1 K' 描き下ろし
2 森気楼
3 2000年

1 舞　描き下ろし
2 森気楼
3 2000年

1 [illegible]
2 TONKO
3 2000年

1 ネオジオフリーク2000年6月号 表紙　集合
2 NAOQ、北千里、TONKO、ヒロアキ、白井影二
3 2000年

1 餓狼 MARK OF WOLVES　ネオジオフリークポスター用
2 TONKO
3 1999年

GAMES 101: SHMUPS

If there's one genre that is horribly misunderstood these days, it's the venerable shoot 'em up a.k.a. shmups. You know what I'm talking about: review after review comes back stating that they're horribly repetitive and way too easy…while either neglecting to mention that the reviewer in question consumed more continues than the gross national product of Japan to pass the 3rd level or set the difficulty on 'baby' (how apt) and called it a day. Well, for those of you that aspire to be a bit more l33t (God, how annoying is that?), welcome to Games 101: Shmups Flight School.

What follows is a simple guide to help you get a bit more out of your shmups so you don't have to be like one of the aforementioned babies and use umpteen continues in order to carry the day.

Furthermore, if you follow these simple little tips, you'll not only get better and get further on one credit, it'll also allow you to save those lives (and continues, if need be) for when you really need them: at the end of the game when all reason breaks down and the programmer decided that you're going to die now… "I want him to die in the games, Sark."

Note: these tips, in many cases, are for playing to beat the game—they don't necessarily apply if your ultimate goal is high score. But seeing as how that seems to be a fairly dead art, we'll concentrate on beating a game first. Also, going for score tends to vary greatly from shmup to shmup, unlike this set of rules which is applicable to 95%+ of the titles out there.

Flight School
Keep Your Eye on the Prize

Rule #1 for would-be shmup aces: keep your eyes on your ship—I can't stress this more fiercely. How many times have you caught yourself watching something on the screen other than the general vicinity of your sprite only to watch your ship go down in a ball of fire cause you were too busy admiring the animation on a boss. Needless to say, you should always be watching your ship unless you're a mutant and can take in the whole screen and all its myriad details at once—I hear the carnival will pay top dollar for someone with your skill...alamone.

Know When to Walk
and When to Run

Another basic skill you have to internalize is the simple notion of economy of movement. In a shmup, less is more—there is no need to careen around the screen at amped up velocities to dodge a volley of enemy fire when a simple move to the side will suffice. For the most part, most of those ugly boss battles you get into that feature wave after wave of unrelenting firepower can be navigated successfully if you just relax and only move the ship in tiny increments. If you start ripping around the screen like Kodomo on crack (it ain't pretty, trust me) you might as well sign the death cert now.

Memories, All Alone
In the Starlight

Another key skill is memorization. Most shooters require at least some of this—especially when you get into some of the more advanced techniques. And I don't mean just memorizing the patterns of enemies on screen. I mean committing

to memory, in minute detail, firing speeds, enemy placement and power-up drop speeds. You need to know every last detail of what you're facing so that you're prepared for when the hurricane starts.

Greed Kills

You all know where this is going: you're cruising along, snapping up power ups, pick ups, etc. and you just have to have that one last power up, coin, doodad, etc., even though, in the back of your puny mind, you know it's a fool's errand. But it doesn't matter, you wade into the thick of it to rescue that one last coin and <BAM!> you're dead and all because you thought "just one more…" Incidentally, this also applies to games like Bomberman. Moral of this story: do not get gold fever—it isn't worth the frustration.

Top Gun In Your Face!

If there's one tactic that shooter jocks (actual and wannabe) under use, it's the concept of encroachment, i.e. getting in an enemies face and ramming both barrels down his throat—after all, if he's dead, he can't very well kill you, can he?

For example, as the screen scrolls, most of the time, you should be as close to the top (for verts) or the right side (for horizontals) whenever possible and whenever your skill level will allow. The logic here is simple: if an enemy can't get onscreen, he can't kill you. Also, as in most shooters, chances are you can fire more shots the quicker the earlier shots are off screen (or in your foe's hide).

In later levels, this can get very dangerous, so just make sure that you're on top of enemies as soon as humanly possible so they're out of the loop before you are. It may feel like you're playing by the seat of your pants (and you are), but if you can master this you'll be that much more unstoppable.

By playing like this you eliminate enemies before they become a threat. This is something you should constantly practice in your shooters, and can make harrowing experiences (such as Gunbird 2) much less so. Be warned, though, this can be a very frustrating road if you haven't memorized some basic things about the enemy in question (see above) and don't have a very solid set of reflexes.

Epilogue

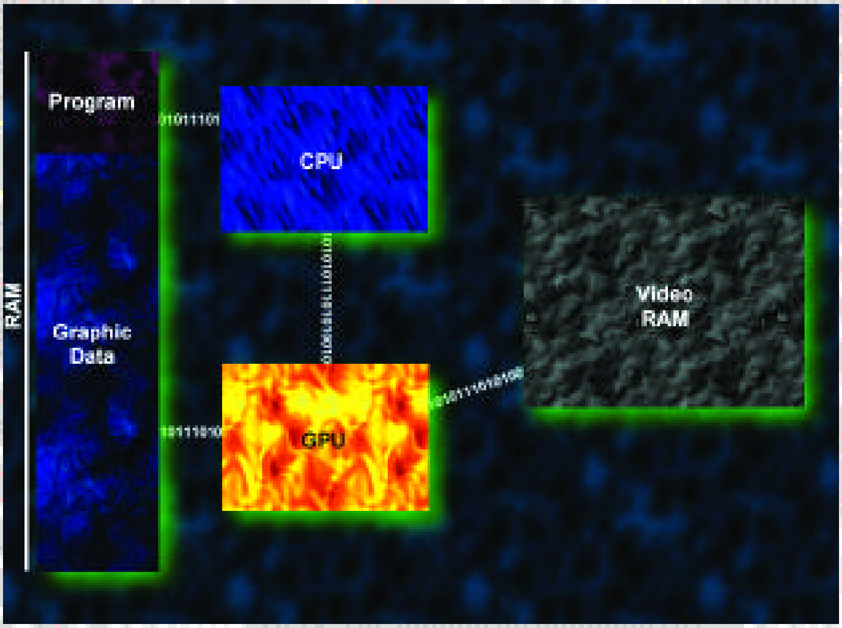
I'm sure some of you are thinking "duh" to a lot of these tips. Good for you if that's the case. You apparently are a bona fide shmup fiend and more power to ya. For the rest, though, I hope this has been at least a little enlightening and might just make you a little more eager to take another crack at that shmup you deemed impossible just the other day. And hey, if nothing else, you got to listen to me ramble for two pages—that's gotta be worth something, right?

—ECM

TECH SUPPORT

Part of the fun of running a technical website (www.gamesx.com) is answering questions from clueless people. I don't necessarily mean that in a bad way, but it's true - a lot of these people simply don't have a clue about the things they're asking about. Which is of course why they're asking. Asking questions about things you don't understand is the first step to picking up the knowledge necessary for understanding.

One of the most common questions involves video, or more specifically, doing things to the video output by your favourite console. Few people understand exactly what happens between the game CD ROM or cartridge and the screen in front of you. Here's a bit of a breakdown, divided into two parts: how the image is assembled inside the console, and how it's transmitted to the display screen.



Part the first:

Forgive me if I cover anything you already know, we're gonna start with the basics and work our way up. Your average game console isn't a lot different than your average computer, except that it's made as cheaply as possible by having everything not strictly necessary removed. If you took a standard desktop computer and chucked out everything not used for gaming, you might end up with something like this. It would contain a Central Processing Unit (CPU), some memory to work with (RAM), a Graphics Processing Unit, and a few other odds and ends to handle controller input, CD ROM access, etc. A simple, single-purpose computer.

Most consoles function the same way when creating a game. The main CPU oversees everything, tells the CD ROM chip when and what to load from the CD, ensures that controller input is shuttled off to the right place at the right time, etc. It also handles game logic, things like race-car physics, enemy intelligence, powerup locations and so on. One of the things a modern console CPU doesn't do is handle the graphics directly. Instead, as you see in (Figure 1), the CPU simply tells the Graphics Processing Unit what to do. The GPU then does as instructed - shuffling bits of background, player and enemy characters (Sprites), effects and whatnot into a special area of RAM devoted to the onscreen display. In the older consoles the CPU and GPU were quite slow, and when the limited amount of time available wasn't enough to accomplish their goals you got things like flicker and slowdown. These display artifacts are a result of the program asking for more than the hardware can deliver. It simply can't put the proper data in the screen RAM in time for the screen to be shown to you, the user.

Computers almost invariably handle graphics the same way internally: by pixels and by the colour of each pixel. A pixel is a PICture ELement - one square on the screen is one pixel. Each pixel is described by its Red, Green and Blue (RGB) value. These three primary colors make up every colour in the rainbow, and by mixing and matching them, you can create the colors needed for the graphics on the screen.

Once the picture has been created and shoved into the video RAM, it needs to be converted into a signal your display can handle. Most Graphics Processing Units output an RGB signal, one that sends each colour along a separate wire and is almost as pure a signal as the data the computer handles internally. And that brings us to...

Part the Second:

Now that the image on the screen is assembled and transmitted as an RGB signal from the video chip, it needs to be modified into something the average TV can use. Most of you have TVs that accept an RF adaptor, or Audio + Video (AV) cable. Some of you have a TV that accepts S-video, and a few of you even have one of those new fangled TVs that accepts Component Video. You might wonder which is better, and more importantly, why. Here's why.

Every time you process a signal, you change it. You may not change it much, but you'll change it nonetheless, and every change lowers the quality a bit. Inside a colour TV there are three electron-spitting "guns" that sweep back and forth, launching electrons at the screen on your TV. These electrons slam into the screen's coloured coating of phosphor, making them glow. That's how your TV works, one gun for each of the Red, Green and Blue signals. If you could wire the RGB output from the video chip directly into the guns on the back of your TV, you'd have the most perfect display possible.

Unfortunately, RGB wiring is almost totally unused in North America. The extra cost involved in cabling (more wires, complicated connectors with more pins, etc) and the lack of a connection standard have limited its acceptance. European and Japanese users have always fared a little better with the adoption of a standard RGB connector, which is built in to many TV sets. Most modern consoles also have an RGB cable available, making it easy to get the best picture possible but while most of us have RGB monitors at home in the form of a VGA monitor, sadly they operate at a far higher speed and resolution than modern game consoles. With the exception of the Dreamcast (and a possible, unreleased accessory for the PS2) all consoles require an expensive converter (See last month's review of the XRGB-2) before a VGA monitor can be used.

For most of us, the best we can get is an S-video connection from our console to our TV. An S-video signal requires a little processing to generate, however, and as you'll recall this drops the quality of the signal a little. S-video carries two signals: Brightness and Colour, also known as Luminance (Luma, or Y) and Chrominance (Chroma, or C). The names of the signals aren't totally relevant, but now you can bore your friends at parties. Transmitting the picture this way eliminates most of the dreaded crossover you get with the next two signals, which is why it's so clear. Crossover occurs when two electrical signals mix and overlap each other, and while a more expensive TV can help reduce crossover it won't eliminate it entirely. Crossover can be seen most clearly by the amount of "dot crawl" present on your screen. If you've ever seen the wavy edges of pixels sliding vertically up or down your TV screen, you've seen the dot crawl introduced by crossover. S-video eliminates this problem by keeping the signals separate so crossover is greatly reduced. S-video is still too expensive for some manufacturers, so they fall back to the old standard, composite video. Composite video is that wire attached to the yellow connector included with your NES or SNES. It carries the RGB data, converted into S-video, and then further converted into a single signal. As you can imagine, the signal has at this point suffered terribly and while it has suffered some dot-crawl and blurring compared to the previous two signal types, it's a lot better than the next one, the awful and revolting...

RF. RF sucks. That auto-switching box that connects between your TV and your cable or antenna is designed for convenience, not quality. It has taken the ugly composite video line and further insulted it by forcing it to run at a certain channel frequency (usually 2 or 3) and also stuffed the sound information in there as well! This poor signal is a far cry from the blissful purity of RGB available inside your console, and should only be used by people visiting relatives or stuck in a hotel.

That brings us to the last oddball of a signal, Component Video. Component is a bit of a misnomer, since RGB and S-video are also component formats - video transmitted by its components, not all squished together like composite video or RF. It's also known as Colorstream, Colour Difference, and YRB. Don't be confused by the Red, Green and Blue connectors they use either - that's for ease of identification, it doesn't mean it's RGB.

Component video has a lot more in common with S-video, and quality-wise can appear anywhere between S-video and RGB depending on the source. It's most commonly seen on DVD players, because that's how DVDs store their information. Component video requires a lot of calculation by both the transmitter and the receiver (your TV). It's made of three streams: Y, R-Y and B-Y. The first one is the same as S-video - a high resolution signal containing only the brightness information. The other two are mathematically generated; the first is Red minus the brightness, the second is Blue minus the brightness. The green is kind of whatever's leftover after the red and blue are removed. The two colour data signals are compressed, and usually contain half the data of the black and white signal. This means you get a crisp, clear black and white picture sloppily painted over with the colour data at half the resolution. It's a messy, tricky solution with no clear quality answers - a lot of variables are involved.

As a side note, don't forget that every step down needs to be re-converted inside your TV, back to RGB for the three guns to draw from. This means your RF signal started as RGB, dropped to S-video (A necessary step when creating composite) and then to composite video, and then frequency altered and combined with the sound data. Then it was reversed inside the TV - the sound was stripped out, the video located and removed, then separated into the Luma and Chroma streams, and then further separated into RGB and fed to the guns. That's a lot of mucking about, and that's why in this author's mind, RGB is the only way to play.

While it's true that gaming looks best using RGB, it can take a lot of effort to make the move and generally isn't going to be as rewarding to anyone but the hardcore. That's most of the GameGo readers, I think. In the next issue we'll cover what kind of hardware is needed - if you don't want to drop two hundred and fifty bucks on an XRGB-2 (What I like to call the easiest and most versatile route) I'll tell you how to bang your head against the problems of monitor selection, cable connection, and mental perplexion.

—NeoGman