

文档

2017年4月26日 19:05

1. 简介
2. 编译
3. 编写一个echo用例
4. 程序的运行时解析
5. 其他

简介

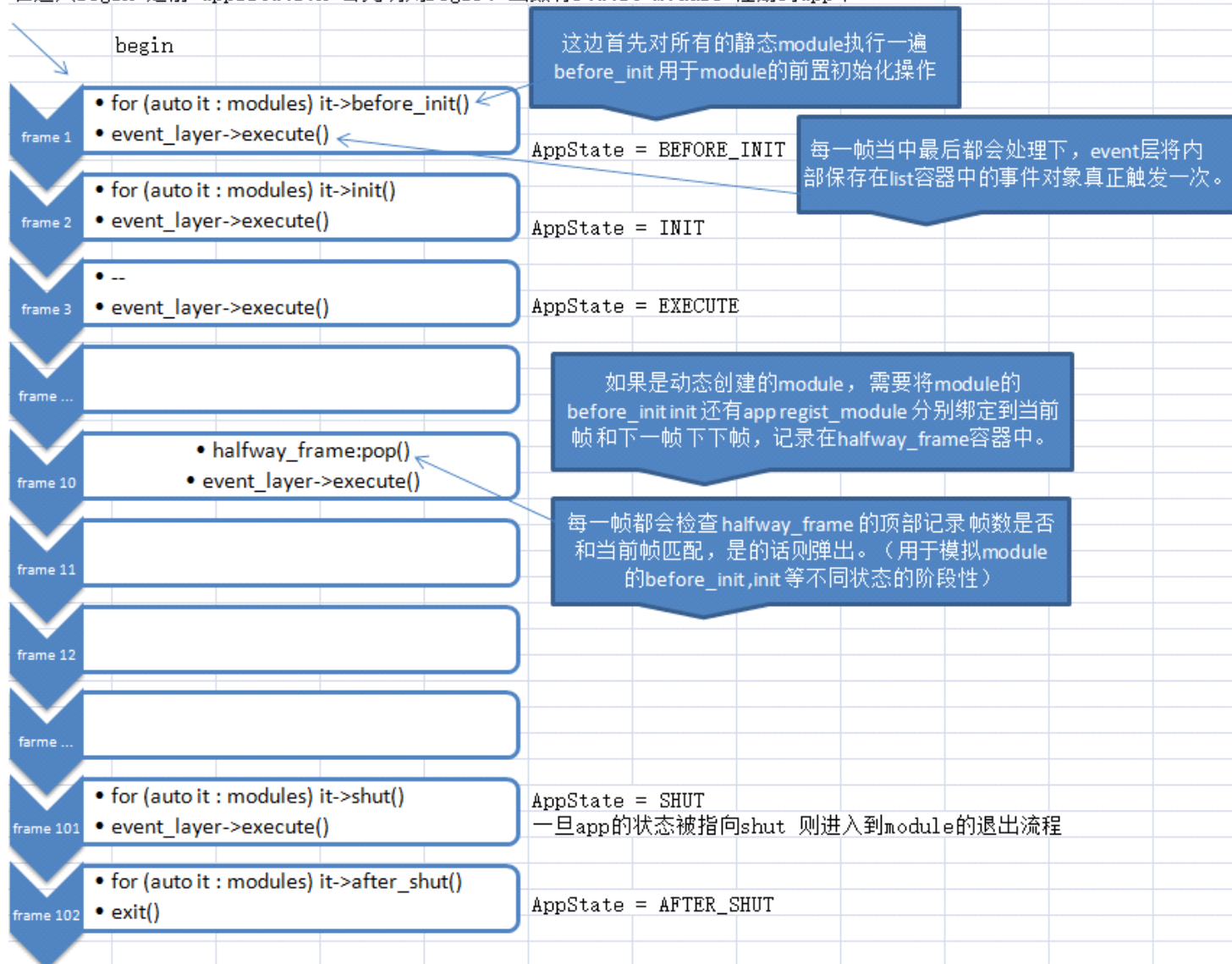
这是一个组件式的游戏服务器框架，其设计意图是帮助团队能够尽可能容易的写出耦合度低的代码。

特性

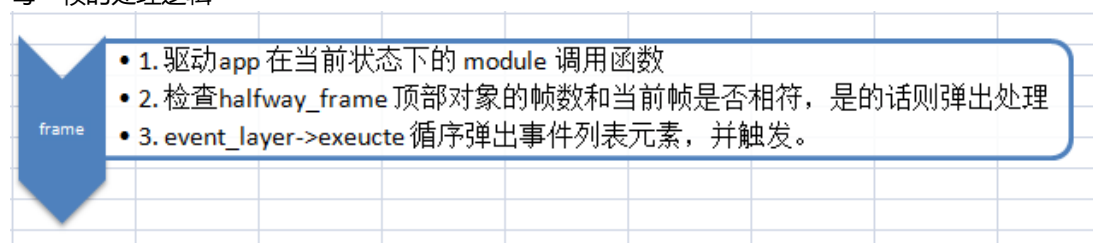
1. 所有的module都完全独立，只能通过event和其他module产生交互行为，由application驱动。
2. 提供args，可以像动态语言一样方便的传参
3. 支持动态的创建和销毁module
4. 支持lua module，让c++和lua 更好的结合，可以通过构建多个lua module来水平扩展逻辑。
5. 支持轻量的集群（待实现）

程序的运行示意

在进入begin 之前 application 会先调用regist 函数将static module 注册到app中



每一帧的处理逻辑



性能分析

hash 容器用于注册，解除 命令和目标函数之间的绑定关系

list 容器用于保存具体的命令对象，在每帧循环弹出处理。
后续会在性能测试阶段 用池封装下命令对象。

可预料的每帧消耗 $list_pop\ O(1) * list_len + * hash_find\ O(1) * list_len$
大约等于 $list_len * 2$ 具体要看后续的性能测试分析



核心接口

- 1. eid::get_module
 - a. 通过module的名称获得module的id，只能用于静态申请的module。
- 2. eid::new_dynamic_module
 - a. 通过module的名称动态创建一个对象（需要在c++中声明好对象类型
- 3. eid::delete_dynamic_module
 - a. 通过module的id在运行时删除这个module

1. event::listen(module_id, event_id, args, func)

a.	module_id	侦听所有往这个module发送的事件，一般使用module自己的id
	event_id	事件id，可以在event_list 中查找
	args	传出的参数列表，需要保证压入和弹出的顺序。
	func(args)	事件处理函数

2. event::dispatch(module_id, event_id, args, callback)

a.	module_id	发往目标的module_id
	event_id	
	args	
	callback	有些发出事件可以被立即处理，会调用callback回调函数

lua层相关

由于整个框架是基于事件的，在lua_script中已经默认完成了事件和lua层相关的绑定，所以在使用lua的过程中只需通过事件id就能进行模块之间的访问，只需在c++层做好创建，重载，销毁 lua_state 等接口。无需各种功能向的绑定

- 1. 使用lua模块
 - a. 在c++层创建一个lua_proxy的module，用于管理lua状态机的创建，重载，销毁等。
 - b. 在lua_proxy中向LuaScriptModule（lua层的绑定器）发送事件创建一个真正的lua状态机。
 - c. 如果这个lua_state module 被用于动态创建的话则需显示声明 REGISTER_CLASS，后通过 new_dynamic_module事件进行自动创建。
 - d. 如果这个lua_state module 是静态的（推荐方式），则只需通过app.regist_module 注册。

```
module = {
  before_init = function() end,
  init = function() end,
  execute = function() end,
```

```
    shut = function() end,  
    after_shut = function() end,  
}
```

lua中module对象的声明方式，其运行机制和c++ module类似。

预设的功能模块（都是由module实现

1. acceptor
2. connector
3. log
4. timer
5. lua_binder
6. dbproxy

可以通过宏WATCH_PERF 开启模块性能监控

```
----- pref -----  
LogModule:0.001500      LuaScriptModule:0.000250      RobotMain:0.000000  
Base_LuaProxy:0.000000  :0.000250      EventModule:0.014625
```