

Star Trek: Supremacy
User Interface Guidelines

Version 0.1
15 April 2006

Written by Mike Strobel

Table of Contents

Layout Considerations	1
Resolution Independence	1
Inconsistent Layout across Themes	1
Design Considerations	2
General Appearance First	2
Don't Limit Your Creativity	2
Make Your Components Reusable	2

Layout Considerations

Resolution Independence

Star Trek: Supremacy is designed to run at the player's desktop resolution with a minimum resolution of 1024x768 and a recommended resolution of 1280x1024 or greater. While some UI components will retain a fixed size at any resolution, others will be expected to grow and shrink based on the amount of real estate available. Figure 1: Layout Scaling demonstrates how the parts of an interface may be expected to adapt to different resolutions, with arrows indicating the direction(s) in which a component's size may expand or contract.

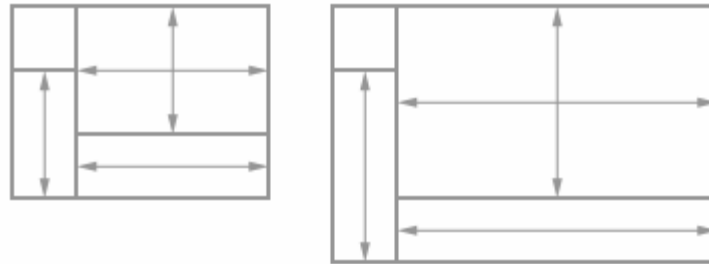


Figure 1: Layout Scaling

When designing a new UI component, it is critical to consider whether or not the component will be expected to change dynamically, and if so, whether or not there shall be any minimum or maximum constraints on the component's width or height.

Inconsistent Layout across Themes

The decision to use the Microsoft Windows Presentation Framework (WPF) for the *Star Trek: Supremacy* user interface was heavily influenced by its support for control styles and templates. Most WPF controls enforce a strict separation between the look of a control and the control's function. By applying a new template to a control, the look of that control may be drastically altered from the standard look envisioned by the control's creator. This principle introduces an important layout consideration, as one may not assume that a control's dimensions, shape, opacity, or margins will remain consistent from one UI theme to another. However, within the scope of the *Supremacy* project, UI designers may assume that a control's dimensions will not vary by a factor exceeding +/- 50% from the control's default template. Template designers shall be expected to honor this restriction, though exceptions may occasionally be made pending the approval of the Project Manager and the Lead Developer.



Figure 2: Standard vs. Custom Control Template

Design Considerations

General Appearance First

When designing a new UI component, the designer may be tempted to draft mockups based on a specific UI theme, such as the "LCARS" theme. However, it is recommended that designers limit their initial design mockups to a "general" black-and-white appearance. This will allow the team to evaluate a design's merits based strictly on layout and functionality. Once a component design has received sufficient positive feedback to warrant further development, the designer may begin to provide mockups of how the component might appear across various UI themes. This will ultimately save the designer time, as he or she will not invest unnecessary creative talent into a preliminary design that is likely to undergo several revisions before being accepted. Note that this guideline is simply a suggestion, and UI designers are encouraged to work in whatever manner they find most productive.

Don't Limit Your Creativity

UI designers may feel obligated to keep their designs simple based on the somewhat outdated convention that game interfaces are static and inflexible. However, the WFP's control template and data binding support afford designers virtually limitless possibilities¹. Designers are encouraged to review some of the Avalon/WPF technology demos readily available on MSDN and other websites to get a feel for how templates and data binding work, and what new possibilities they open up.

Make Your Components Reusable

While some of the proprietary UI components created for *Star Trek: Supremacy* may be highly specialized, there may be cases in which similar functionality is required in different places. For example, the same build queue item template might be used for both planet-side production and orbital ship production. In this case, it would be both wasteful and unnecessary to produce two different templates for build queue items. Reusing the same template would not only increase consistency across the entire game UI, but would reduce the number of templates that would have to be re-styled for each UI theme. UI designers are encouraged to examine their component designs and evaluate whether or not a component could be usefully broken down into two or more individual components.

¹Many technology demos of the WPF may be found at MSDN Channel 9 <<http://channel9.msdn.com/tags/Avalon>>.