

The Ergonomics of PC Video Games

I've been playing a lot of video games on my PC lately that have wildly varying degrees of success with their input schemes. The design of a game's input scheme requires forethought and attention just like any other aspect of the game, so I decided to isolate the design trends of good and bad examples. It's a little long-winded so it's all below the break:

Video game consoles enjoy the benefit of being designed exclusively for playing video games. With each generation the controllers are updated to be more ergonomic and better suited for the games which will be played on that console. Personal computers, on the other hand, have been designed for "productivity", which generally means typing arbitrary text and navigating bounded rectangular areas. And yet somehow people manage to make video games that play comfortably with this interface that has been shoehorned into the role of video game controller.

This is possible by the very deliberate design of a game's input scheme to be considerate of the position of a player's hands with respect to the keyboard and mouse. An ideal input scheme allows a user to reach all inputs he could need without changing the placement of his hands. With that in mind we can isolate three distinct input "modes" that determine which inputs are available to a player.

I've created some crude illustrations just to get the point across about where the user's hands are located, and which inputs a player can be expected to comfortably reach from those positions.

FPS mode



This is the mode most people are familiar with because its used in almost all games that use a first-person viewpoint, and because our hands are usually oriented like this when we're surfing the web. It's familiar, gives us access to half the keyboard and the mouse, and is comfortable for extended periods.

Indie mode



Named for its popularity with indie games, you'll often see this limited to the arrow keys and the z, x, and c keys. The great thing about this mode is that the arrow keys are usually the first thing people will go for when asked to move a 2D avatar about on screen.

Notable example: [The Binding of Isaac](#) is a fantastic roguelike dungeon crawler that used this mode and FPS conventions to create a robotron-esque control scheme. The wasd keys move your character and the arrow keys fire in those directions; space and e are used as action keys.

Qwerty mode



All users that know their home rows will be familiar with this input type. It's perfect if you have no need for the mouse and want lots of input keys.

Notable example: [Nethack](#) has a pretty awful control scheme, but they've certainly used this layout to its potential with almost every key having an associated action, including 8 way movement (based on Vim cursor movement keys, gross).

Mnemonic vs. Proximity

The game's needs dictate which mode the developer should select, and this in turn determines whether the key bindings will be mnemonic or proximity based. FPS and indie mode are typically proximity based: keys are selected such that the keys most often used will require the least reach of the player's fingers. In FPS games, this typically means wasd for movement with q, e, r, and f being used for important and often accessed actions. Games that use the Qwerty mode will often use a mnemonic based binding scheme: i for inventory, f for fire, c for character sheet, etc.

The flaw of choosing your key bindings based on mnemonics is of course that they will be scattered. The amount of use a key sees could have nothing to do with how hard it is for the player to reach it. This isn't much of a problem in Qwerty mode because a) all of the keys are easy to reach for your users that know how to type and b) these games typically move at the player's own pace and never require an immediate response, so it's okay if it takes a moment to find the key.

So who's doing it wrong?

This all seems kind of obvious and pedantic in hindsight because most games are really good about keeping their players comfortable. Unfortunately though there are a lot of games that just entirely disregard all of this. Genre conventions seem to play a big part in this.

The biggest offender is the ARPG genre. Having recently played a lot of Diablo III and then Torchlight II, I noticed a total disregard for order in their default key bindings. What these games have done is to use a mnemonic based scheme while requiring you to have one hand on the mouse. This results in requiring your left hand to have entire control of the keyboard, and the game requires immediate feedback from players. Combat will typically have you hovering the number keys because those are the default action keys, but you may have to jump over the other side of the keyboard if you want to peek at your inventory. I didn't last long with the defaults before coming up with my own much more sane layout involving the wasd keys.

So how do you do it right?

I'm working on a Roguelike right now in my spare time, so I've also been playing a lot. It's a really hard genre to apply a sane input scheme to, though there have been some great attempts at doing so:

Dungeons of Dreadmor uses wasd movement because it allows movement in conjunction with the mouse which it requires, but it still uses some mnemonic keybinds on the opposite side of the keyboard. **Brogue** is a little more classic using Qwerty mode (with the option for mouse controls), and even has all the keybinds you need displayed prominently on screen. The only downside here is the use of vim movement keys which just don't make sense to me (though they offer numpad movement if you have one).

AliensRL does a great job of simplifying the core gameplay experience down to only needing a few action keys, and they are all grouped around the left side of the keyboard (with the exception of weapon changing keys which are entirely mnemonic) so you can have your right hand on the arrow keys (or numpad arrow keys). If you don't have a numpad you can't move diagonally, which is a big handicap, so I essentially can't play

handicap, you can't move diagonally, which is a big handicap, so I essentially can't play on my laptop. The one that gets it all right, though not a traditional roguelike, is **Red Rogue**. The entire game can be played with your left hand using wasd movement, or you can use the arrow keys if you prefer. The menu system uses the same movement keys to navigate the menu. It's actually quite a brilliant input scheme. And the actions are available both as keybinds near the wasd keys, or via the action menu option.

The roguelike I'm working on is still in its early stages, but here are some of the things I'm planning on doing to keep things sane and accessible:

- Use wasd for movement. This way I can use the same movement keys between any of the modes if I want to make the mouse optional.
- Allow diagonal movement with the wasd keys. Players will intuitively try pressing two directions together to go diagonally, and you can do it even with tile based movement. I open sourced my **input capturing library** that can do that for you.
- Use 'e' for contextual actions: FPS games do a great job of abusing the e key. It's easy to reach and it does almost all interactions. Attack, fire, confirm, etc. They'll all be done with e.
- Use the tab key for selecting auxiliary panels. Things like your character sheet, inventory, spells, feats, etc.; these don't need their own dedicated keys. Instead they'll get their own menu which you navigate with tab key.
- Shift and Ctrl as movement modifiers: Perhaps I've just played too many FPS games, but I really like Ctrl as crouch/sneak and Shift as sprint and plan on using them as such.
- Only use 1-5 for hotkeys. 6-0 are too hard to reach with your left hand.
- Remaining actions will be reduced as much as possible and assigned to q, r, f, c, z, and x in that priority.

Taking the time to plan out a sane input scheme for your players will make your game better for everyone, and more accessible to people that aren't already insiders. Don't scare people off with vim movement keys, and don't litter your keys around the board, especially if you expect players to use the mouse.

