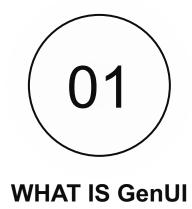




CONTENTS









GenUl

GenUI是一个创新的SFP前端框架,使用Rust语言开发,最初受到Vue3和Makepad的启发。旨在帮助用户更有效地使用Rust编写前端项目。

What

- Rust
- SPA
- SFP
- DSL in .gen file
- Flexible

GenUI

GenUI的SFP功能是指其可插拔设计,可以根据开发者的需求调整和使用不同的底层技术或框架,如Makepad或Slint。这种设计允许GenUI通过专门为这些底层框架设计的插件来扩展和自定义其功能,从而更好地适应各种开发场景。

WHY SFP

- 通过插件提供强大的能力
- 不受固有底层限制
- ●同时也是一种挑战







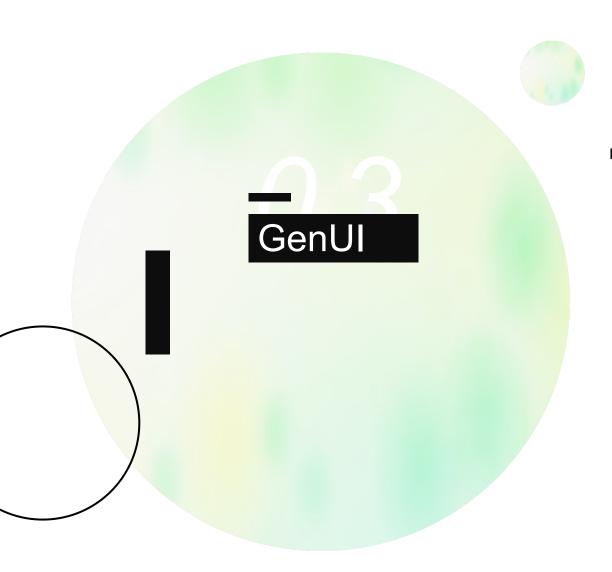
Ability

多框架兼容性

⊘ 灵活

○ 可扩展

尊重用户



Target Group

- 前端开发人员和全栈开发人员
- 创新和实验项目的开发人员
- ●开源社区
- 独立开发人员
- 中小型项目开发 (现在)

GenUI

在GenUI中,代码编写并不是随意的。它要求开发人员遵循一套严格而灵活的规则。这种设计理念确保了代码的一致性和可维护性。

Design

- DSL
- Parser
- Converter
- Generator





<template>

语法让人联想到HTML,但经过了增强,可以进行更结构化的处理。

保持清晰、重点突出的模板的主要注意事项:

- 尽量不要使用直接字符串文字。禁止使用类似JS的模板语法;请使用Rust的format!改为值绑定。
- 2. 仅使用基元类型声明标记上的基本属性;复杂类型应该单独绑定。
- 3. 函数体不能直接写入属性中;相反,请使用函数绑定。
- 4. 属性是强类型的,确保了清晰明确的类型关联。

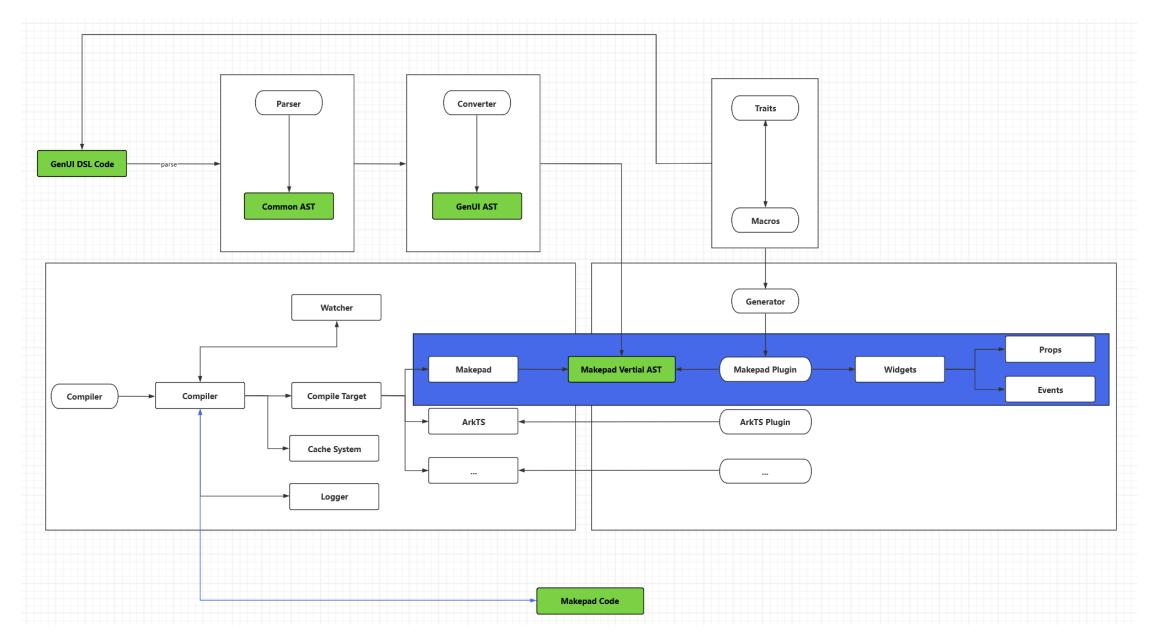
<script>

- 1. 提供了对Rust语法的完全支持,允许使用强大的脚本功能。
- 2. 允许与特殊框架集成,从而扩展了功能和定制的可能性。

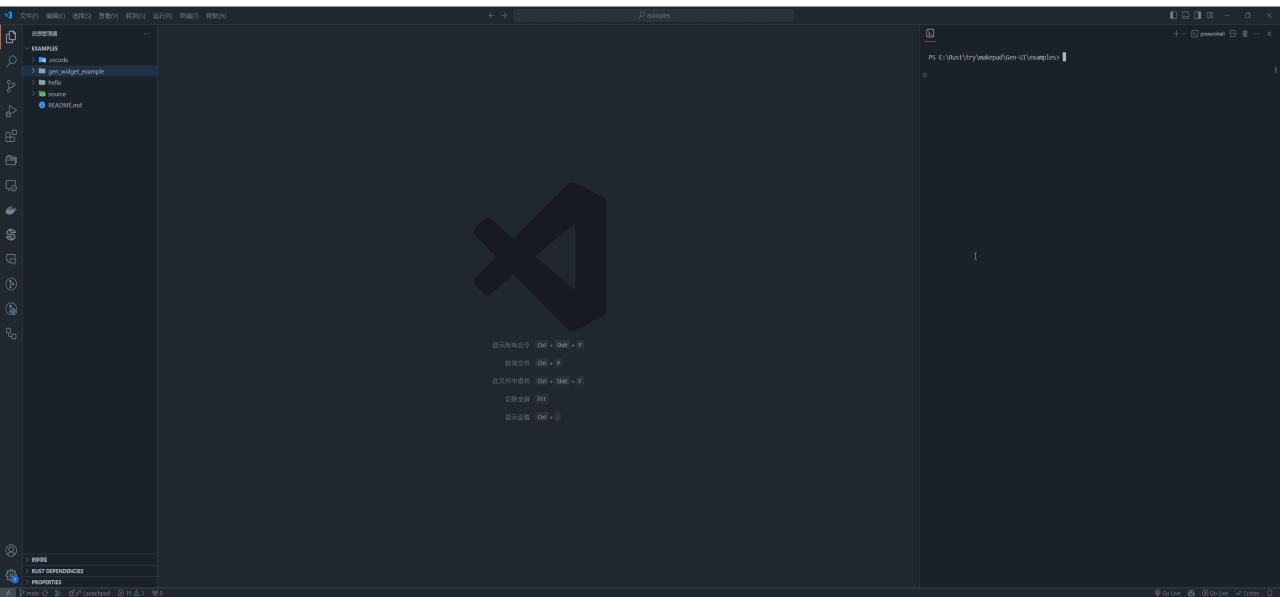
<style>

- 1. 样式通过名称绑定到标签,从而实现简单直观的样式设置。
- 2. 允许且建议使用嵌套样式,提供了一种反映传统CSS的分层样式方法。
- 3. 样式中支持函数和绑定,允许基于逻辑和条件进行动态样式设置。

Work Flow

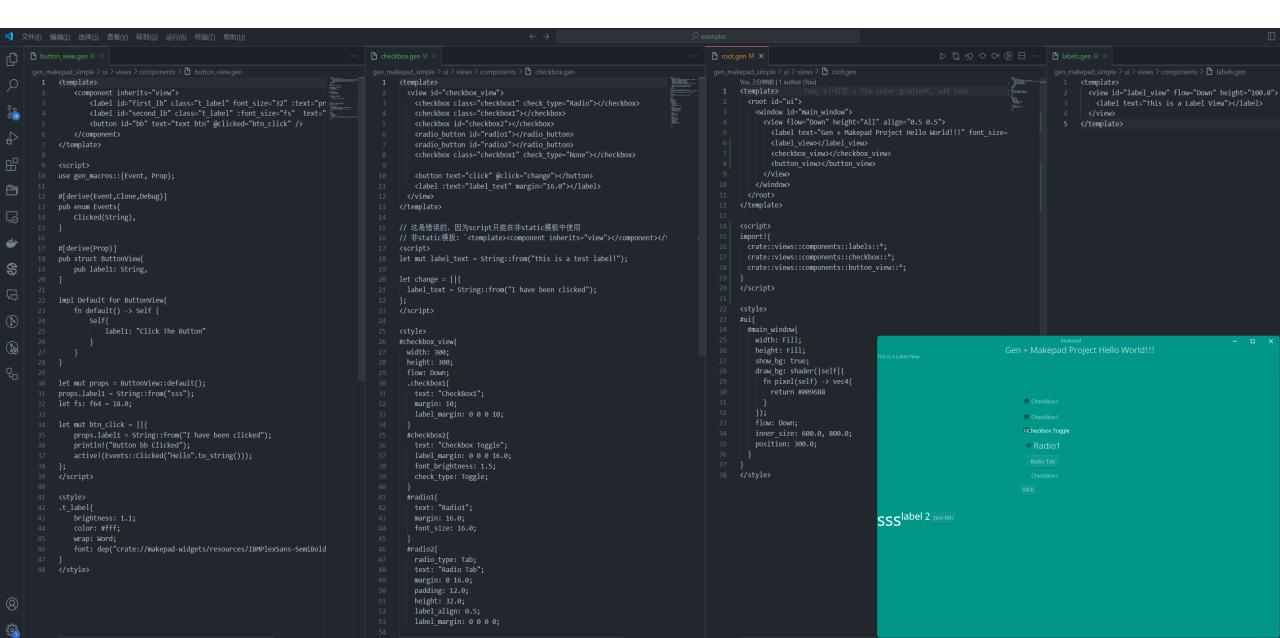






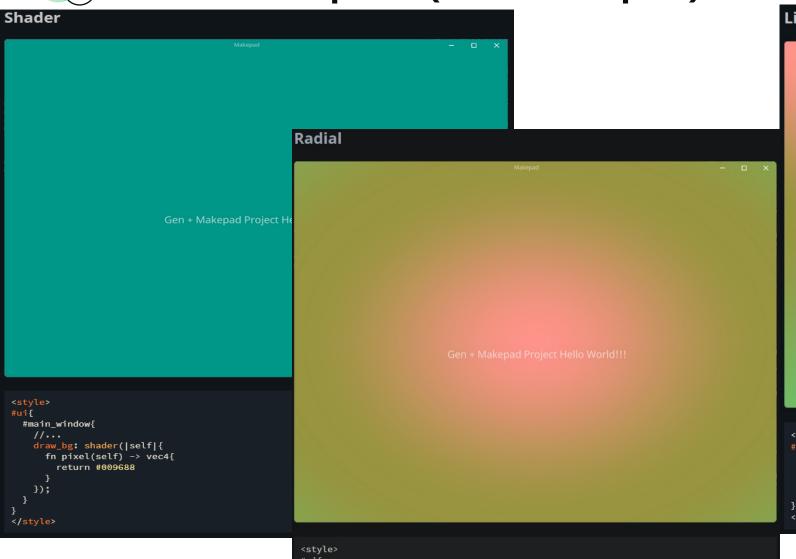


Other Examples (with Makepad)





Other Examples (with Makepad)

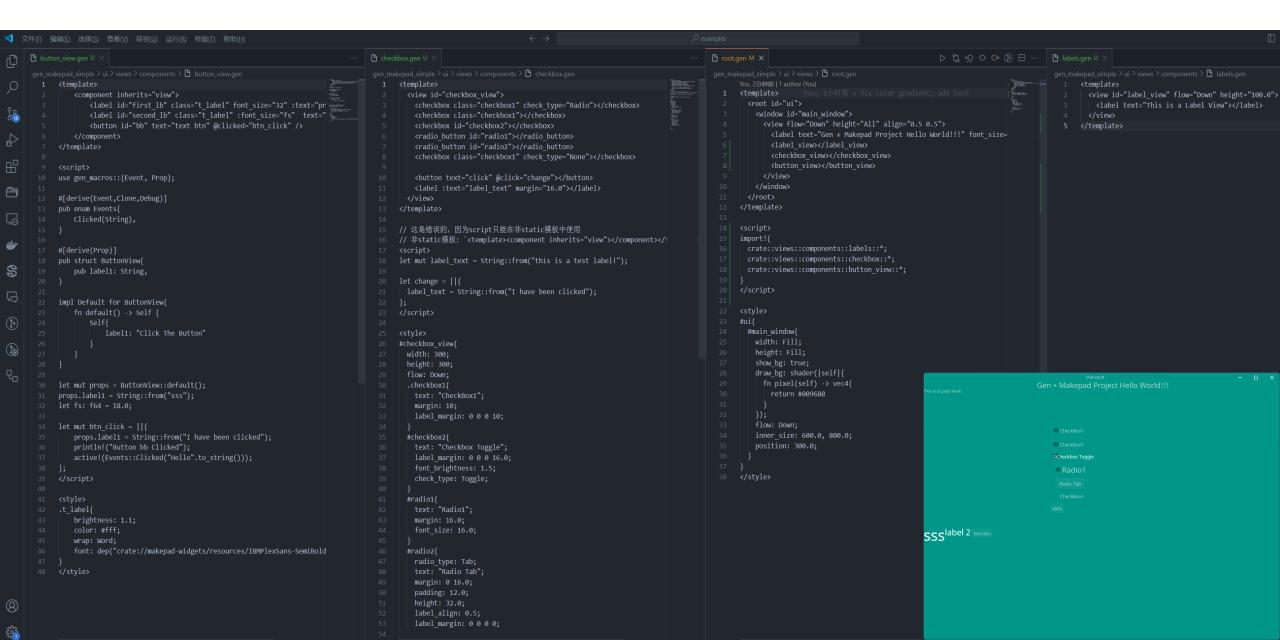


```
Linear
   #main_window{
     draw_bg: linear_gradient(120deg, #FF938A, #98943F 40%, #6BC46D);
```

```
<style>
#ui{
    #main_window{
        //...
        draw_bg: radial_gradient(#FF938A, #98943F, #6BC46D);
    }
}
</style>
```



Other Examples (with Makepad)



Features

- 完成Makepad所有内置Widget的静态和动态解析(完美集成Makepad)
- 添加GenUI内置Makepad Widget
- 动画写法
- 控制 (for, if-else, slot)
- 完成GenUI-Traits和GenUI-Macro使得直接在rs文件中构建Widget (可脱离DSL)
- 优化框架代码
- ...