使用 Rust 开发的视觉小说引擎

王宇逸 <Strawberry_Str@hotmail.com>

2022 年 11 月 5 日

清华大学工程物理系

项目描述

什么是视觉小说

视觉小说是一种电子游戏，是有声读物的衍生产品。被称作 Galgame 的视觉小说分支，通常与二次元亚文化联系紧密，是年轻人非常喜爱的游戏类型之一。

图 1: 《白色相簿 2》(2010)，现实主义爱情 Galgame，其相关研究被称为“白学”

视觉小说的特点

视觉小说以文字、配音、画面、背景音乐四部分为主。传统上玩家通过选择不同分支开启不同的线路，达成不同的结局。

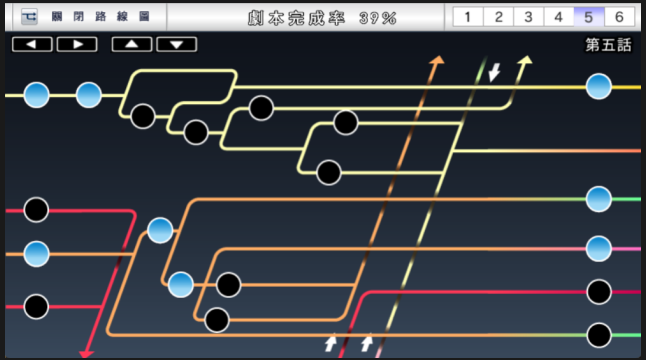

图 2: 《日在校园》(2005) 的路线分支。这是一部校园言情 Galgame，分支复杂结局众多

视觉小说是自由的

视觉小说作为一种内容主导的游戏，玩法可以传统也可以灵活。

图 3: 《高考恋爱 100 天》(2014)，国产 Galgame，玩法灵活，结局多样

内容是视觉小说的基石

视觉小说本质上仍然是内容为主的游戏，文本创作至关重要。

图 4: 《冬滚滚》(2021)，伪装成废萌的科幻 Galgame

我们的目标

市面上有许多视觉小说引擎，各有利弊。我们要开发新的引擎，主要关注以下几个目标：

简单

定义一种简单的、内容导向的视觉小说描述文件，使用简单的语法快速创作。

自由

基础功能精简、模块化，提供高度自由的插件系统。

专注

视觉小说的前端、后端、内容创作应该分离。

项目内容

描述文件

视觉小说描述文件以 YAML 语法为基础。文本内容中有类似 TEX 语法的标记，以及可以嵌入脚本语言。

```yaml
- tag: para1
  title: 段落一
  texts:
    - 这是第一行话
    - 这是第二行话
    - 下面是一个选项
    - switches:
      - 跳到段落二 || $end = false
      - 直接结束   || $end = true
      - 钝角       |  false
    - exec: $next = if(!$end, "para2")
  next: \var{next}
```

```yaml
- tag: para2
  title: 段落二
  texts:
    - 到了第二个段落
```

文本命令

文本中使用的类似 TEX 语法的标记，基础的命令如下：

命令	说明
\ch	指定本行的角色以及别名
\res	通过索引获取对应的本地化资源
\var	获取脚本中对应名称的变量值

其余命令允许插件系统实现。

Ayaka Script

定义了一种脚本语言 Ayaka Script，是一种简单的、动态类型的语言。它只支持元类型~、布尔类型、整数类型（实现为 i64）、字符串类型。

```
pub enum RawValue {
    Unit,
    Bool(bool),
    Num(i64),
    Str(String),
}
```

脚本语言示例

结合段落的分支功能，这个脚本语言已经能够执行一些计算，例如斐波那契数列。

```
- tag: init
  texts:
    - 1
    - exec: $n = 50; $a = 1; $b = 1; $i = 1;
    - \var{b}
  next: loop
- tag: loop
  texts:
    - exec: c = $b; $b += $a; $a = c; $i += 1;
    - \var{b}
    - exec: $next = if($i < $n, "loop")
  next: \var{next}
```

国际化/本地化

视觉小说作为文化产品，是全人类的精神财富。我们定义的描述文件也有国际化支持。资源 res 和段落 paras 都要指定对应的语言，描述文件也要指定基础语言 base_lang。对系统语言的适配，以及选择哪一种语言的资源的算法，利用 ICU (International Components for Unicode) 的 CLDR (Unicode Common Locale Data Repository) 实现 [1]。这一功能已作为单独的 crate 发布在 crates.io，名为 language-matcher。

```rust
let matcher = LanguageMatcher::new();
let accepts = [langid!("en"), langid!("zh-Hans"), langid!("zh-Hant")];
assert_eq!(
    matcher.matches(langid!("zh-CN"), &accepts),
    Some((&langid!("zh-Hans"), 0))
);
```

简化翻译工作量

描述文件的一行可能有许多文本与命令，但是只有文本是需要翻译的。

ja	zh-Hans
- bg: 0	-
- \ch{rd}団長！車の用意できました！	- 团长！车已经准备好了！
- switches:	- switches:
- おう! || $end = false	- 哦!
- 止まるんじゃねぇぞ! || $end = true	- 不要停下来啊!
- 止まれ! | false	- 停下!

已有的 WebAssembly 插件框架

目前已经存在至少三种开源的 WASM 插件框架，各有优劣，对比如下：

	fp-bindgen	wit-bindgen	wasm-bindgen
类型定义	Rust 已有类型 ✔	.wit 描述文件定义	Rust 已有类型 ✔
宿主引擎	Wasmer(Rust) ✔	Wasmtime(Rust) ✔	浏览器
序列化格式	MessagePack ✔	Canonical ABI	JSON
系统调用	自定义集合	WASI ✔	Javascript
反射	-	-	?

反射是我们最需要的功能，但是这几种框架都没有自动化的解决方案。因此我们决定结合它们的优点，自己开发插件框架。

已经开发的插件

插件的目标架构为 wasm32-wasi 或 wasm32-unknown-unknown。WASI 可以为插件提供一些 IO 接口，方便开发与调试。

名称	描述
ayalog	日志
basictex	基础 TEX 命令
live2d	Live2D 命令支持
markdown	处理 Markdown 文本
media	多媒体命令
random	生成随机数

简单的插件开发流程

添加引用 ayaka-bindings，使用提供的宏导出需要的函数。

脚本插件的示例：

```rust
use ayaka_bindings::*;

#[export]
fn plugin_type() -> PluginType {
    PluginType::default()
}

#[export]
fn hello_world(_args: Vec<RawValue>) -> RawValue {
    RawValue::Str("Hello world".to_string())
}
```

完整的文本处理流程

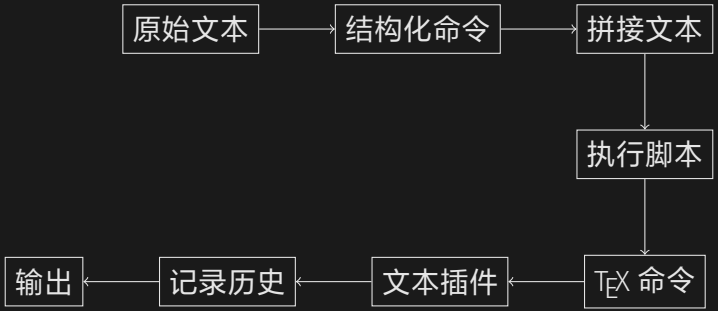

图 5: 文本的处理流程

文本是惰性求值的，目前并没有缓存机制，也没有观察到明显性能损失。

Ayaka Logo

图 6: 咕到很久才做出来的 Logo

GUI 框架选择

我们希望借助成熟的网页前端技术开发 GUI 前端，在这一领域比较流行的框架是 Tauri。在这个框架发布 1.0 不久我就开始了开发。前端技术选择了 Vue，是考虑到技术成熟，文档丰富。前端语言选择了 Typescript，它的类型系统比较现代。

图 7: 游戏界面

多语言显示

最近我们又一次重构了文法，以确保国际化支持与 Ayaka Script 解耦。于是可以添加前端的多语言显示功能。

图 8: 中日双语

Live2D 支持

使用网页前端技术的好处是可以使用 WebGL 获得高级的动画效果。我们最终选用开源社区维护的 `pixi-live2d-display` 做 Live2D 支持。

图 9: 这是 Live2D 官方示例提供的模型 Hiyori

总结

任务列表

已经完成：

- 跨平台的 GUI 程序
- 背景图、背景音乐、播放视频、语音
- Live2D 简单支持
- 历史记录、无限回退
- 自动播放、快进播放
- 存档、读档
- 区分已经播放过与从未播放过的文本

尚未完成：

- 验证 Linux 上的兼容性
- 商业游戏平台的 API 对接
- 前后端编译解耦合
- 脚本服务器？
- 编辑器？
- 实战事例？

总结

- 我们正在开发一个视觉小说引擎，着眼于**简单、自由、专注**。
- 这是一个前后端分离的，使用 Rust 写作的跨平台引擎。
- 以 YAML 为基础的描述语言，以及内嵌的脚本系统。
- 以 ICU CLDR 为基础的国际化支持。
- 使用 WebAssembly 的插件系统，并与脚本语言互操作。
- 我们将继续维护并增加新功能。

本项目的仓库托管在 GitHub 上：Uni-Gal/Ayaka

本项目得到了 OSPP 的支持。