

Design Paper: Plataforma de Construção de Ferramentas via UI Retained■Mode

Resumo Este Design Paper descreve a arquitetura e componentes de uma plataforma de UI retido■mode, destinada não só a renderizar interfaces finais, mas a construir outras ferramentas (editores de animação, partículas, materiais, etc.). O framework une práticas de sistemas consolidados (Unity UI Toolkit, Unreal Slate, Qt/QML, WPF) e introduz um modelo data■driven, extensível via plugins, com ênfase em performance AAA, modularidade e reusabilidade.

1. **Introdução Visão:** Criar um ecossistema onde cada widget é um bloco de construção para editores avançados. **Motivação:** Ferramentas modernas de produção exigem editores internos sofisticados — curve editors, graph editors, particle editors — que compartilham infraestrutura de UI e recursos de renderização.

2. **Requisitos de Alto Nível** 1. Retained■Mode e Batching de draw calls. 2. Data■Driven / Declarative UI (UXML/JSON + USS/CSS■like). 3. Extensibilidade via módulos/plugins C++ registrados em tempo de execução. 4. Designer■First: WYSIWYG, live preview e hot■reload de definições. 5. Ferramentas Composáveis: flow graphs, curve editors, particle editors, material graph editors. 6. Suporte Vetorial: SVG e ícones de fonte. 7. Temas Dinâmicos e skinning completo. 8. Performance Escalável: grandes árvores de UI, centenas de nós de grafo, milhares de partículas.

3. **Panorama de Sistemas Consolidados** - Unity UI Toolkit: Retained■Mode, UXML + USS, C# scripts, Batching. - Unreal Slate: Immediate/Retained, C++ Widgets, Plugins C++, Manual batch. - Qt Quick (QML): Retained, QML + JS, C++ Extensions, SceneGraph. - WPF: Retained, XAML + Data Binding, .NET Assemblies, Visual Tree.

4. **Arquitetura Geral** 4.1. **Camadas Principais** - Core: UIContext, EventBus. - Widget Layer: UIElement, UIContainer, UIScrollView, UIButton, UICollection, UIFlowGraphView, UICurveEditor, UIParticleViewport, UIMaterialViewport, UIImageSVG, UIText, UIColorPicker. - Layout & Styling: LayoutEngine (Flex, Grid, Absolute), StyleSheet, ThemeManager. - Tooling API: FactoryRegistry, ReflectionBindings, GraphCompiler.

5. **Modelo Data■Driven** 5.1. UXML■Like (UI Definition) [Exemplo de UXML]

5.2. USS■Like (Styling) [Exemplo de USS]

6. **Ciclo de Vida e Dirty■Flags** - Hot■Reload, Update, Layout Pass, Render Pass.

7. **Extensibilidade via Plugins** - Registro de widgets e bindings automáticos.

8. **Ferramentas Comuns** 8.1. Flow Graph Editor: NodeModel, EdgeModel, UIFlowGraphView, Bézier Paths. 8.2. Curve Editor: Curve, UICurveEditor, keyframes, tangents. 8.3. Particle System Editor: ParticleSystem, ParticleEditor, UIParticleViewport. 8.4. Material Graph Editor: Model, UIFlowGraphView, UIMaterialViewport, GraphCompiler.

9. **Performance & Otimização** - Batching, Pooling, Atlasing, Dirty■Flags, Threading.

10. **Exemplo de Fluxo de Desenvolvimento** - Definição, Estilo, Implementação, Hot■Reload, Extensão de módulos.

11. Conclusão Framework unificado, data■driven, extensível, para construção de ferramentas internas AAA.

12. Referências 1. Unity UI Toolkit — UXML & USS 2. Unreal Engine Slate UI Framework 3. Qt Quick (QML & SceneGraph) 4. WPF Architecture & XAML 5. NanoSVG — SVG tessellation library