



Give Life to 2D

Live2D 技术实践



Live2D 是什么？



Live2D();
</>



Live2D 是一个将平面图形进行动画化的技术方案.

通常使用在与动漫（ACG）相关的产品中.

Live2D 的优点：

- 效果很不错！(◡‿◡) ✧
- 开发很迅速！
- 不需要制作 3D 模型！
- 不需要逐帧绘制动画！
- 可以处理任意平面图形！
- 只需要分层的图片素材就好了！
- 死宅很喜欢！

此类技术目前通常用在 …



游戏

* 此游戏实现非 Live2D，为 E-mote



个人娱乐



其他任何需要的地方

Live2D 怎么玩？

粗略的 Live2D Workflow

一. 画素材

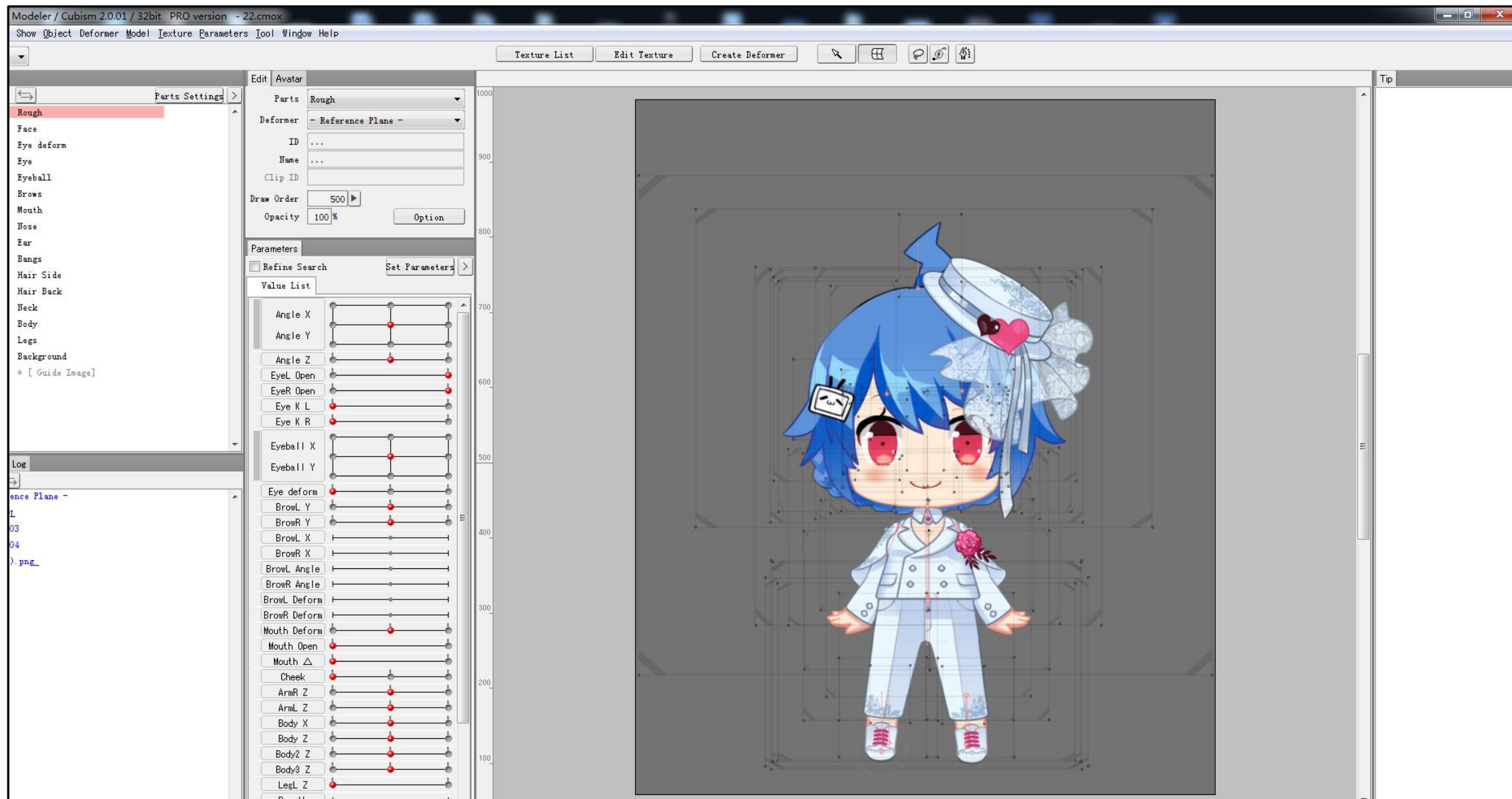
萌萌的素材是 Live2D 坚实的基础 (◡‿◡)^و

设计师需要绘制模型的 T-Pose 待机动作的贴图



然后将文件导入至 [Live2D Modeler](#) 中.

这是 Live2D 模型的配套制作软件.



二. 摆动作

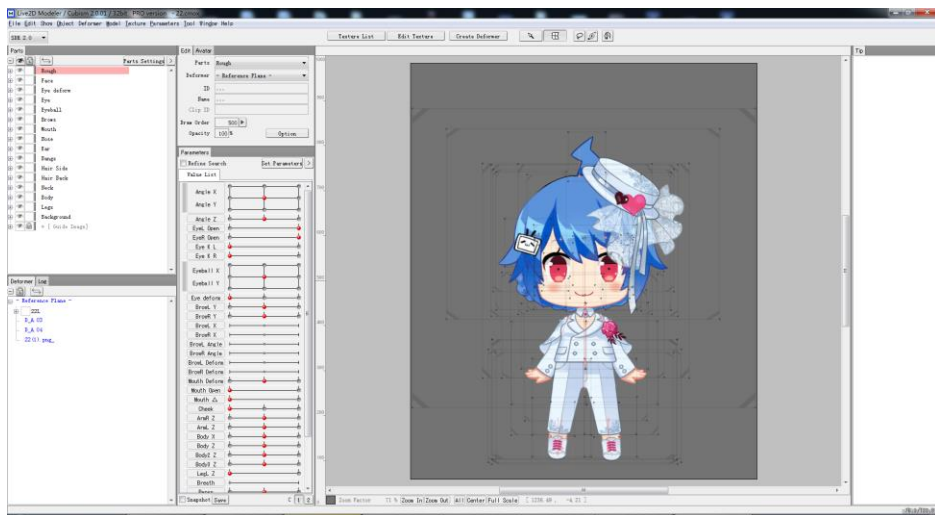
潇洒可爱的动作是 Live2D 的生命力  

Live2D 丰富的动作将在此阶段制作.

在刚刚导入贴图的软件 [Live2D Modeler](#) 中根据贴图建模。

此过程中将生成：

- 模型文件
- 动作文件
- 贴图
- 其他素材（音频等）

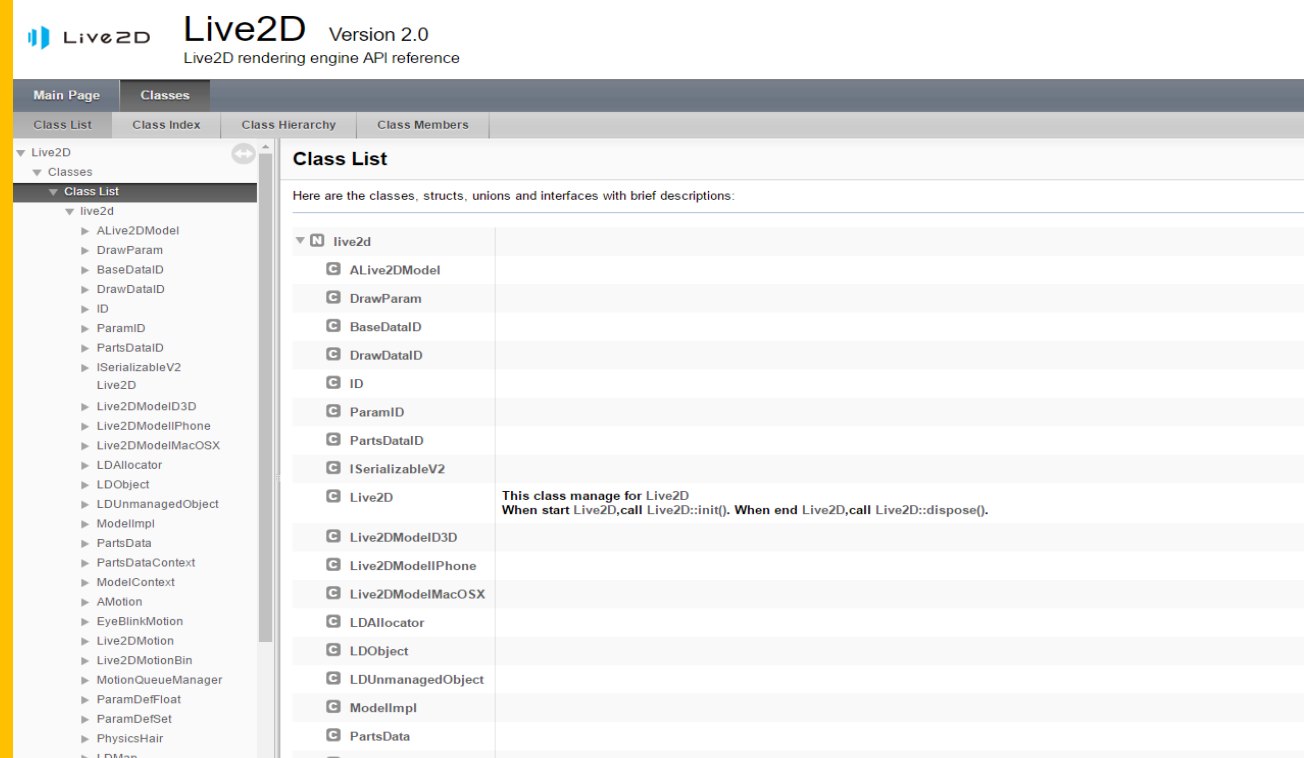


关于软件和其使用方法可以从官网获取…

三. Live2D();

[illegible]

旧版文档：
<http://doc.live2d.com/api/core/cpp2.0e>



Live2D @ GitHub

This page contains downloads and info for obtaining the new Cubism 3.0 SDKs For Unity and Native Development.

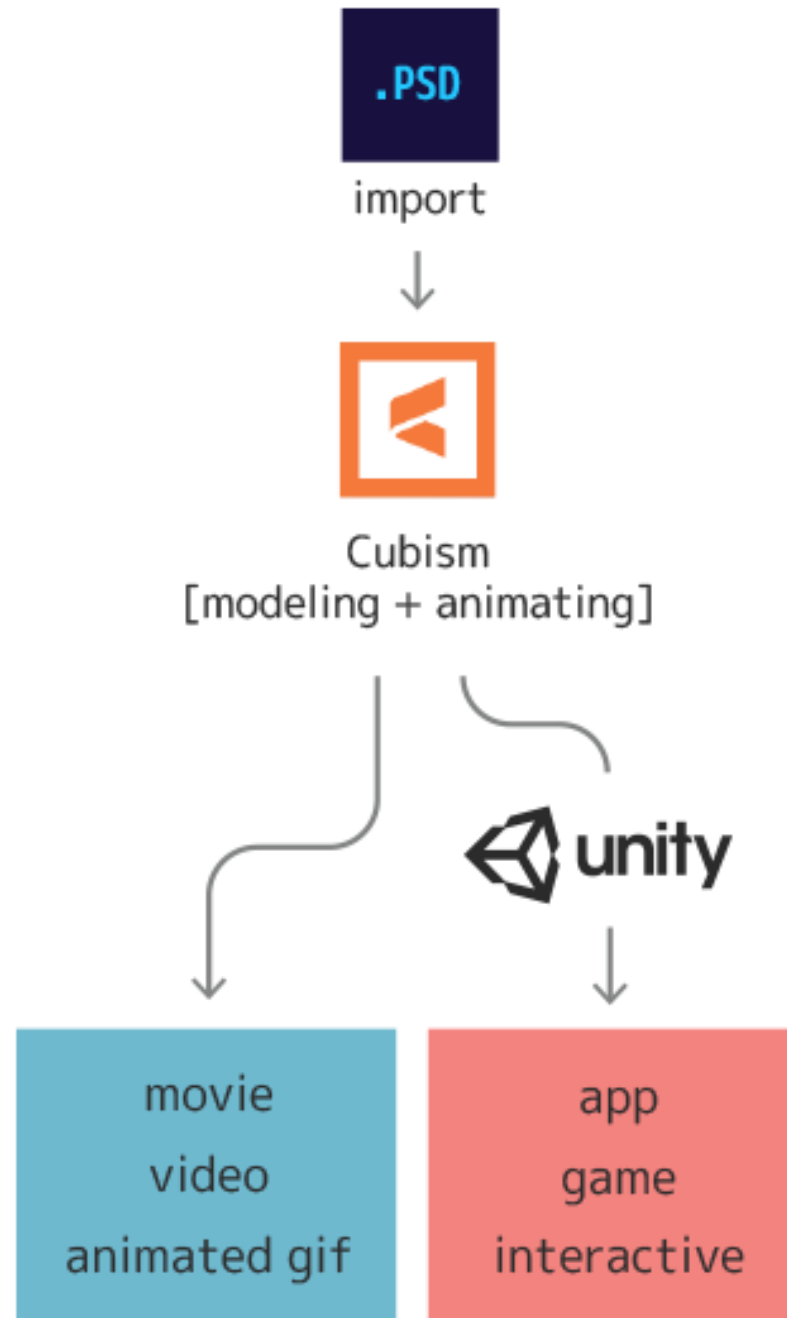
- [SDK For Unity](#)
- [SDK For Native Development](#)



新版文档（还在施工）：<https://live2d.github.io/>

总结一下 Workflow:

Live2D Workflow.



所以一个项目中会需要……



阿宅产品经理



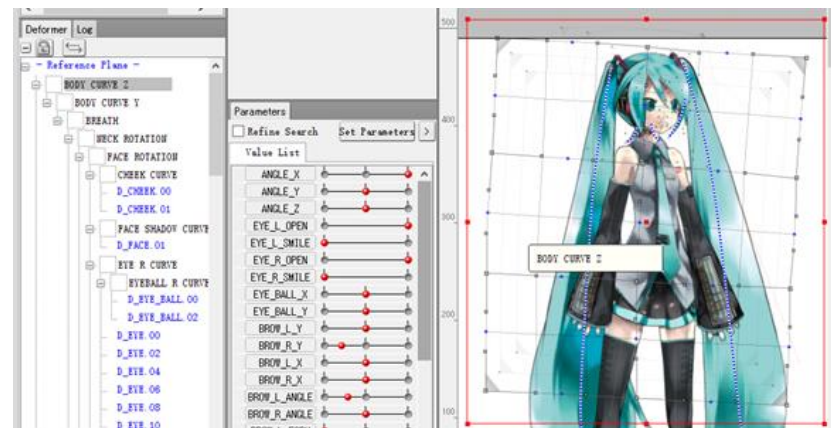
阿宅设计师



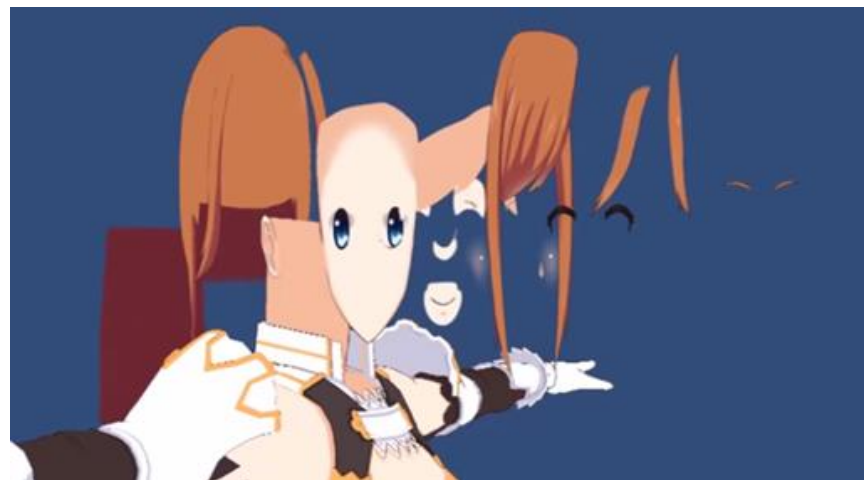
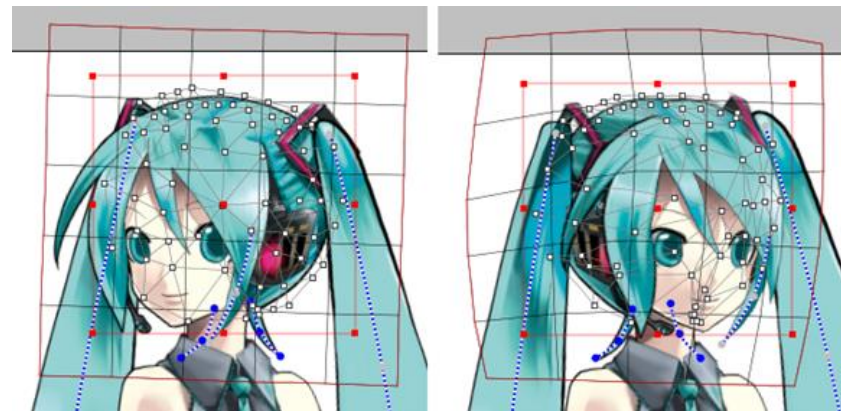
阿宅程序员

Live2D 的大致原理

简单概括一下的话是……



旋转， 矩阵变换， 视差滚动…



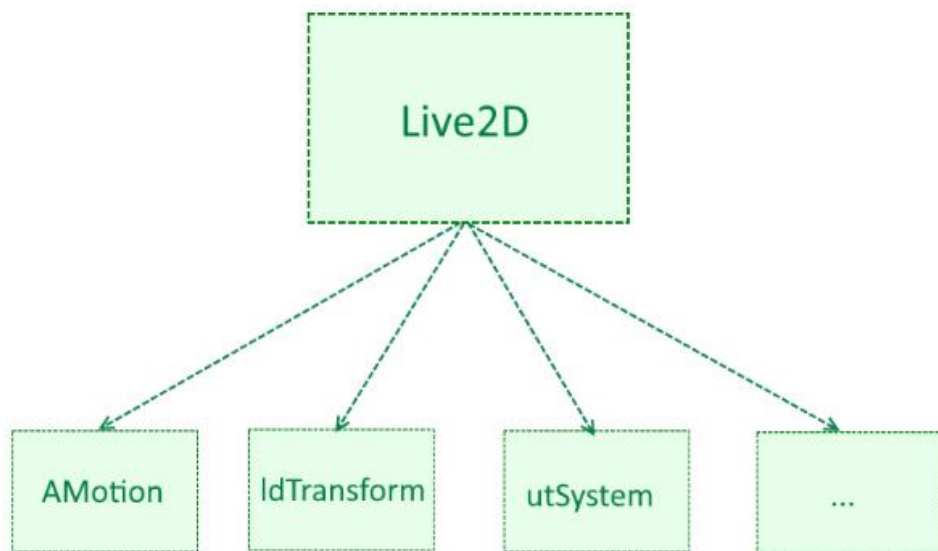
Live2D Web 端开发实践

好的效果也是开发出来的

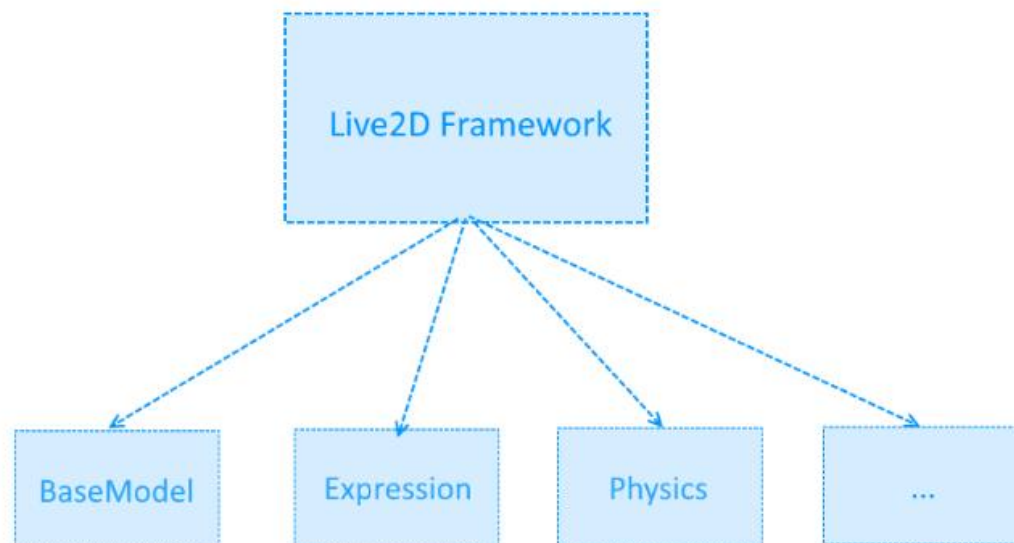


一. Live2D 文件结构

1. Live2D Web 端 SDK 结构

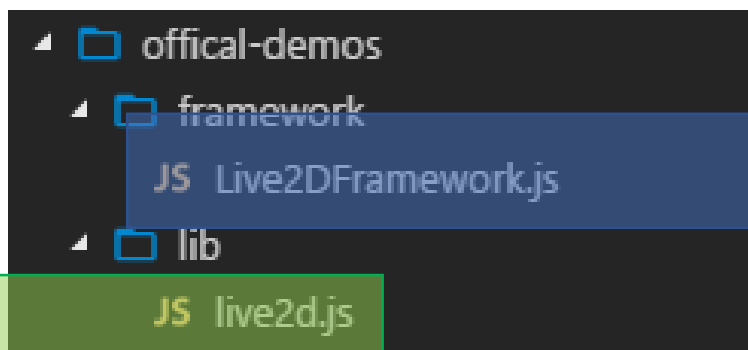


Live2D 核心库
提供 Live2D 工作的基础功能



Live2D 框架
提供基于平台的业务功能

对应文件：



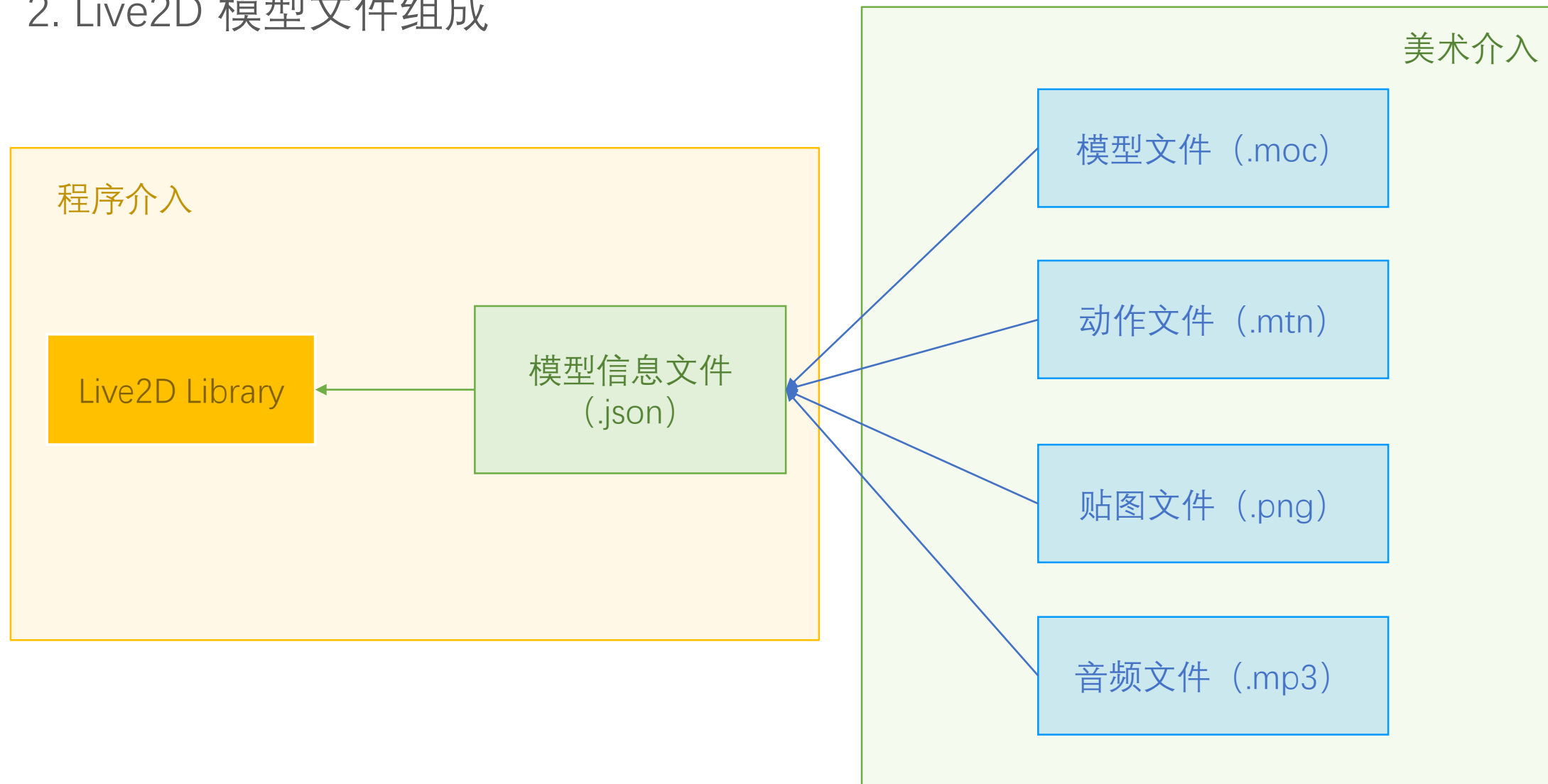
核心库
Live2d.js

基础功能：实现 Live2D 能够启动的所有必要逻辑。
(仅仅是能够启动)

Framework
Live2dFramework.js

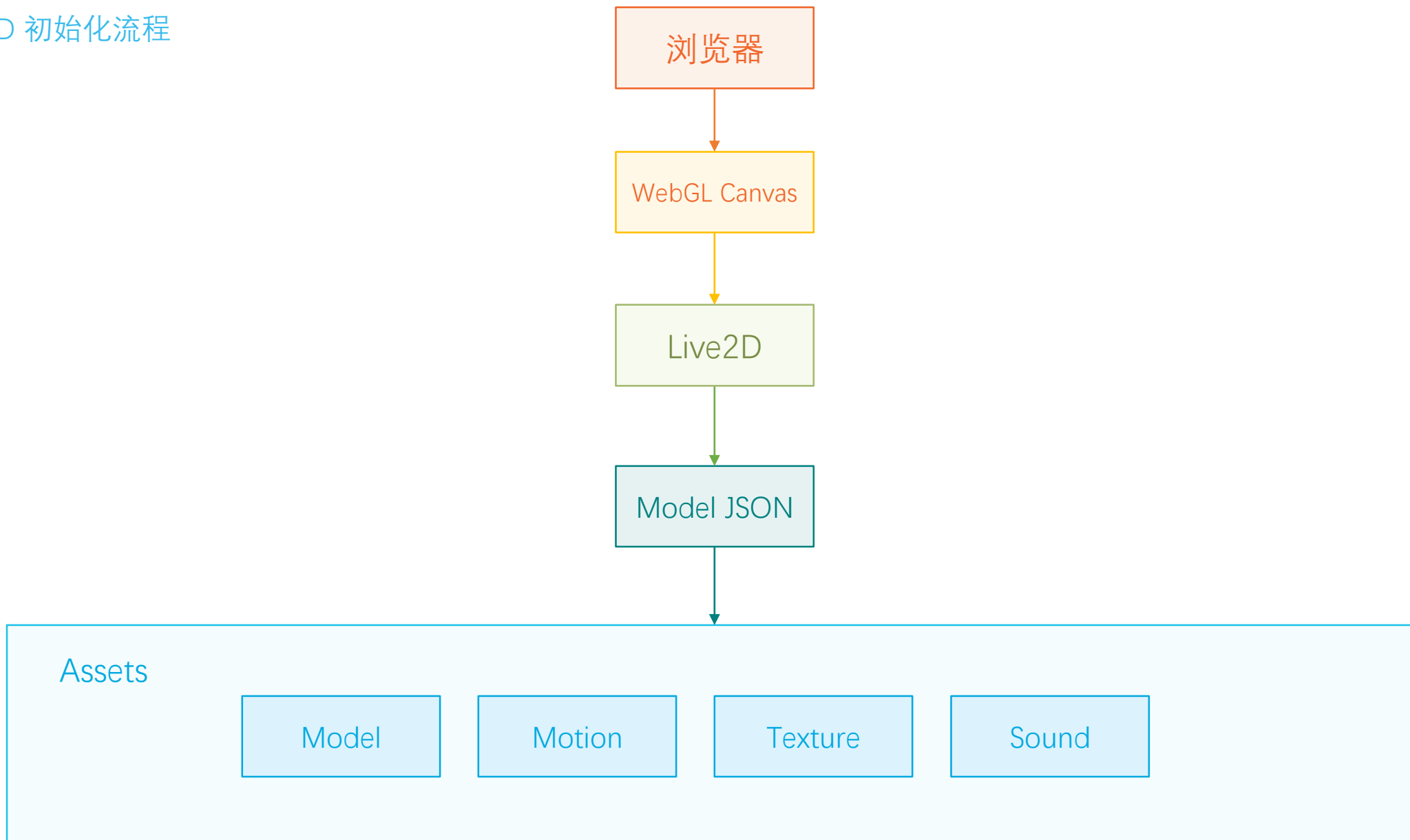
业务功能：除了基础功能外的其他所有功能.

2. Live2D 模型文件组成

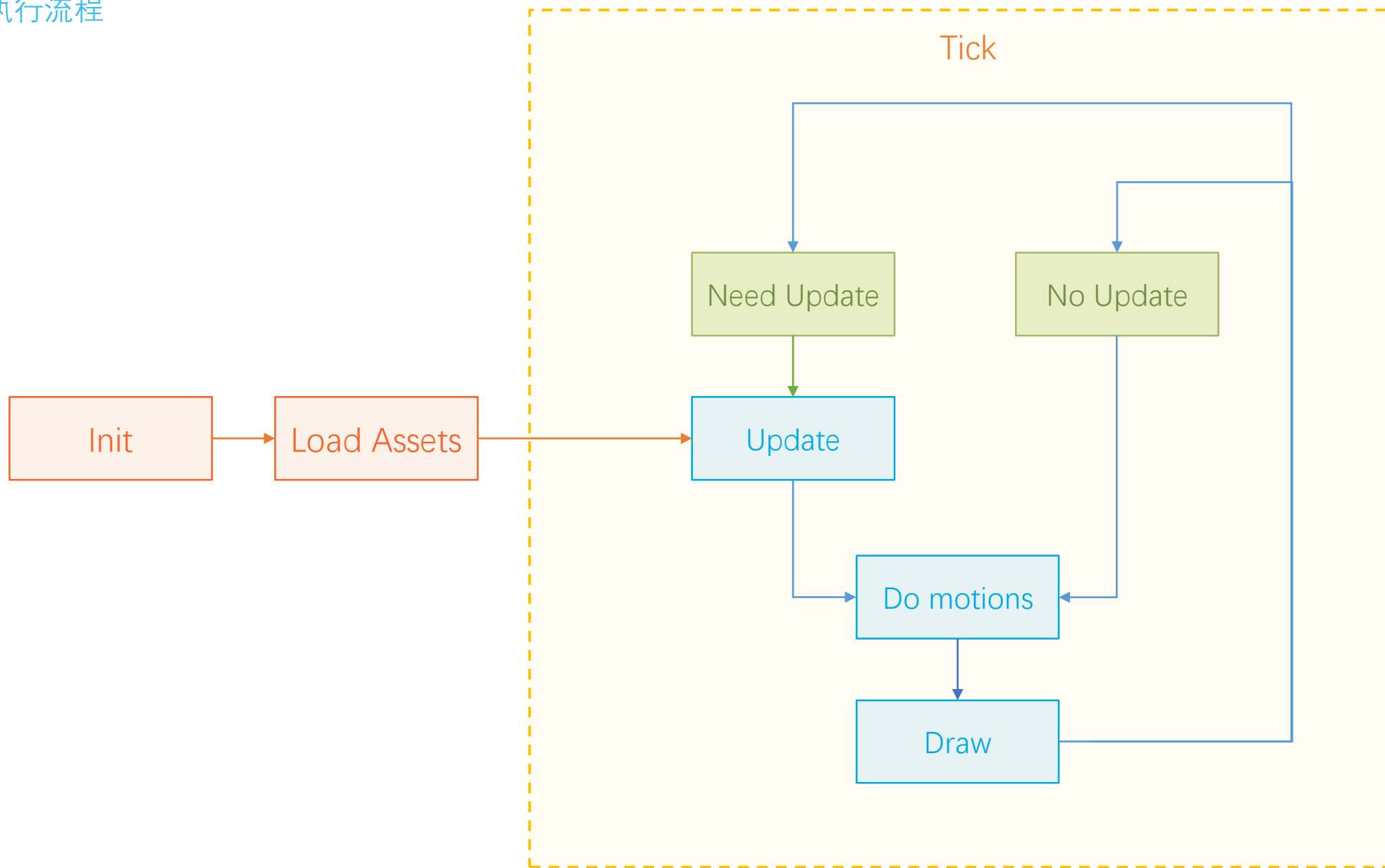


二. Live2D 执行流程

Live2D 初始化流程



Live2D 执行流程

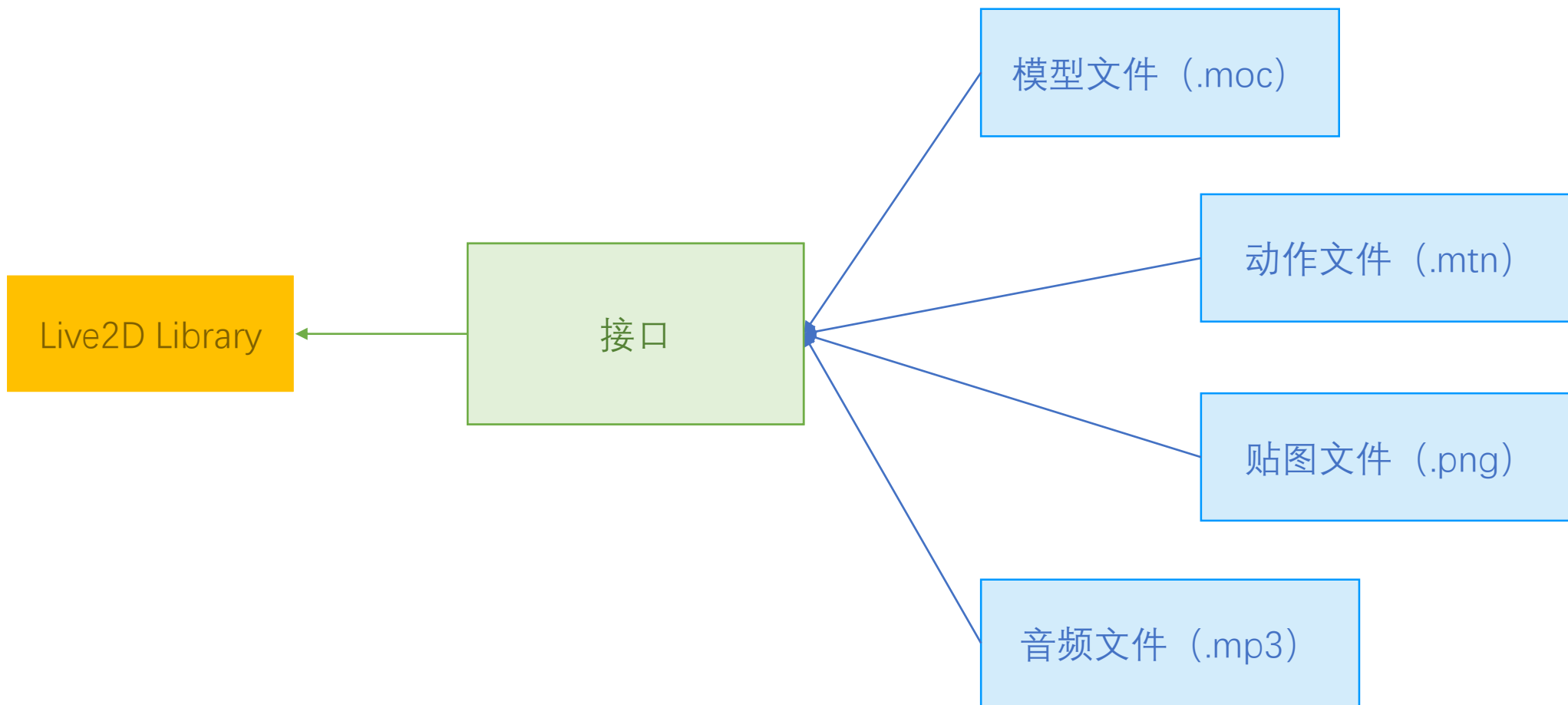


三. 开发实践

开发怎么可能不踩坑 (´□`)┐┌

1. 和服务端进行结合

纯粹的 Live2D 直接读取静态资源即可
实际业务中需要载入服务器端的个性化数据



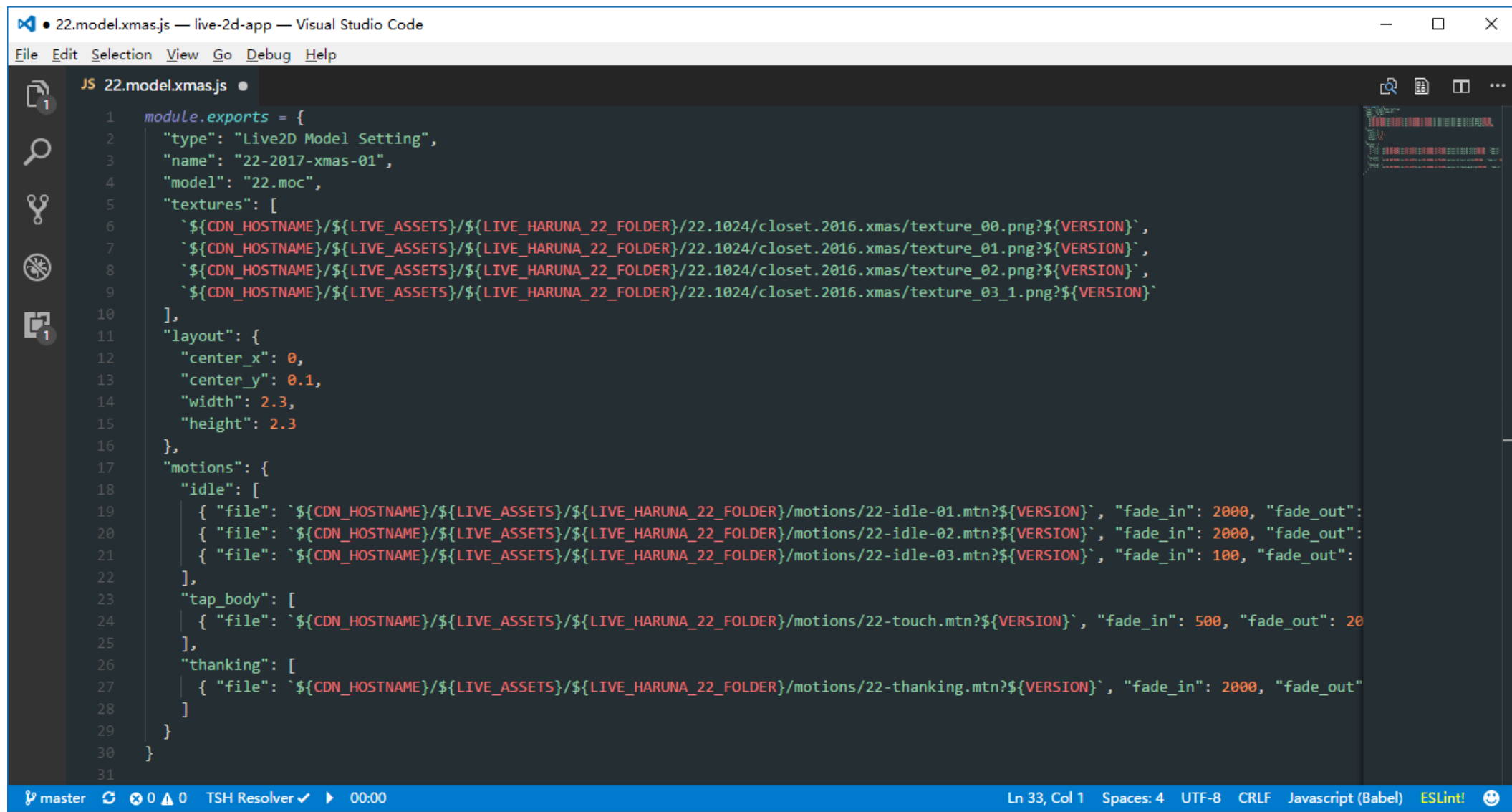
2. 资源管理

使用后台配置来自定义静态资源路径

方便文件替换与增量更新

Live2D 更变相对普通业务较少，可酌情考虑人工管理

对模型信息文件进行动态配置



```
1  module.exports = {
2    "type": "Live2D Model Setting",
3    "name": "22-2017-xmas-01",
4    "model": "22.moc",
5    "textures": [
6      `${CDN_HOSTNAME}/${LIVE_ASSETS}/${LIVE_HARUNA_22_FOLDER}/22.1024/closet.2016.xmas/texture_00.png?${VERSION}`,
7      `${CDN_HOSTNAME}/${LIVE_ASSETS}/${LIVE_HARUNA_22_FOLDER}/22.1024/closet.2016.xmas/texture_01.png?${VERSION}`,
8      `${CDN_HOSTNAME}/${LIVE_ASSETS}/${LIVE_HARUNA_22_FOLDER}/22.1024/closet.2016.xmas/texture_02.png?${VERSION}`,
9      `${CDN_HOSTNAME}/${LIVE_ASSETS}/${LIVE_HARUNA_22_FOLDER}/22.1024/closet.2016.xmas/texture_03_1.png?${VERSION}`
10   ],
11   "layout": {
12     "center_x": 0,
13     "center_y": 0.1,
14     "width": 2.3,
15     "height": 2.3
16   },
17   "motions": {
18     "idle": [
19       { "file": `${CDN_HOSTNAME}/${LIVE_ASSETS}/${LIVE_HARUNA_22_FOLDER}/motions/22-idle-01.mtn?${VERSION}`, "fade_in": 2000, "fade_out": 2000 },
20       { "file": `${CDN_HOSTNAME}/${LIVE_ASSETS}/${LIVE_HARUNA_22_FOLDER}/motions/22-idle-02.mtn?${VERSION}`, "fade_in": 2000, "fade_out": 2000 },
21       { "file": `${CDN_HOSTNAME}/${LIVE_ASSETS}/${LIVE_HARUNA_22_FOLDER}/motions/22-idle-03.mtn?${VERSION}`, "fade_in": 100, "fade_out": 100 }
22     ],
23     "tap_body": [
24       { "file": `${CDN_HOSTNAME}/${LIVE_ASSETS}/${LIVE_HARUNA_22_FOLDER}/motions/22-touch.mtn?${VERSION}`, "fade_in": 500, "fade_out": 2000 }
25     ],
26     "thanking": [
27       { "file": `${CDN_HOSTNAME}/${LIVE_ASSETS}/${LIVE_HARUNA_22_FOLDER}/motions/22-thanking.mtn?${VERSION}`, "fade_in": 2000, "fade_out": 2000 }
28     ]
29   }
30 }
```

Ln 33, Col 1 Spaces: 4 UTF-8 CRLF Javascript (Babel) ESLint!

3. 库的改进

Live2D 库本身需要进行一些改进来适配业务开发

Live2D 是一个很惊艳的工具

不过官方库还是有很多不方便的地方…

- 代码不能直接整合至现代工作流（不是现阶段前端开发人员的工程习惯）
- 深入开发时的代码可维护性较差（全局变量，使用 this 引用 window，大碗面条，结构不分明 ……）
- 在业务中需要调用底层方法（不太友好）
- 没有完善的事件回调，无法满足业务需求（功能对于复杂业务讲不太全）
- 没有缓存机制，播放一个动作就请求一次（不必要请求）
- 没有跨域能力，无法访问 CDN 文件（不适合中大型规模站点部署）
- …

于是做了点微小的工作…

- 将代码进行 模块化 拆离以便整合至现代工作流 (ES Module)
- 封装了 Live2D 模型 控制函数 供业务直接使用 (简化业务代码)
- 加入了一些 事件钩子 以方便实现特定功能 (加入必要函数)
- 人性化了一点 API 风格 (Promise, 链式, 函数名 ...)
- 加入了可配置的 缓存 与 跨域 功能 (完善功能)
- 顺便 修正 了一点点 不完善 的地方
- ...

```
import live2DConfig from './config/live-haruna'
import { Live2DManager } from './live-2d'
import { getGLContext, getAnimationFrame } from './utils'
```

```
/**
```

```
 * Live2D App 初始化逻辑.
```

```
 * 函数中的 this 为组件的上下文对象.
```

```
 *
```

```
 * @export
```

```
 */
```

```
module.exports = function live2DAppInit () {
```

```
  const vm = this
```

```
  // 变量定义.
```

```
  const canvas = this.$refs.canvas
```

```
  const glContext = getGLContext(canvas)
```

```
  // Live2D App 控制对象.
```

```
  const live2DApp = {
```

```
    canvas,
```

```
    glContext,
```

```
    modelJsonPath: '///api.live.bilibili.com/i/getUserKanBanModel',
```

```
    live2DManager: new Live2DManager(canvas, glContext, {
```

```
      presetMotion: 'idle'
```

```
    })),
```

```
    isDrawing: false,
```

```
    dragManager: new window.L2DTargetPoint(),
```

```
    viewMatrix: new window.L2DViewMatrix(),
```

```
    projMatrix: new window.L2DMatrix44(),
```

```
    deviceToScreen: new window.L2DMatrix44(),
```

```
    oldLen: 0,
```

```
    lastMouse: {
```

```
      x: 0,
```

```
      y: 0
```

```
    },
```

```
    isModelShown: false
```

以前的 live2DAppInit()

接近 200 行

现在的 live2DAppInit()

就这么多！

```
live2DAppInit () {
  const live2DApp = new Live2DApp({
    canvas: this.$refs.canvas,
    appConfig: live2DConfig
  })

  // 挂载至视图对象.
  this.live2DApp = live2DApp
},
```

以前处理事件

需要调取底层函数

```
287
288     thisRef.dragMgr.setPoint(0, 0);
289 }
290
291
292 function mouseEvent(e)
293 {
294     e.preventDefault();
295
296     if (e.type == "mousewheel") {
297
298         if (e.clientX < 0 || thisRef.canvas.clientWidth < e.clientX ||
299             e.clientY < 0 || thisRef.canvas.clientHeight < e.clientY)
300         {
301             return;
302         }
303
304         if (e.wheelDelta > 0) modelScaling(1.1);
305         else modelScaling(0.9);
306
307     } else if (e.type == "mousedown") {
308
309         if ("button" in e && e.button != 0) return;
310
311         modelTurnHead(e);
312
313     } else if (e.type == "mousemove") {
314
315         followPointer(e);
316
317     } else if (e.type == "mouseup") {
318
319         if ("button" in e && e.button != 0) return;
320
321         lookFront();
322
323     } else if (e.type == "mouseout") {
324
325         lookFront();
326
327     } else if (e.type == "contextmenu") {
328
329         changeModel();
330
331     }
332 }
333
334
335 }
```

现在处理事件

使用封装后的公开方法

```
// 简单创建 Live2D App!
const live2DApp = new Live2DApp({
    canvas: '#live-2d-canvas',
    appConfig: config
})

// 模型载入完毕.
live2DApp.onload(info => {
    console.log('Model loaded: ', info)
})

// 更换模型. 不指定参数则挨个更换已配置模型.
live2DApp.changeModel().then(modelInfo => {
    console.log('Model changed: ', modelInfo)
})

// 随机播放动作.
live2DApp.startRandomMotion()

// 播放特定动作组.
live2DApp.startMotionGroup('thanking')

// 通过 URL 直接载入外部模型.
live2DApp.loadModelByUrl('///api.live.bilibili.com/i/getUserKanBanModel')
    .then(modelInfo => {
        console.log('Model loaded by url: ', modelInfo)
    })
```


详细的 API 在这里不再陈述…

关于 Live2D 的 License

- 独立开发者可以不用订阅任何 License 即可免费使用.
- 商业用户需要订阅付费 License 才能使用.

更多信息：<http://www.live2d.com/en/products/releaselicense>

* 独立开发者：公司年收入 1000, 0000 日元（大约 61 万人民币）以下的用户.

* 商业用户：非以上用户.

谢谢大家！

Q&A