Style Guide: UI Map

What is a UI Map?

A UI Map is a layout of menus that is arranged into a flowchart. The UI Map is used to help understand the navigation path and potential roadblocks within the UI menu design. Every menu has a list of options that the user can interact with. These options usually will link to another menu or input decision. All options, features, input requirements or notifications should be documented in the UI Map.

How is it used?

UX designers and engineers use a UI Map to understand and quantify the work needed to create a functional menu system within a product. The UI Map is used to track and adjust the navigation of the menus and can also be used to identify areas in which the menus can be optimized for development and production.
Any additional features that are added during production can be entered into the UI Map and identify any flow issues. By using the UI Map as a roadmap, it will help with any questions before any production is scheduled.

A more detailed interactive version can be created to identify any flow and messaging issues.

Get all stakeholders on board

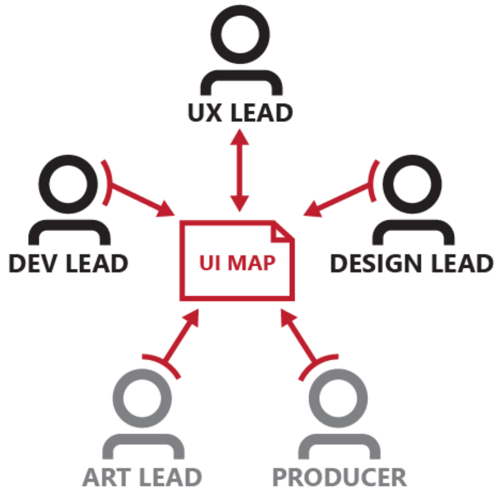
It is very important to create a UI Map in the very early stages of a product's development. After core features have been identified, the next step is to create a UI Map. This should involve the UX Lead, Design Lead and Engineering Lead so that everyone is on board from a functional standpoint. Once the UI Map has identified all the areas within the menus, the Art Director and Producer should be consulted to understand the scope of the intended flow.

By getting all the lead stakeholders of the project involved at this early stage, the UI Map can act as a way to communicate any technical, feature or artistic changes that will occur during production. The UI Map will have to be updated with any changes and then can be tracked by the leads and producers accordingly to help with production deadlines and budget concerns and staffing.

Layers of Information

There are many variations of a UI Map that can be created, depending on who will be using it as a reference guide in development. To make the UI Map useful to the lead stakeholders, break out the details into three main areas of information:

- Navigation *(basic flow)*
- Interaction *(console buttons, touch gestures, voice)*
- Backend integration *(APIs)*



Use the UI Map to help all disciplines understand the scope and design of the project throughout production.

Navigation

It is important to keep the navigation to areas within the UI Flow to a minimum and some designs will incorporate shortcuts to circumnavigate an entire menu section.

Use the UI Map to count the number of screens the player has to move through to complete actions. The most important actions should be easily accessible from a central location and be completed as quickly as possible – but don't be afraid to move the more complex or hardcore actions deeper in the flow.

Beware of too many entry or exit points on the same screen. It's tempting to give the player as many shortcuts between screens as possible, but over redundancy makes it difficult for the player to develop a consistent mental model of the UI flow. In other words, if your UI Map looks like a rat's nest, it's going to feel like one too.

Interaction

The standard interaction of menus within a video game is A button to enter or select, and the B button to cancel or back out of a menu. It is also important to call out any special buttons used on a screen.

Review the whole UI Map to make sure that you're using the same buttons consistently to perform the same actions. (For example, don't use left/right bumpers to tab through a list on one screen, and then use left/right triggers to tab on another screen.)

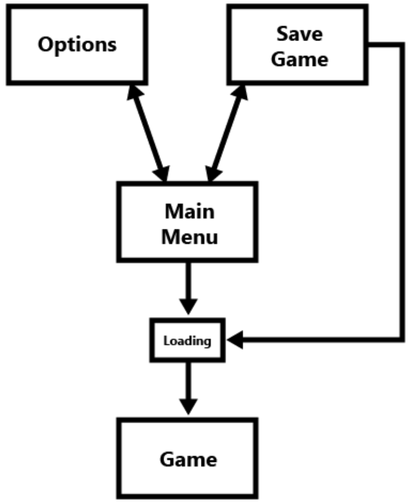
Levels of Detail

Before diving deep into the details, think about creating the UI Map like a painting and start with a high level sketch, and work in the details as you refine the UI Map with the lead stakeholders.

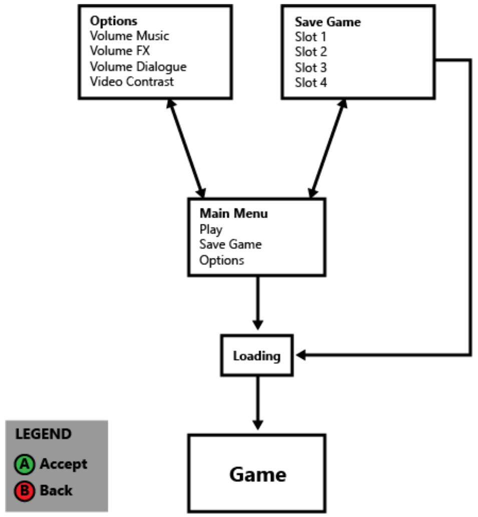
The first version should be high level to record all the design features that the product wants to interface to the user. Each menu should be high level enough that each interaction is recorded as a new menu or notification and can easily show how the user can navigate the menu UI.

The second version is a detailed listing of options and features within each menu screen. This will help with determining template patterns and component reuse. This level of detail will help with engineering, design and production scope of the project.

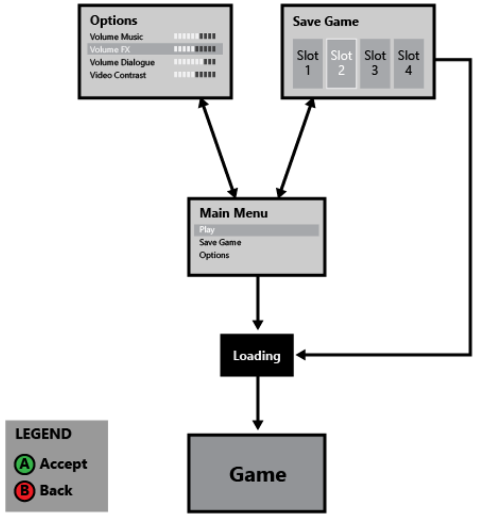
The third version is optional, but this version can be used to show off that actual menu and feature designs, and can also be used to create an interactive walkthrough prototype before production begins.



V1 - Basic Feature Set

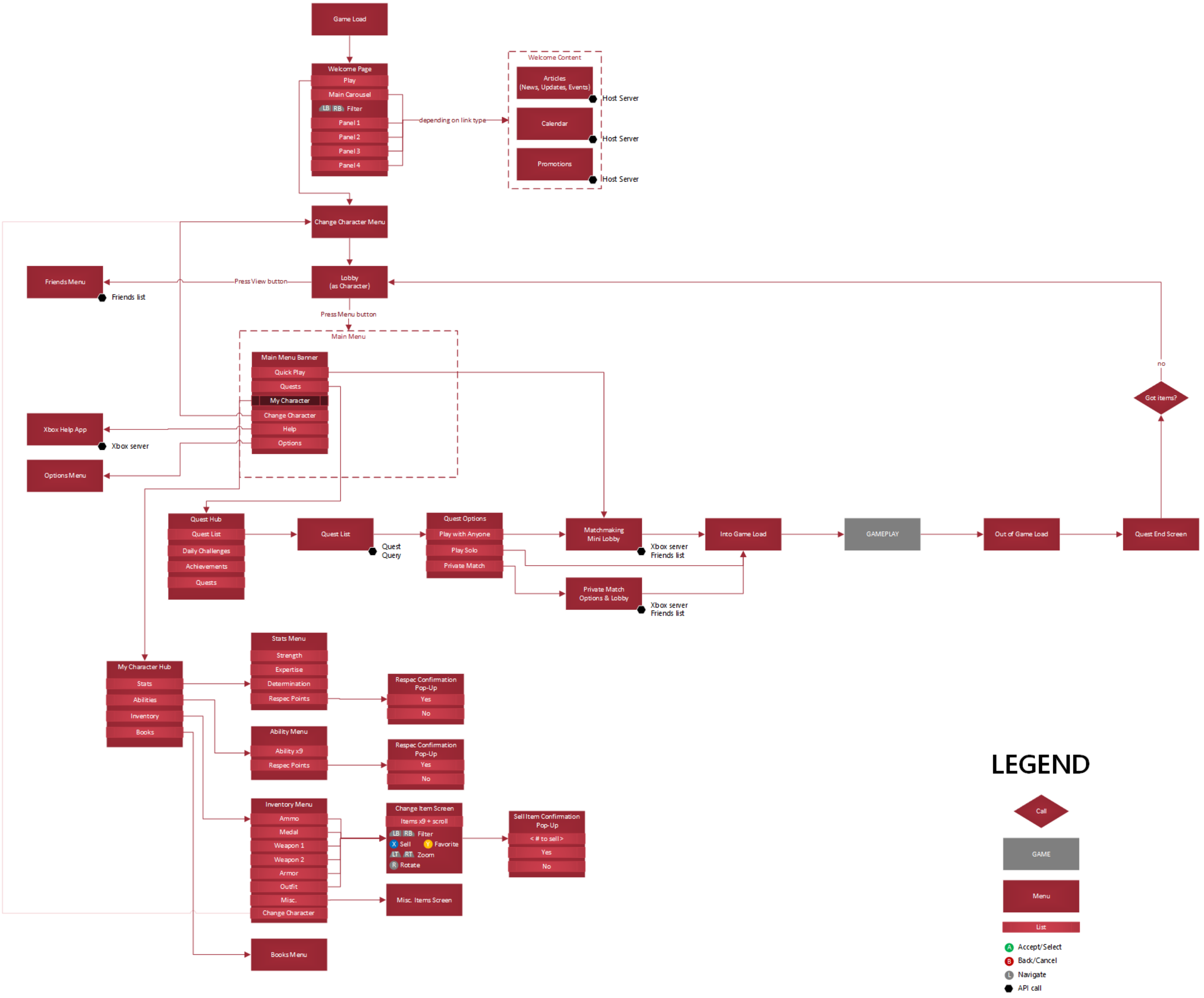


V2 - Detailed Information

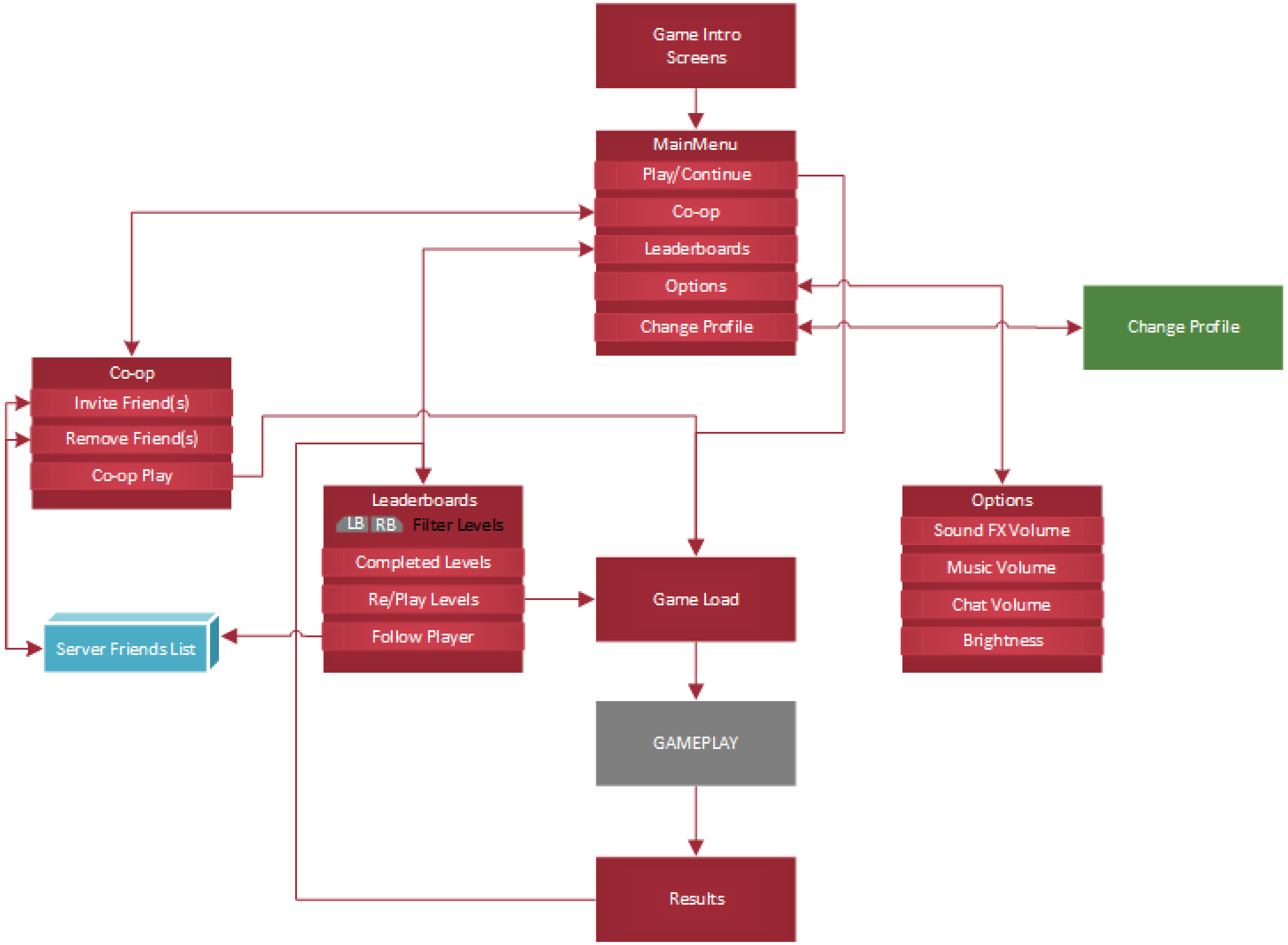


V3 - High Level Features

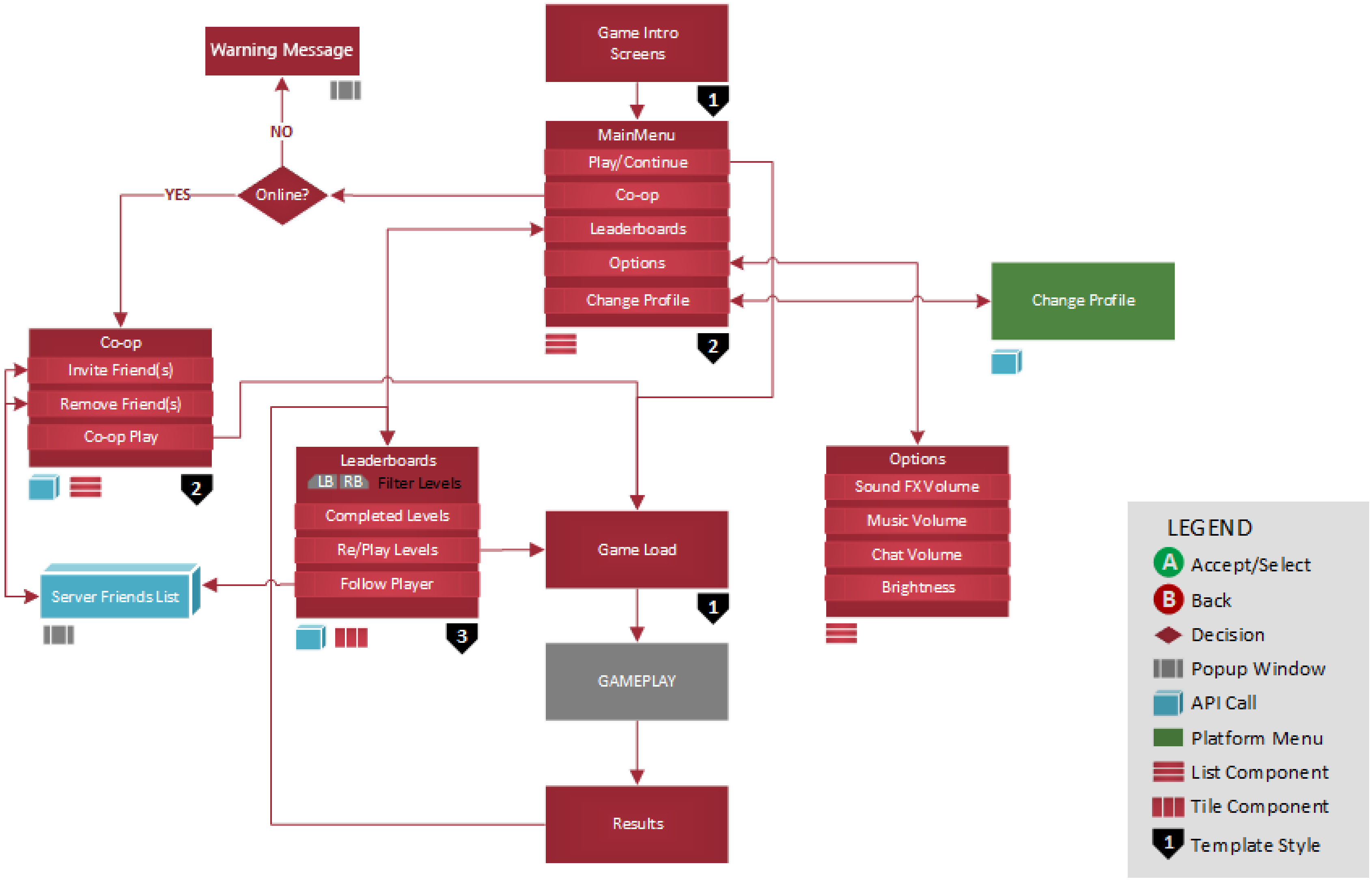
Examples



Detailed UI Map with Legend



Basic UI Map showing menu flow



Detailed UI Map with Legend that indicates templates, components and API calls.

UI MAP: Best Practice Checklist

□ Consult with Lead Developer and Lead Designer to get a full understanding of all the products features.

□ Start with a basic feature set for the UI Map and in detail when the basic flow has been approved.

□ Get all lead stakeholders to use the UI Map as a guide to communicate features and navigation throughout the product.

□ Update the UI Map every time there is an addition or deletion of a feature, screen, component or interaction.

□ Create three different layers of the UI Map that call out: Navigation, Interaction, and Backend Integration.

□ Create a Legend to help distinguish between device input methods, shape and color of each menu within the UI Map.

□ Call out any special button interaction and be consistent in their usage throughout the UI Map flow.

□ Keep the UI Map organized and easy to follow the flow. Do not force everything into a small screen space.

□ Make sure the most important actions are easily accessible from a central location. Move hardcore actions deeper in the flow.

□ Be sure not to add too many entry and exit points (shortcuts) on the same screen.

□ Recommend using a software that allows for easy iteration.