插件说明

每个插件一个文件夹，都放在插件放置目录内

插件放置目录（和VSCode插件同级）

用户目录/.laya/plugins/

编辑器使用的插件配置文件

plugin.cfg

例子：

{

"plugins":

[

"testplugin",

"testas3plugin"

]

}

以上的例子，表示启用了testplugin 和testas3plugin两个目录下的插件

支持两种类型的插件

Panel类型的插件指可以参与面板布局的插件，类似于属性面板，资源面板

Dialog类型的插件指弹出框类型的插件，类似于项目设置弹出框

支持两种语言开发

支持使用js语言和as3语言进行插件开发

下面分别讲解使用js和as3分别开发Panel类型插件和Dialog插件的实例

Js插件实例

Testplugin插件为纯js写的插件

Testplugin目录下包含以下三个文件：

ui.div

config.cfg

code.js

ui.div为插件的ui布局，为普通的html标签

<div class="main" style="height:100%;width:100%;" >

<div name="txt" class="content" style="background-color: #ffff00; color: #00ff00; height:100%;width:100%;"><p>我是内容</p></div>

</div>

里面写了几个div标签，并定义了class name等信息便于在code.js里通过代码获取对应的对象

config.cfg为该插件的配置文件

{

"type":"js",//表示是js写的程序

"menus"://注册菜单

[

{

"title":"打开JSPlugin",//菜单名

"notice":"openJSplugin"//点击菜单发送的消息

},

{

"title":"关闭JSPlugin",

"notice":"closeJSplugin"

},

{

"title":"打开JSDialog",

"notice":"openJSDialog"

},

{

"title":"关闭JSDialog",

"notice":"closeJSDialog"

}

],

"codes"://包含的插件类

[

{

"code":"code.js",//加载的js文件

"ui":"ui.div", //使用的ui文件

"class":"plugin.Plugin",//插件类全路径

"isPanel":true,//是否是Panel

"title":"JSPlugin"//插件标题

},

{

"code":"code.js",//加载的js文件

"ui":"ui.div", //使用的ui文件

"class":"plugin.PluginDialog",//插件类全路径

"isDialog":true,//是否是Dialog

"title":"JSPlugin"//插件标题

}

]

}

code.js为可执行的js代码，里面为插件逻辑代码

//面板型的插件

function Plugin()

{

var ui;//ui对象

var panel;//panel对象

//在创建时ide会将插件的ui实例传进来

this.setUI=function(panel)

{

console.log("setUI",panel)

this.panel=panel;

this.ui=panel.div;

this.txt=this.ui.getElementsByClassName("content")[0]

console.log("setUI",this.ui,this.txt)

}

this.init=function()

{

//监听消息中心派发的消息

ide.managers.Notice.listen(laya.ide.event.IDEEvent.PROPPANEL\_SETBYOBJ, this, this.initByObj);

ide.managers.Notice.listen("openJSplugin", this, this.open);

ide.managers.Notice.listen("closeJSplugin", this, this.close);

}

this.open=function()

{

if(this.panel) this.panel.show();

}

this.close=function()

{

if(this.panel) this.panel.close();

}

this.initByObj=function(data,type,id)

{

console.log("initByObj:",data,type,id)

//显示获取到的数据

var tKey;

var tStr="";

var lines=[]

for(tKey in data)

{

lines.push(tKey+":"+data[tKey])

}

tStr=lines.join("\n");

this.txt.innerText=tStr;

}

this.init();

}

//对话框型的插件

function JSPluginDialog()

{

var ui;

var panel;

this.setUI=function(panel)

{

console.log("setUI",panel)

this.panel=panel;

this.ui=panel.div;

this.txt=this.ui.getElementsByClassName("content")[0]

console.log("setUI",this.ui,this.txt)

}

this.init=function()

{

this.width=300;

this.height=200;

ide.managers.Notice.listen("openJSDialog", this, this.open);

ide.managers.Notice.listen("closeJSDialog", this, this.close);

}

this.open=function()

{

if(this.panel) this.panel.popup();

}

this.close=function()

{

if(this.panel) this.panel.close();

}

this.init();

}

//创建插件的包，并把插件放到包内

plugin={}

plugin.Plugin=Plugin;

plugin.PluginDialog=JSPluginDialog;

AS3插件实例

testas3plugin为as3开发的插件

testas3plugin目录下包含以下两个个文件：

config.cfg

code.js

config.cfg为插件配置文件

{

"type":"as3",//as3写的插件

"menus"://注册菜单

[

{

"title":"打开AS3Plugin",//菜单名

"notice":"openas3plugin"//点击菜单发送的消息

},

{

"title":"关闭AS3Plugin",

"notice":"closeas3plugin"

},

{

"title":"打开AS3Dialog",

"notice":"openas3Dialog"

},

{

"title":"关闭AS3Dialogn",

"notice":"closeas3Dialog"

}

],

"codes"://包含的插件类

[

{

"code":"code.js",//加载的js文件

"class":"laya.ide.plugin.as3plugin.TestPlugin",//插件类全路径

"isPanel":true,//是否是Panel

"title":"ASPlugin"//插件标题

},

{

"code":"code.js",//加载的js文件

"class":"laya.ide.plugin.as3plugin.TestPluginDialog",//插件类全路径

"isDialog":true,//是否是Dialog

"title":"ASDialog"//插件标题

}

]

}

code.js为as3插件项目发布生成的代码

创建as3插件项目

使用as3开发插件项目与普通的LayaAir项目区别不大，只需要引用提供的IDE接口包就可以了，另外还需要写一个module.def文件，这个文件的作用是将项目代码和库代码分开发布

下面以testas3plugin项目为例

module.def

//将path设为引用的任意一个库即可

mainmodule:"laya.core"

path:"../../../../laya/libs/LayaAir/core/src"

//将path设为插件包

module:"code"

path:"src/laya/ide/plugin"

TestPlugin为Panel类型的插件,只需要继承PluginPanel即可

TestPlugin.as

package laya.ide.plugin.as3plugin

{

import ide.managers.Notice;

import laya.display.Sprite;

import laya.display.Text;

import laya.ide.event.IDEEvent;

import laya.ide.plugin.PluginPanel;

import laya.ui.Box;

import laya.ui.Label;

/\*\*

\* 面板式插件Demo

\* 本demo演示如何像属性面板一样显示编辑器当前选中对象的数据信息

\*/

public class TestPlugin extends PluginPanel

{

public function TestPlugin()

{

creatUI();

init();

}

public var text:Label;

/\*\*

\* 创建UI

\*/

private function creatUI():void

{

text = new Label();

text.color = "#ff0000";

addChild(text);

text.top = text.bottom = text.left = text.right = 0;

}

/\*\*

\* 初始化时创建消息监听

\*/

public function init():void

{

//监听设置属性面板数据的消息

Notice.listen(IDEEvent.PROPPANEL\_SETBYOBJ, this, initByObj);

//监听注册的菜单消息

Notice.listen("openas3plugin", this, this.show);

Notice.listen("closeas3plugin", this, this.close);

}

/\*\*

\* 显示数据

\*/

public function initByObj(data:Object, name:String, id:int):void

{

var tKey:String;

var tStr:String = "";

var lines:Array = [];

for (tKey in data)

{

lines.push(tKey + ":" + data[tKey]);

}

tStr = lines.join("\n");

this.text.text= tStr;

}

}

}

TestPluginDialog为Panel类型的插件,只需要继承PluginDialog即可

TestPluginDialog.as

package laya.ide.plugin.as3plugin

{

import ide.managers.Notice;

import laya.events.Event;

import laya.ide.plugin.PluginDialog;

import laya.ui.Box;

import laya.ui.Button;

import laya.ui.Label;

/\*\*

\* 弹出框式插件Demo

\* 本demo演示如何创建一个自己的对话框

\*/

public class TestPluginDialog extends PluginDialog

{

public function TestPluginDialog()

{

creatUI();

init();

}

public var text:Label;

/\*\*

\* 创建UI

\*/

private function creatUI():void

{

var box:Box;

box = new Box();

text = new Label();

text.color = "#ff0000";

box.addChild(text);

box.size(300, 300);

text.top = text.bottom = text.left = text.right = 0;

text.text = "hello plugin Dialog";

var btn:Button;

btn = new Button();

btn.skin = "comp/button.png";

box.addChild(btn);

btn.right = 20;

btn.bottom = 5;

btn.label = "OK";

btn.on(Event.MOUSE\_DOWN, this, btnClick);

setContent(box);

}

/\*\*

\* 点击关闭

\*/

private function btnClick():void

{

this.close();

}

/\*\*

\* 初始化时创建消息监听

\*/

public function init():void

{

//监听注册的菜单消息

Notice.listen("openas3Dialog", this, this.popup);

Notice.listen("closeas3Dialog", this, this.close);

}

}

}