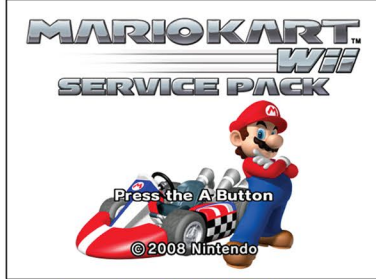


# Applying Design Thinking To Improve Video Game Menus

Zachary Rubin '24 - SOIS Digital Design & User Experience

## Intro and Background



Screenshot of the MKW-SP title screen



MKW-SP Github Repository Contributors

My name is Zachary Rubin and my major is in Digital Design & User Experience, with a minor in Game Design. My capstone was a project I have wanted to do for a while, which combines my love of both video games and user experience, and allows me to gain skills on the more technical side of game development as well.

Mario Kart Wii was a Nintendo game released for the Wii Console in 2008. As it went on to sell 38 million copies, it would go on to be a popular game for many people. The Wii was also notoriously easy to hack, allowing many games on it to develop prolific modding communities, where fans of the game create unofficial expansions and add new content.

Mario Kart Wii Service Pack (shortened to MKW-SP, or just SP) was a fan created mod for Mario Kart Wii. It initially released in January 2022 and has since gained several developers and contributors to the open source project.

Most people involved in MKW-SP were coders, so any design aspects of the project were mostly neglected or just thrown together haphazardly, resulting in an in-game settings menu created that was not ideal for both end users and developers alike. I decided to take it upon myself to re-design the menu to improve upon its many flaws.

## Design Limitations, Are They Good?



Chart comparing all 4 controllers that Mario Kart Wii supports.

Part of the reason it is so difficult for people to make a good UI for MKW-SP is that Mario Kart Wii has a lot of restrictions imposed upon it which require creative design thinking to make a solution that works around these restrictions. These include the fact that the Wii is a 16 year old console that outputs between 360p and 480p, meaning that any elements such as text and buttons need to be fairly large to be visible at such a low resolution, unlike modern machines which can output much higher resolutions. Additionally the Wii has to support both 4:3 and 16:9 aspect ratios, but since you probably aren't going to be designing two entirely different versions of the same menu for both aspect ratios, in practice this just means you're limited to the space 4:3 allows when designing menus, and in 16:9 you kind of just get more breathing room for the elements but not actually any useable space.

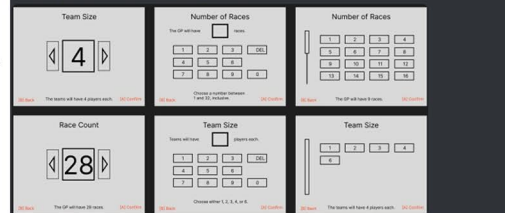
Finally, Mario Kart Wii supports 4 different control schemes, and obviously the menus need to function with all of them. Again in practice we are limited by the lowest common denominator which is using the Wii Remote on its side, which only has 11 buttons. This ends up meaning the menus essentially need to be designed around this small number of buttons if we want to give all users the same experience, though it is unfortunate that on the other controllers a lot of buttons become useless.

Thinking about menus for options that are number based. If we want a "one size fits all" solution there isn't really one perfect one, both have good (top row) and bad (bottom row) uses.

The horizontal scrolling number is the simplest and easiest to navigate and is good for settings with a small amount of numbers, and good for settings that aren't all-inclusive with numbers like team count since you'll never be able to make an "invalid" selection.

Its bad for settings that have a large amount of numbers since you have to scroll one by one through them all.

The number keypad input kinda sucks but its here anyways, since its the closest to something that exists in the base game (the FC input menu). Its only really good for settings with a large amount of numbers, but is bad for ones with a small amount or ones with invalid numbers.



My process of exploring pros and cons of different designs for the number based option menu, along with mockups

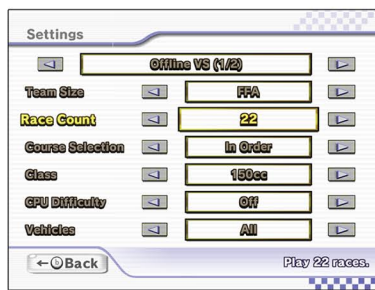
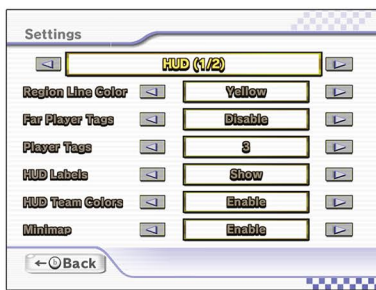
Something that I feel was important with this project was the amount of limitations I had to deal with which forced me to really use design thinking and think creatively to come up with functional solutions. Many normal computer programs these days dont have many limitations like these and thus end up all feeling sort of similar as designers default to what has been proven to work in the past. Here, however I was not able to do that which was a good exercise when brainstorming how I this system would function.

## Old Version & Flaws

Below is the old layout of the settings menu, which has quite a few flaws. The first flaw is that for each setting, you can only see one of the possible options at a time. This means if you are unfamiliar with the possible options, you have to scroll through each of them one by one to try and get a sense of all the possible options. This becomes kind of hard for certain settings like the Speedometer which has over 5 options for just that one setting, meaning you have to memorize each option and what it does if you want to compare and choose the one that is right for you. Ideally users would be able to see all options for a given setting all at once on screen for easier comparison.

The second issue is that we are limited to this single input method for each setting, where you can only scroll left or right to interact with the options. This seems ok at first, but for certain types of settings it begins to become cumbersome. For example the Race Count setting can be set between 1 and 32, but it's really bad to have to scroll all the way one by one. Additionally this limits what kind of settings could even be implemented, as settings that require different types of user input like a name entry system wouldn't be possible like this.

The third issue is the category system. Only 6 settings can fit per page so categories with more than that have to be split into multiple pages. I have seen several people get confused over the (2/2) indicator that the multi-page categories have thinking that that is the number of pages in total, not just pages for that category, thus not realizing there are a whole bunch of extra pages that they just don't know to scroll to. Obviously your user not being able to find the majority of your settings is not a great thing.



Screenshots showing the old settings menu.

## New Versions and Improvements

Although this is still a work in progress and is subject to change, this is what I have come up with. A menu based on a wheel where you can scroll up and down to choose the setting, meaning that all settings of a chosen category can be all together on one page for better organization.

To switch categories you can press + to open the category menu, which has the benefit of the user being able to parse all the categories at once rather than having to parse them one at a time while scrolling through the pages like the old menu.

Once you select a setting, you can have any sort of menu design appear for selecting the option. For most settings which have between 2 and 5 options, we have this menu. It shows all of the options nicely laid out for the user, as well as the descriptions for each, which can help the user compare between the options and pick the one that is right for them. You can then click on the option to make the star icon move to it which shows that that option is now the currently selected one. Then when you exit the menu the text on the right says the current selected option.

For settings that have more than 5 options, unfortunately we cant display every description on screen at the same time, but we can still display each option and have the description appear in the bottom right when said option is being hovered on. And once again you can click on a button to select the option and make the star icon move there.

For number based settings I decided to make a grid of buttons, since users don't really need the description for each individual button here and since numbers are a lot shorter than option titles, I could fit a lot onto one page to minimize the time spent scrolling to find the correct number. Eventually for options needing more than 30 numbers we will add pages to scroll by 30's, though this has not been implemented yet.



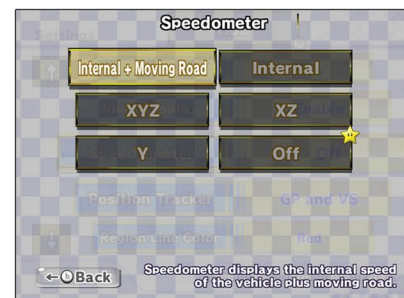
Screenshot of the main settings menu wheel



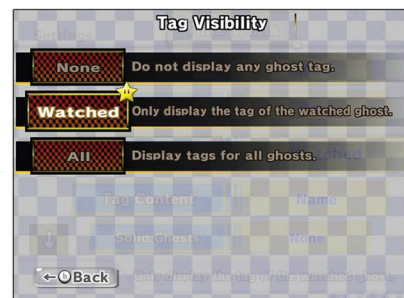
Screenshot of the new category select page



Options page for number based settings

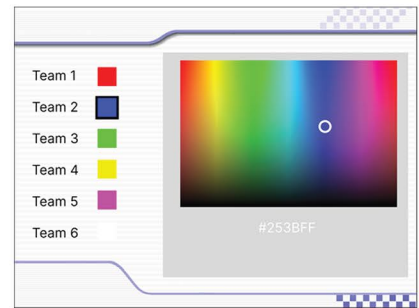


Options page for settings with 6 or more options



Options page for settings with 5 or less options

## Future



Mock-up for a team color customizer screen for color blind accessibility

What has been shown here is what has been done so far, but in future I want to do many more things with this. We have settings for volume which I want to have a volume slider for as that is the most intuitive method for controlling volume across many applications. We also have a colorblind team color setting which is currently hardcoded to only one set of alternate colors for teams. This isn't really ideal since colorblindness comes in many different types so there is no "one solution fits all". In the future I want to implement a color picker menu so people can individually tune the team colors so they can tell them apart no matter how good your vision is. This wouldn't have been possible with the old settings menu, but with my system we can add a new UI page just for the colorblind setting.

The process of creating the menus was interesting, as I was working with the developers of MKW-SP in collaboration to make this happen, I got some good experience about how articulate I need to be when describing ideas as sometimes there would be miscommunications when I wasn't detailed enough. Same thing when it came to testing and reporting bugs as since this is a small team I became the de-facto QA tester as well.

I plan to continue to work on this until it is fully complete, polished and tested and then hopefully merged into the main MKW-SP branch.

Ask me for a live demo!  
Feel free to give suggestions  
and ask questions!