

Menus Matter



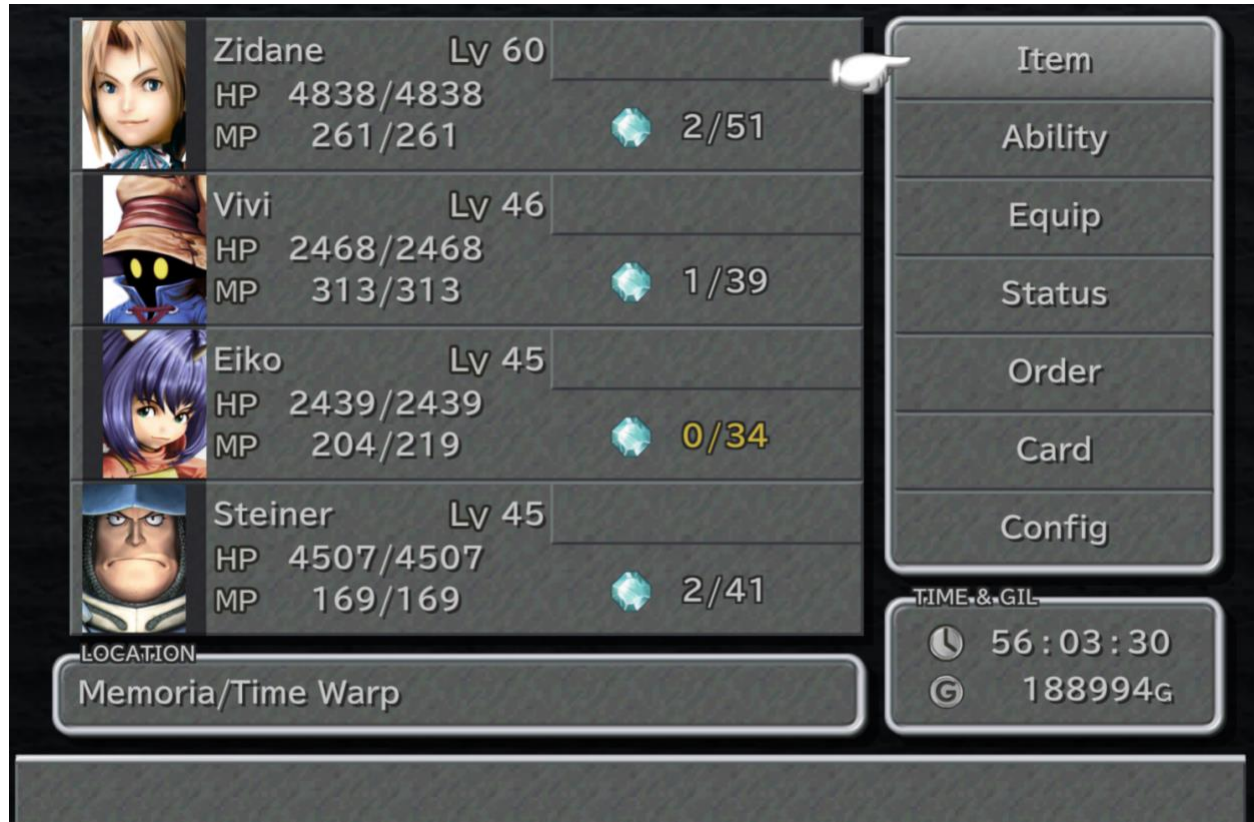
It seems in many western games; the developers consider user interfaces and menus a cost of development instead of a feature. I'm not going to call out specific franchises but think to the last 'open-world' AAA game you played and think about how annoying it is to do anything with the menus. A lot of these games have clunky menus that take too long to navigate or are confusing to navigate. Also, very important is that those menus aren't satisfying.

This is a factor I considered while designing the menus for my capstone class, since it being an action game with role-playing elements, utilizing menus will be a large part of the user experience. To create a cohesive and enjoyable experience for those utilizing the menus in my game, I researched menus that I've liked using in the past. Below are the results of that research.

What are Menus

Menus in games, particularly action or role-playing, are the means to have the user be prepared to interact with core mechanics. They are configuration interfaces, not just to application settings, but also the properties of a player character.

Menus in JRPGs

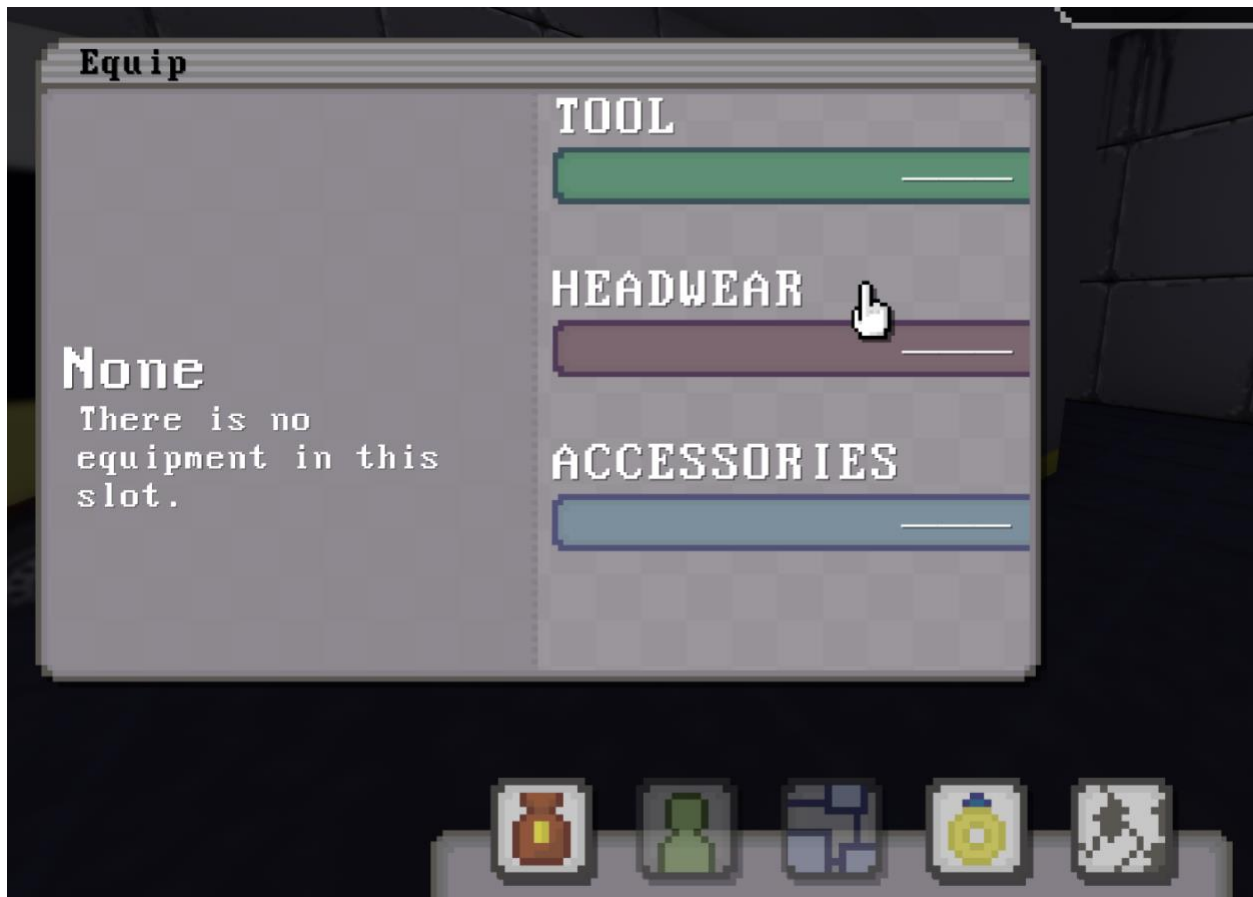


Japanese role-playing games are a particular flavor of role-playing that most commonly have a focus on menus, they're usually turn-based, and have roots in retro gaming. Games like Dragon Quest, and Final Fantasy have their core mechanics controlled through menus. Everything from equipping armor to attacking is done through menus. Naturally the "menuing" experience must feel good to use. Nobody is going to play a game where 80-90% of the game is bad UI.

Qualities of Good Menus

I. Straightforwardness

A menu should do exactly as it says with little need of explanation, a player should look at whatever text a menu displays and know instinctively what it does. The mind naturally abstracts visuals in a hierarchical manner, from **container**, to **component**, to **action**. Let's break down an example from a current project I am working on:

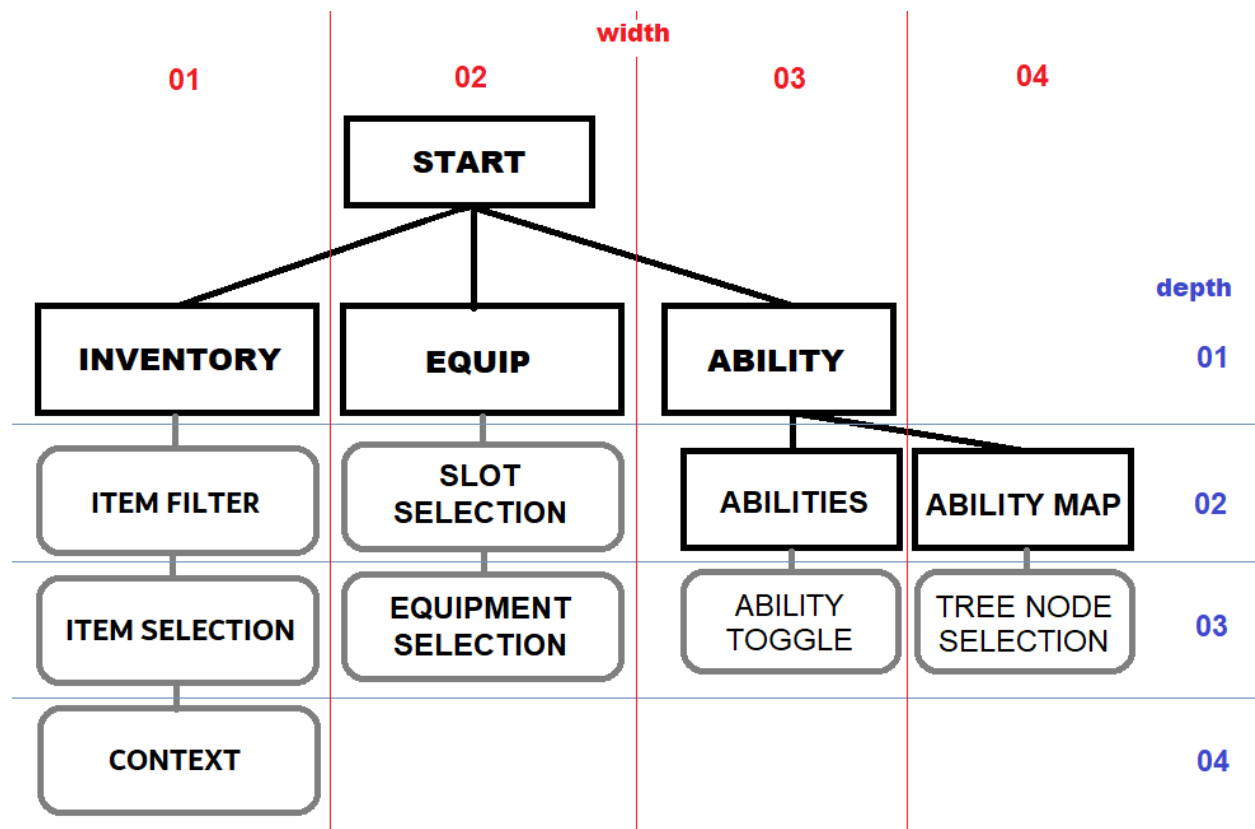


Shown to the right is an equipment menu. I likely didn't even have to say that because it's evident by the *container* titled "Equip." A '**container**' is a shape that indicates it holds contents, particularly *components*. **Components** are features of a UI that display or offer change of parameters, and generally do one thing (may also be contextual.) A player selects a component to perform an action. The *components* shown in the image are "Tool," "Headwear," and "Accessories." These components imply action when selected if you select "Tool" in an Equip container, your assumption should be "I'm going to equip a Tool". The major difference between containers and components is that components represent data for interaction, and containers *hold* the data to interact with. Finally, **Actions** are the *selection* of a *component* which implies the change or display of data or state.

In short, a user views a *container* to *select* a *component* which will perform an *action*. Using this design framework helps to create a menu that is intuitively straightforward.

II. Simplicity

To extend straightforwardness, a good menu maximizes simplicity, meaning a player takes as little actions as possible (within reason,) to achieve a task. This is the major issue I have with modern western game menus. One of those "AAA open-world role-playing" games I played last year had menus on menus on menus. Nesting menus is necessary to keep a structured flow, but there's an upper limit to how many menus you can layer before you have a **UI maze**. Imagine associated menus as a tree:



Black boxes are containers, gray boxes are components. In the above example, the complexity of this menu chain is 4x4. The player will navigate no more than 4 , and will have to make no more than 4 divergent choices. (Notice "Abilities" doesn't get its own column as child menu choices have one choice that follows a forward chain, no navigational choice.) I believe a complexity of 5x5 is what I would consider a maximum, I tend to get lost in game menus with a complexity greater than that.

III. Style

Styling a menu includes both visuals and sounds. There isn't really an objective way to cover this as it enters the realm of art, which is outside the scope of this article.

However, this aspect is objectively important for games that feature a lot of menus as it adds a great deal of personality. Put a lot of care and consideration into creating a style for a menu that matches its brand identity and stands out against the mountain of rushed utilitarian menus developed by your standard AAA studio.

Menu Design Sins

In no order, a list of menu design choices that I see often in games that bother me.

- Stopping music while in a menu: Just because the action stops, doesn't mean the tension should have to. I've played several games where you open a menu to change equipment, and the music just stops. It takes the player out of the experience and makes it harder to resume with the same amount of interest.
- Delayed feedback: Menus that have animations that are too long, and menus that have input delay. When I interact with something, I want instant feedback. An animation should last no more than one second and should start moving the moment I press the input.
- Bait-and-switch options: If you have ever played a game with dialog trees, often the writer makes the option shorter than the actual response, which makes sense. However, there's a limit to how different an option selection can be when compared to the actual response. Menu selections should do exactly what it says on the tin, and not much else.

Closing thoughts

Design is only half the battle, while having good ideas of how you're in-game menu should be, implementation could be your weakest link. Consider code performance, and reusability. I've heard that UI programming is soul-sucking work, and while I personally like it, make sure to create a UI system with elements that can be reused and redesigned with ease. The Unity Engine has a implementation of this concept called [UI Toolkit](#), I'm using it in my project and I'm impressed with how easy it is to create unique and reusable game menus.

If there's anything to take away from this article, it's to give pause when designing a menu. It's very likely that even in non-role-playing game genres, your player will be using a lot of menus. Menus in games can be a feature, and not a cost of development.

