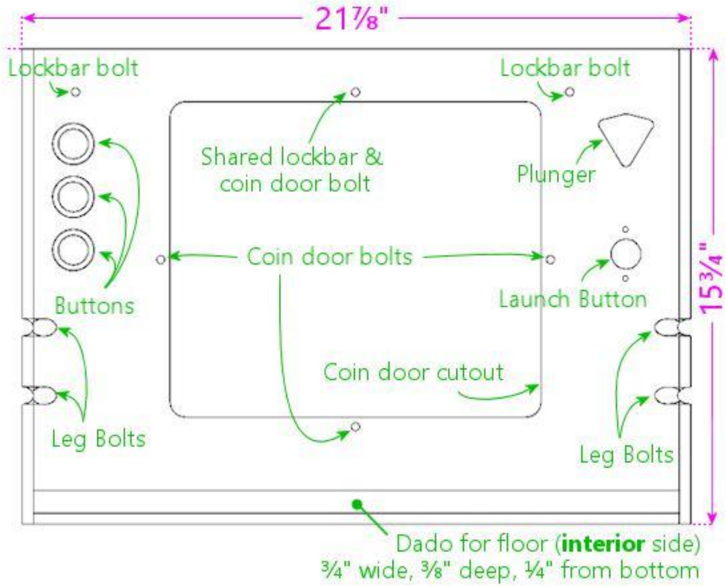
Now clamp the board to a horizontal surface. Make sure the router is unplugged! Place the router base flat on top of the board, with the bit against the edge. Using the router's cut depth adjustment (see your router's instructions), adjust the bit depth so that the slot cutter blade lines up with the center line you marked in the previous step. Lock the router at this depth - this sets the bit to cut the slot at the center of the board's edge.

You're all set to cut the slot. Plug the router in. Make sure the board is securely clamped to a horizontal surface. Place the router flat against the board with the bit hanging over the edge right next to the starting point for slot. For safety, always make sure that the bit isn't touching the work piece (or anything else!) when you switch the router on - so position it so that the bit is just clear of the work piece, at the point you want to start the cut, with the base flat against the board. Keep the router base flat against the board at all times throughout this procedure, and hold the router in both hands to keep it steady. When the router is up to speed, gently slide it sideways into the edge of the board to start cutting the slot. The bit's pilot point will automatically stop the bit at the correct slot depth, so just keep sliding it into the edge until it hits the pilot point. Now slowly move the router along the edge of the board, parallel to the edge, keeping the pilot point pressed against the edge, until you reach the end of the span where you want the slot to be. Finally, withdraw the bit from the slot, by sliding the router sideways away from the edge of the board just far enough for the bit to move clear of the slot, then turn off the router.

Front wall

The front wall is the most complex section of the cabinet. It has a whole bunch of things attached: the coin door, several pushbuttons, the plunger, the lockbar, and the leg bolts. There are so many things vying for a limited amount of space that the positioning of each part is pretty constrained; everything fits together like a 3D puzzle.

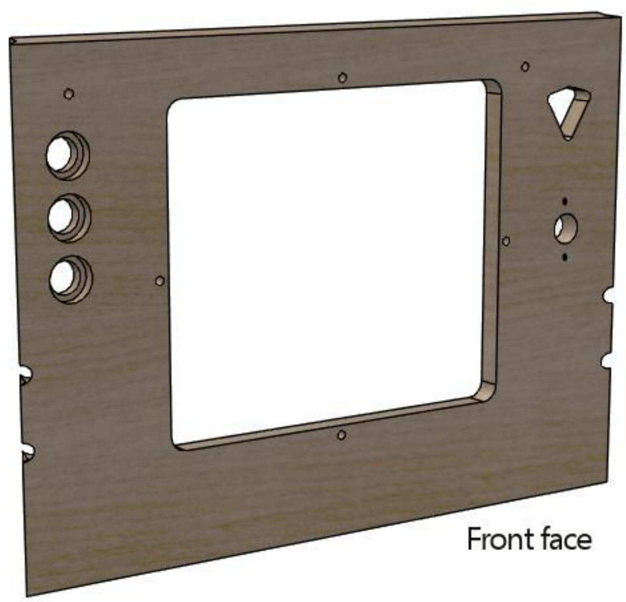
I initially tried to cram all of the measurements for all of the cutouts into a single diagram, but I quickly abandoned that idea, since it was way too busy. So instead, I've broken it out into several diagrams, one for each set of cutouts. We'll start with the basic outline and its overall dimensions, with the purpose of each cutout labeled.

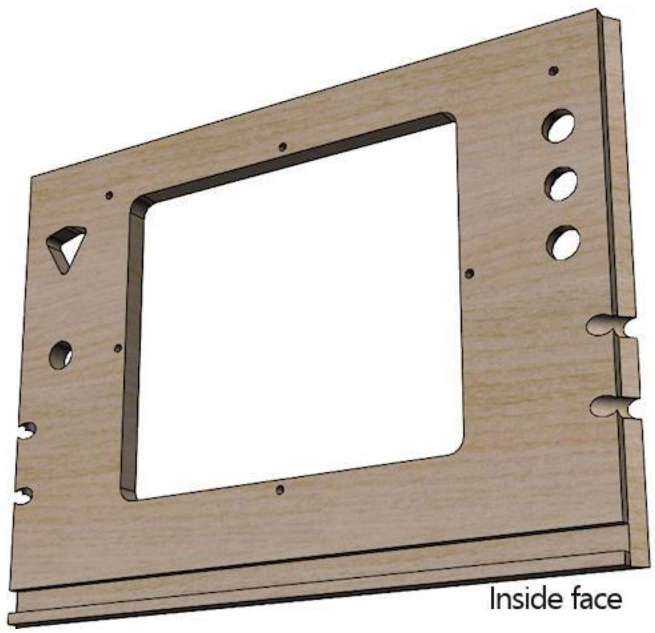


Main cabinet front panel, viewed from the front (exterior side).

Remember that we're measuring the dimensions based on a mitered rabbet join at the corners, and that you might need to adjust the dimensions slightly if you're using a different join style. See Joinery above.

More views for visualization:



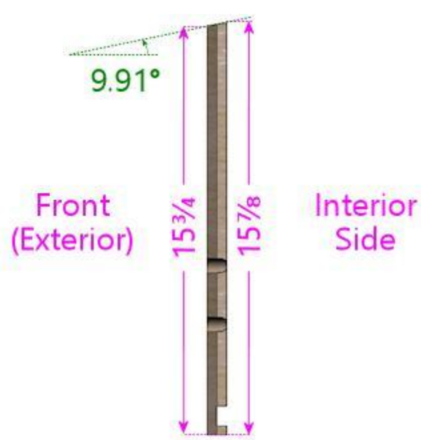


The overall width is based on the standard-body design. If you're building a wide-body or custom-width cabinet, adjust the width of this piece accordingly. Keep the coin door cutout centered horizontally at the new width, and keep the buttons and plunger at the same distance from their respective side walls.

The dado at the bottom is for joining with the floor of the cabinet. This is exactly the same as the ones in the side walls: route a ¾" wide groove to a depth of ⅜" (half the thickness of the plywood) along the whole bottom edge, on the interior side, ¼" from the bottom.

The leg bolts go through the corners of the front face at a 45° angle, just like the way they work with the side walls. Route notches for the bolts exactly as we described earlier for the side walls. Use the same positioning (measured from the bottom edge) as for the front legs on the side pieces. The front notches in the side walls need to align with the notches in the front wall when the cabinet is assembled.

If you want to get fancy, cut the top edge of the front wall to match the slope of the side walls. The slope is 10°, which corresponds to a rise of about ⅛ over the thickness of the plywood. In other words, the height at the back face (the side facing the interior of the cabinet) should be about ⅛ taller than the height at the front face (the exterior side), as illustrated below.



Side view of front panel (viewed from right) showing the slight angle at the top, to match the slope of the side walls.