MICROOH 麦可网

Android-从程序员到架构师之路

出品人：Sundy
讲师：高焕堂（台湾）

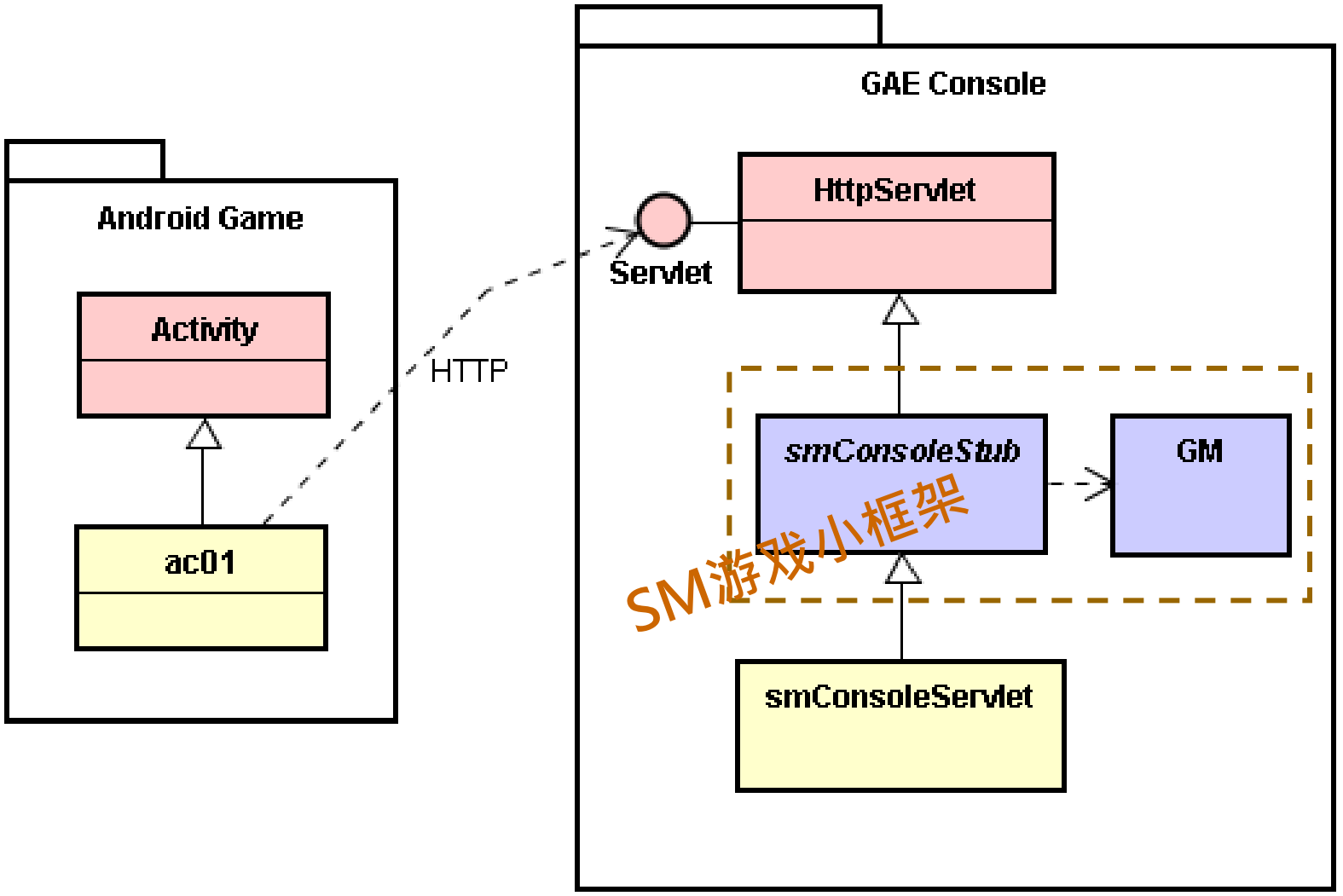
http://www.microoh.com

F05_d

观摩：Android端云整合与分工策略(d)

By 高煥堂

前言


GAE Console
Android Game
HttpServlet
Servlet
Activity
HTTP
smConsoleStub
GM
SM游戏小框架
ac01
smConsoleServlet

这Stub部分包含两个类：
GM类和smConsoleStub类。

Stub范例代码

```
/*---- GM.java ----*/
package GameFramework;
import java.util.Random;
public class GM {
        public String state_var;
        public int current_amount = -1;;
        public int current_rank = -1;
        public Boolean status = null;
        //-----------------------------------------
```

```
public GM()    {  this.go_state_0();  }
public void go_state_0(){  state_var = "0";  }
public void go_connected_state_1(int amt){
             if(! state_var.contains("0")){
                          status = false;
                          return;
             }
            state_var = "1";
            current_amount = amt;
             this.go_state_2();
  }
public void go_state_2(){
               state_var = "2";
               status = true;
  }
```

```
public void go_prizes_state_3(int amt, int bet){
        if(! state_var.contains("2")){
                status = false;    return;    }
        state_var = "3";
         // 計算獎金
         RC obj = new RC();
         current_rank = obj.getRandomInt(0, 1000);
         int prize = current_rank * bet;
         if(prize > 0) amt += bet;
         current_amount = amt + prize;
         this.go_state_2();
    }
public void go_finished_state_4(){
         if(! state_var.contains("2")){
                status = false;  return;  }
         state_var = "4";
         this.go_state_2();
    }
```

```
public class RC{
    public int getRandomInt(int min,int max) {
        try { Thread.sleep(2);
        } catch (InterruptedException ex)  {  ex.printStackTrace();  }
        Random randomizer =
                new Random(System.currentTimeMillis());
        int k = randomizer.nextInt(max-min+1)+min;
        if(k < 500) return 0;
            if(k<750) return 1;
            if(k<875) return 2;
            if(k<938) return 3;
            if(k<969) return 4;
            if(k<984) return 5;
            if(k<994) return 6;
            if(k<1000) return 7;
            return 0;  }
}}
```

- 此RC类是依据随机值(Random)而换算出获奖的奖项(Rank)，其实各家游戏场都有不一样的奖项决定规则，而且随时都可能更换新的奖项规则。上述RC类只是一个简单范例而已。

```
/*---- smConsoleStub.java ----*/
// ........
public abstract class smConsoleStub extends HttpServlet {
    private GM gm = null;
    private User user = null;
    private String strResult;

    protected void doGet( HttpServletRequest req,
                          HttpServletResponse resp)
        throws ServletException, IOException {
            UserService userService =
                    UserServiceFactory.getUserService();
            user = userService.getCurrentUser();
            gm = new GM();
            String gm_state = null;
            HttpSession session = req.getSession();
```

```
        Object obj = session.getAttribute("gmState");
        if(obj != null ){
                    gm_state = (String)obj;
                    gm.state_var = gm_state;  }
        String strCode = req.getParameter("code");
        String sv1 = req.getParameter("value1");
        String sv2 = req.getParameter("value2");
        int code = Integer.valueOf(strCode);
        if(code == 0) {
             process(sv1);
             if(gm.status == false) strResult = "Fail:99,99";  }
    else       strResult = Test(sv1, sv2);
    session.setAttribute("gmState", gm.state_var);
    resp.setContentType("text/plain");
            resp.getWriter().println(strResult);
}
```

```
private void process(String msg){
    char cmd = msg.charAt(0);
    if(cmd == 'I'){
        gm.go_state_0();
        gm.go_connected_state_1(onInitialAmount(user));
        String strAmt = String.valueOf(gm.current_amount);
        strResult = "Init:" + strAmt + ",99";
     }
    if(cmd == 'B'){
        int idx = msg.indexOf(",");
       String str_a = msg.substring(5, idx);
       String str_b = msg.substring(idx+1);
       int amt = Integer.parseInt(str_a);
       int bet = Integer.parseInt(str_b);
       gm.go_prizes_state_3(amt, bet);
       String strAmt = String.valueOf(gm.current_amount);
       strResult = "Bett:" + strAmt + ","
                                + String.valueOf(gm.current_rank);
    }
```

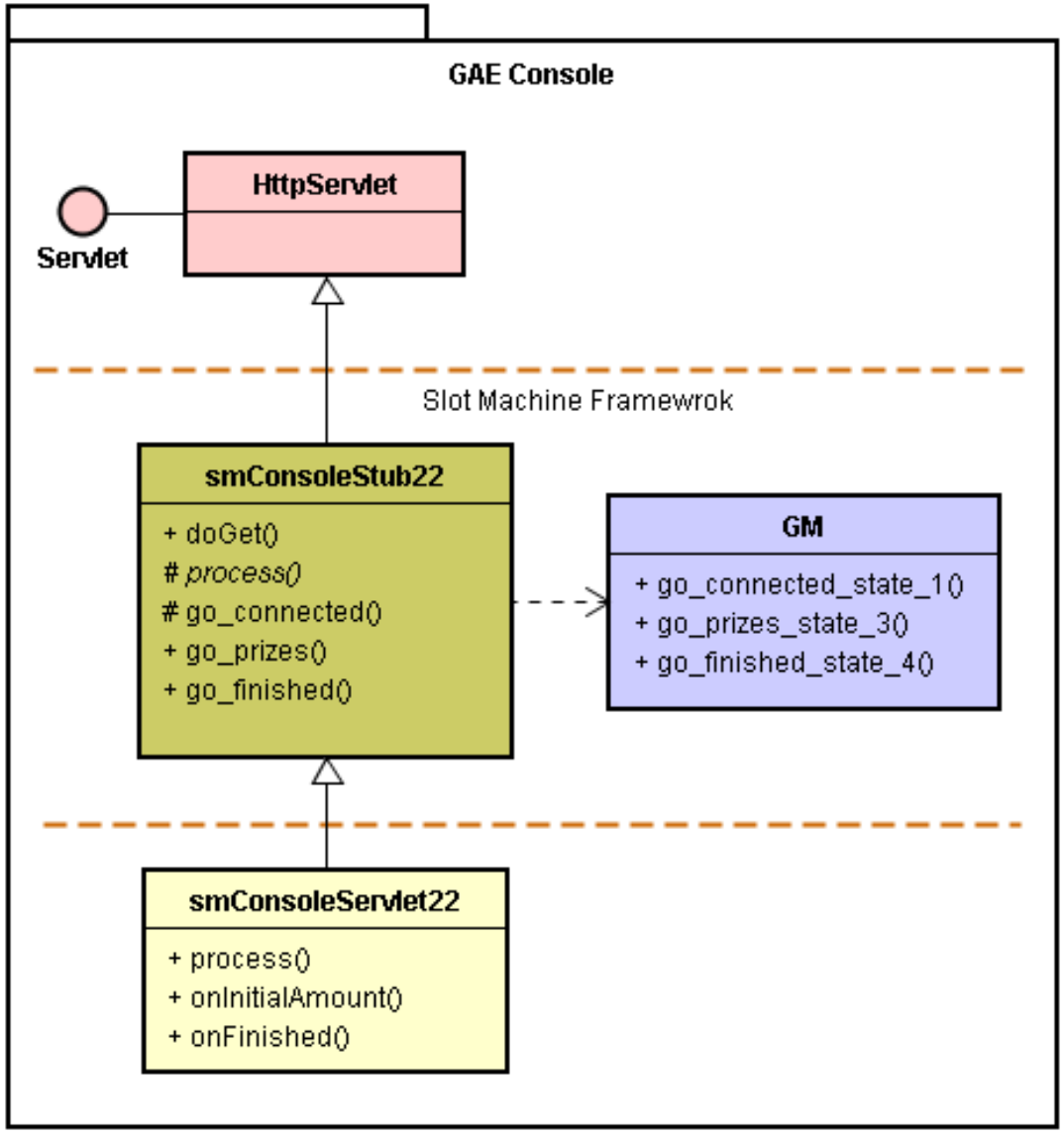
```
        if(cmd == 'F'){
            int idx = msg.indexOf(",");
            String str_amt = msg.substring(5, idx);
            int amt = Integer.parseInt(str_amt);
            boolean ret = onFinished(user, amt);
            if(ret){
                gm.go_finished_state_4();
                strResult = "Fini:99,99";  }
            else  strResult = "Fail:99,99";
    }}
private String Test(String sv1, String sv2) {  return "***";   }
abstract protected int onInitialAmount(User user);
abstract protected boolean onFinished( User user,
                                       int final_amount);
}
```

- Android游戏机端传送HTTP讯息给GAE云层，就转而調用上述的doGet()函数。此时诞生一个GM对象，并从session取得"gmState"的值，并将此值存入GM对象里，设定了GM对象的状态值。

- 接着，转而調用process()函数来解析讯息内容，在依据内容而調用GM对象的函数或应用子类的函数，最后回传讯息给游戏机端。

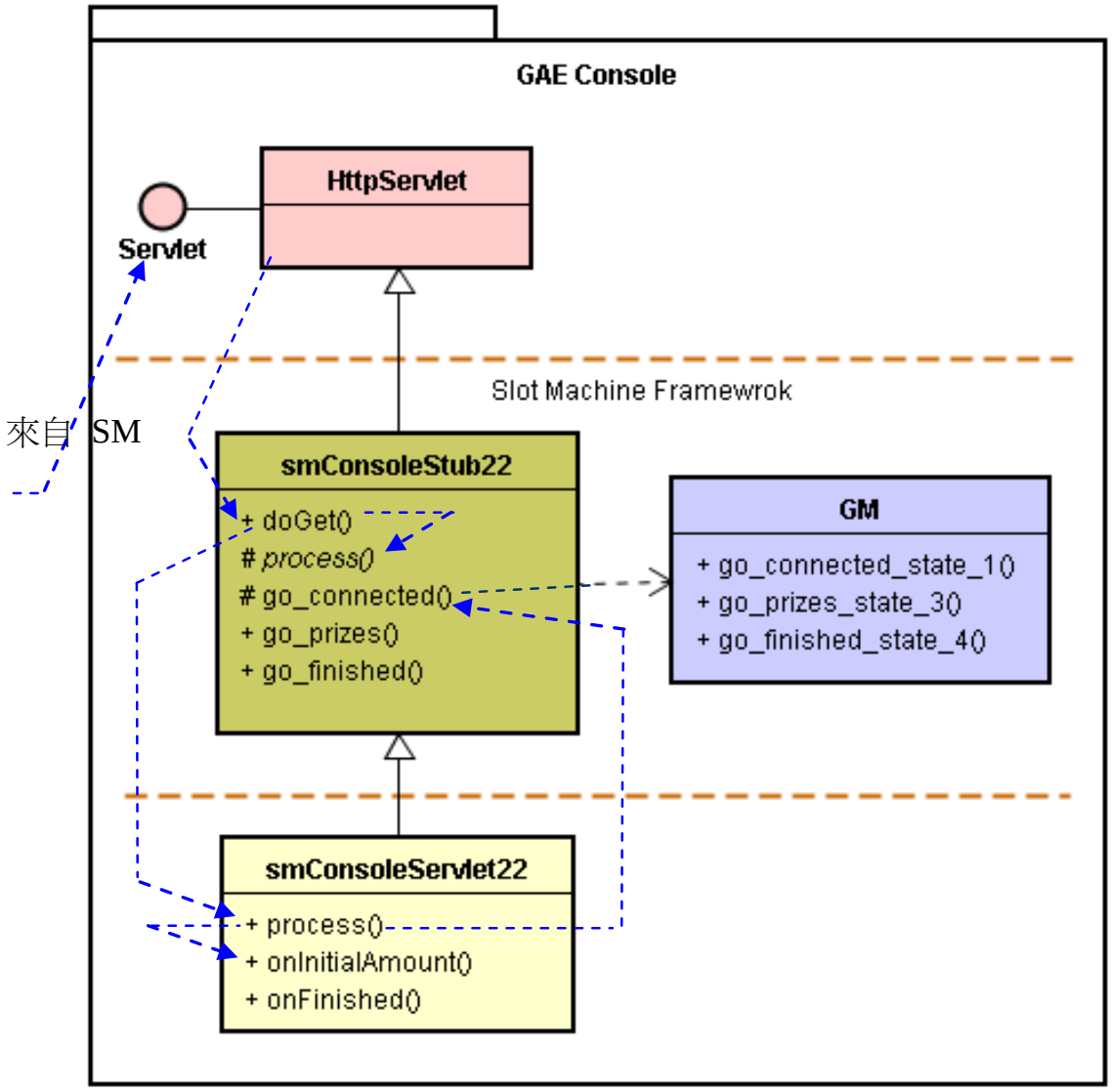
架构师的决策：
选择另一种Stub设计方案

- 在上述方案一里，Stub设计师决定了Android游戏端与云层之间沟通讯息的格式，而App开发者遵循之，而不能制定自己喜欢的讯息格式。
- 如果想让App开发者能自行决定上述的讯息格式，就可更改框架设计如下图：


GAE Console
HttpServlet
Servlet
Slot Machine Framewrok
smConsoleStub22
+ doGet()
# process()
# go_connected()
+ go_prizes()
+ go_finished()
GM
+ go_connected_state_1()
+ go_prizes_state_3()
+ go_finished_state_4()
smConsoleServlet22
+ process()
+ onInitialAmount()
+ onFinished()

- 在此新方案里，smConsoleStub22类别的process()是抽象函数，让App的smConsoleServlet22子类来实作之。
- smConsoleStub22类别只是将讯息转达给App子类smConsoleServlet22而已，并不决定讯息格式，也不解析讯息。

- 而是由smConsoleServlet22子类的process()函数来解析讯息。
- 由于Android游戏端的ac01类和GAE云的smConsoleServlet22子类都属于App，由ac01类与smConsoleServlet22子类之间的沟通讯息格式，是App开发者可以自订了。

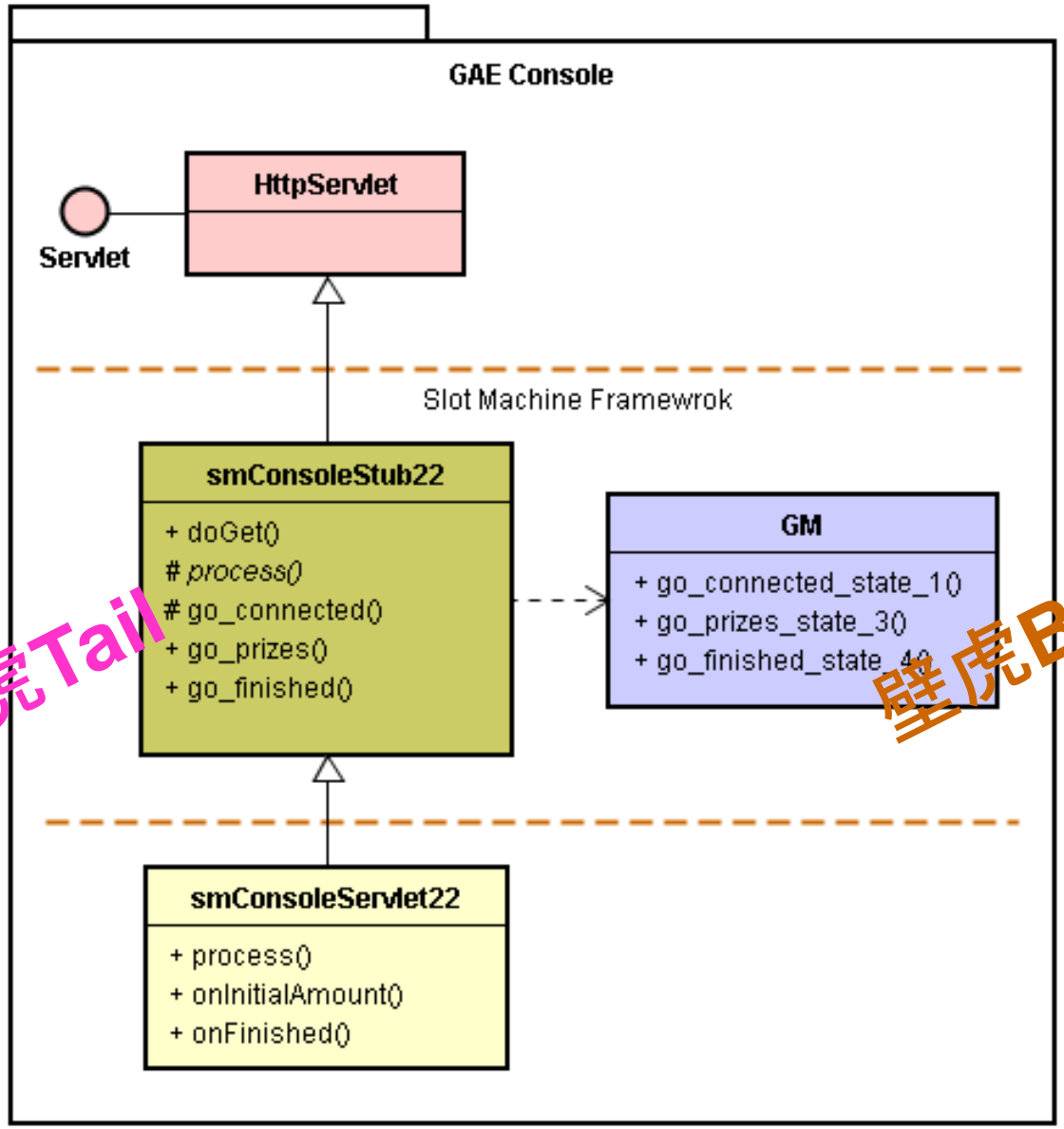

GAE Console
HttpServlet
Servlet
Slot Machine Framewrok
來自 SM
smConsoleStub22
+ doGet()
# process()
# go_connected()
+ go_prizes()
+ go_finished()
GM
+ go_connected_state_1()
+ go_prizes_state_3()
+ go_finished_state_4()
smConsoleServlet22
+ process()
+ onInitialAmount()
+ onFinished()

- Android游戏机端传送HTTP讯息给GAE云层，就转而呼叫上述的doGet()函数。此时诞生一个GM对象，并从session取得“gmState”的值，并将此值存入GM对象里，设定了GM对象的状态值。
- 接着，转而呼叫process()函数来解析讯息内容。

- process()函数解析到Android游戏端传来“Init:”讯息时，就先呼叫smConsoleStub22的onInitialAmount()去DB里读取玩家的余额，然后呼叫父类smConsoleStub22的go_connected()函数，转而呼叫GM的go_connected_state_1()而将余额传送给GM。
- 随后，将余额回传给Android游戏端，显示于画面上。

结语

- 以上Stub小框架里的GM类是<壁虎Body>，而smConsoleStub类是<壁虎Tail>；可以展现弃尾求生的效果，所以是一种跨(云)平台的架构设计模式。


GAE Console
HttpServlet
Servlet
Slot Machine Framewrok
smConsoleStub22
+ doGet()
# process()
# go_connected()
+ go_prizes()
+ go_finished()
GM
+ go_connected_state_1()
+ go_prizes_state_3()
+ go_finished_state_4()
smConsoleServlet22
+ process()
+ onInitialAmount()
+ onFinished()
壁虎Tail
壁虎Body

Thanks...

高煥堂