

# omo课程对接

## 加载cdn设置

```
private async loadResource() {
  try {
    await RES.loadConfig("default.res.json", window.__math2_res_config__ || "resource/");
    await RES.loadGroup("loading");
    this.initFont();
    const loadingView = new LoadingUI();
    this.stage.addChild(loadingView);
    await this.loadMusic();
    await this.loadTheme();
    await RES.loadGroup("preload", 0, loadingView);
    this.stage.removeChild(loadingView);
    window.platform.sendMessage(10001, "G1SummerCU1L161", -1);
  } catch (e) {
    console.error(e);
  }
}

private initFont() {
  let fonts = ["roman"];
  for (let i = 0; i < fonts.length; i++) {
    let label = new eui.Label("aaaaaaaaaaaaa");
    label.fontFamily = fonts[i];
    this.addChild(label);
  }
}

private async loadMusic() {
  let musicSourceArr = RES.getGroupByName("music");
  for (let i = 0; i < musicSourceArr.length; i++) {
    await RES.getResAsync(musicSourceArr[i].name);
  }
}
```

使用RES模块加载，修改为 `RES.loadConfig("default.res.json", window.__math2_res_config__ || "resource/");`

所有资源使用同步加载，全部通过RES模块进行加载。

## 数据通信

```

8      getUserInfo(): Promise<any>;
9
10     login(): Promise<any>
11     getGameInfo();
12
13     sendMessage(code, info, isRight);
14 }
15
16 declare function ____paopaoEgretTextbookMessageHandler__(message: Object);
17
18 You, a few seconds ago | 1 author (You)
19 class DebugPlatform implements Platform {
20     async getUserInfo() {
21         return { nickName: "username" }
22     }
23     async login() {
24
25     }
26     getGameInfo() {
27         return { name: document.title }
28     }
29     sendMessage(code, info, isRight) {
30         if (!GameDevice.isAI && !GameDevice.isOmo) return;
31
32         console.log("sendToNative", isRight);
33         console.log(`${code} sendToNative \ninfo:${info} \nisRight:${isRight}`);
34         egret.ExternalInterface.call("sendToNative", JSON.stringify({
35             //10001 开始 10002答题
36             code: code,
37             msg: "success",
38             data: {
39                 appId: null,
40                 appName: null,
41                 userId: null,
42                 studentId: null,
43                 studentName: null,
44                 gameId: null,
45                 quesId: null,
46                 answer: info,
47                 isRight: isRight, //【0: 错误, 1: 正确, 2: 未作答】
48                 answerTime: 0
49             }
50         })));
51
52         if (typeof ____paopaoEgretTextbookMessageHandler__ != "undefined")
53             ____paopaoEgretTextbookMessageHandler__(
54                 //code : 10001 载入游戏完成 10002答题 10003透传

```

[Platform.ts](#) 主要使用Platform的替换，通过sendMessage进行数据派发。

加载完成 window.platform.sendMessage(10001, "", -1);

作答部分

window.platform.sendMessage(10002, "", 0); 失败

window.platform.sendMessage(10002, "", 1); 成功

window.platform.sendMessage(10002, "", 2); //未作答

window.platform.sendMessage(10002, "", 3); //作答完成，游戏结束

## 焦点处理及暂停管理

```

100, 2 months ago | 2 authors (100 and Others)
9 class Main extends eui.UILayer {
10
11
12     protected createChildren(): void {
13         super.createChildren();
14
15         egret.lifecycle.addLifecycleListener((context) => {
16             // custom lifecycle plugin
17         })
18         XDFEventManager.addEventListener(XDFEventManager.PAUSE, () => {
19             console.log("PAUSE")
20             egret.ticker.pause();
21             soundController.muteBG();
22             soundController.muteEffect();
23         }, this);
24         XDFEventManager.addEventListener(XDFEventManager.RESUME, () => {
25             console.log("RESUME")
26             egret.ticker.resume();
27             soundController.restoreBGVolume();
28             soundController.restoreEffectVolume();
29         }, this);
30
31         //inject the custom material parser
32         //注入自定义的素材解析器
33         let assetAdapter = new AssetAdapter();
34         egret.registerImplementation("eui.IAssetAdapter", assetAdapter);
35         egret.registerImplementation("eui.IThemeAdapter", new ThemeAdapter());
36         this.runGame().catch(e => {
37             console.log(e);
38         })
39     }
40 }

```

删除egret自带的对焦点的处理，加入暂停事件的监听，内部主要暂停音频的播放及游戏的进行。

## 发布

[config.ts](#) [myplugin.ts](#) [index.html](#) [GameDevice.ts](#)

总共包含3个文件，config.ts中，主要定义了如何发布AI版本，即egret publish --version XXXXXXXX，若XXXXXXX中含有"**AI**"字段就会生成AI版本的，通过myplugin.ts的复制操作将index.html复制到最终项目中，与原有相比，index.html多了 `window["isAI"] = true` 字段，游戏内部再通过GameDevice.isShowReturn和isShowRefresh判断是否显示返回和刷新按钮。