

uniapp安卓YYEVAPlayer MP4礼物播放器原生插件

目 录

使用说明	1
使用方法	1.1
更新日志	1.2
插件方法	2
开始播放	2.1
停止播放	2.2
插件事件	3
网络地址开始下载	3.1
网络地址下载中	3.2
网络地址下载完成	3.3
网络地址下载出错	3.4
视频开始播放	3.5
视频重复播放	3.6
视频播放结束	3.7
播放器被销毁	3.8
播放出错	3.9

使用方法

介绍

YYEVA (YY Effect Video Animate) 是一个开源的支持可插入动态元素的MP4动效播放器解决方案，包含设计资源输出的AE插件，客户端渲染引擎，在线预览工具。

对比传统的序列帧的动画播放方式，具有更高的压缩率，硬解码效率更高的优点，同时支持插入动态的业务元素；对比SVGA、Lottie等播放器，支持更多的特效支持，如复杂3D效果、描边、粒子效果等，达到所见即所得的效果。

本插件是根据github开源代码<https://github.com/yylive/YYEVA>开发，详情请参考github

相关文档介绍

- [第一篇:直播间礼物动效 - 实现方案](#)
- [第二篇:透明MP4礼物](#)
- [第三篇:变换矩阵在动画上一些应用](#)
- [第四篇, 让MP4静态资源也能够动态起来](#)
- [第五篇设计规范](#)
- [第六篇数据结构](#)

YYEVA视频教程

- [视频1相关工具安装](#)
- [视频2环境搭建](#)
- [视频3:YYEVA-如何创建一个mask合成](#)
- [视频4:YYEVA-制作mask_text](#)
- [视频5:YYEVA-制作mask_image](#)
- [视频6:YYEVA-如何安装或更新CEP插件](#)

联系作者

使用方法

关注微信公众号可联系作者



插件地址

<https://ext.dcloud.net.cn/plugin?id=16747>

示例文件

如果下载的文件不是zip文件，请将扩展名改为zip即可

演示程序

百度网盘

下载地址：<https://pan.baidu.com/s/1IsdkHsE6u6of7zMjTs9tEg>

提取码：tplg

插件权限

- android.permission.INTERNET
- android.permission.WRITE_EXTERNAL_STORAGE
- android.permission.READ_EXTERNAL_STORAGE

用法

在需要使用插件的页面加载以下代码

```
<leven-yyevaPlayer ref="refLevenYyevaPlayer" style="flex:1; height: 300px;" @onDownloadComplete="onDownloadComplete" @onDownloadError="onDownloadError" @onVideoComplete="onVideoComplete" @onVideoDestroy="onVideoDestroy" @onVideoError="onVideoError">
</leven-yyevaPlayer>
```

插件仅支持nvue页面

页面内容

```
<template>
  <view>
    <uni-card title="uniapp安卓YyevaPlayer MP4礼物播放器原生插件">
      <leven-yyevaPlayer ref="refLevenYyevaPlayer" style="flex:1; height: 300px;"
        @onDownloadComplete="onDownloadComplete" @onDownloadError="onDownloadError"
        @onVideoComplete="onVideoComplete" @onVideoDestroy="onVideoDestroy" @onVideoError="onVideoError">
      </leven-yyevaPlayer>
      <button type="primary" @click="play">开始播放</button>
      <button type="primary" @click="stop">停止播放</button>
      <button type="primary" @click="logStr = ''">清空日志</button>
    </uni-card>
    <uni-card class="uni-card-box" title="日志">
      <view><text style="font-size: 14px; flex-wrap: wrap;">{{logStr}}</text></view>
    </uni-card>
  </view>
</template>
```

```
<script>
  export default {
    data() {
      return {
        logStr: ""
      }
    },
    methods: {
      // 开始播放
      play() {
        if (this.$refs.refLevenYyevaPlayer) {
          this.$refs.refLevenYyevaPlayer.play({
            url: "http://lxcode.bs2cdn.yy.com/084e52e9-fd58-4967-ba8b-cd3c4d6c1849"
          }, res => {
            this.writeLog(JSON.stringify(res))
          })
        }
      },
    },
  },
```

```

// 停止播放
stop() {
  if (this.$refs.refLevenYyevaPlayer) {
    this.$refs.refLevenYyevaPlayer.stop(res => {
      this.writeLog(JSON.stringify(res))
    })
  }
},
// 网络地址开始下载
onDownloadStart(e) {
  this.writeLog("onDownloadStart:" + JSON.stringify(e))
},
// 网络地址下载中
onDownloading(e) {
  this.writeLog("onDownloading:" + JSON.stringify(e))
},
// 网络地址下载完成
onDownloadComplete(e) {
  this.writeLog("onDownloadComplete:" + JSON.stringify(e))
},
// 网络地址下载出错
onDownloadError(e) {
  this.writeLog("onDownloadError:" + JSON.stringify(e))
},
// 视频开始播放
onVideoStart(e) {
  this.writeLog("onVideoStart:" + JSON.stringify(e))
},
// 视频重复播放
onVideoRestart(e) {
  this.writeLog("onVideoRestart:" + JSON.stringify(e))
},
// 视频播放结束
onVideoComplete(e) {
  this.writeLog("onVideoComplete:" + JSON.stringify(e))
},
// 播放器被销毁
onVideoDestroy(e) {
  this.writeLog("onVideoDestroy:" + JSON.stringify(e))
},
// 播放出错
onVideoError(e) {
  this.writeLog("onVideoError:" + JSON.stringify(e))
},
// 写日志
writeLog(str) {
  let logStr = uni.$lv.date.format(null, "yyyy-mm-dd hh:MM:ss") + " " + str
  this.logStr = logStr + this.logStr;
}
}

```

使用方法

```
}  
</script>  
  
<style>  
  
</style>
```

更新日志

2024-05-31

- [优化] 部分细节优化

2024-02-21

首次发布

开始播放

方法名

play

用法

- 用法如下：

```
if (this.$refs.refLevenYyevaPlayer) {
  this.$refs.refLevenYyevaPlayer.play({
    url: "http://lxcode.bs2cdn.yy.com/084e52e9-fd58-4967-ba8b-cd3c4d6c1849.mp4",
  }, res => {
    this.writeLog(JSON.stringify(res))
  })
}
```

- 参数说明

参数名	参数类型	是否必填	默认值	参数描述
url	String	是	无	播放的本地或网络地址

回调

- 示例

```
{
  "message": "",
  "code": 0,
  "data": {}
}
```

- 回调说明：

参数名	参数类型	参数描述
message	String	消息提示

开始播放

参数名	参数类型	参数描述
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

停止播放

方法名

stop

用法

- 用法如下：

```
if (this.$refs.refLevenYyevaPlayer) {
  this.$refs.refLevenYyevaPlayer.stop(res => {
    this.writeLog(JSON.stringify(res))
  })
}
```

- 参数说明
无

回调

- 示例

```
{
  "message": "",
  "code": 0,
  "data": {}
}
```

- 回调说明：

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

网络地址开始下载

方法名

onDownloadStart

用法

- 用法如下：

```
onDownloadStart(e) {  
    console.log("onDownloadStart:" + JSON.stringify(e.detail))  
}
```

- 参数说明

无

回调

- 示例

```
{}
```

- 回调说明：
无

网络地址下载中

方法名

onDownloading

用法

- 用法如下：

```
onDownloading(e) {  
    console.log("onDownloading:" + JSON.stringify(e.detail))  
}
```

- 参数说明

无

回调

- 示例

```
{  
    "percent": 0.0535341242009247,  
    "size": 17067,  
    "totalSize": 318806  
}
```

- 回调说明：

参数名	参数类型	参数描述
percent	Float	下载进度
size	Integer	当前已下载的文件的大小
totalSize	Integer	文件总大小

网络地址下载完成

方法名

onDownloadComplete

用法

- 用法如下：

```
onDownloadComplete(e) {  
    console.log("onDownloadComplete:" + JSON.stringify(e.detail))  
}
```

- 参数说明

无

回调

- 示例

```
{  
  "path": "/storage/emulated/0/Android/data/test.leven.uniplugin.com/cache/084  
}
```

- 回调说明：

参数名	参数类型	参数描述
path	String	下载后存储的本地路径

网络地址下载出错

方法名

onDownloadError

用法

- 用法如下：

```
onDownloadError(e) {  
    console.log("onDownloadError:" + e.detail)  
}
```

- 参数说明

无

回调

- 示例

下载出错

- 回调说明：
无

视频开始播放

方法名

onVideoStart

用法

- 用法如下：

```
onVideoStart(e) {  
    console.log("onVideoStart:" + JSON.stringify(e.detail))  
}
```

- 参数说明

无

回调

- 示例

```
{}
```

- 回调说明：
无

视频重复播放

方法名

onVideoRestart

用法

- 用法如下：

```
onVideoRestart(e) {  
    console.log("onVideoRestart:" + JSON.stringify(e.detail))  
}
```

- 参数说明

无

回调

- 示例

```
{}
```

- 回调说明：
无

视频播放结束

方法名

onVideoComplete

用法

- 用法如下：

```
onVideoComplete(e) {  
    console.log("onVideoComplete:" + JSON.stringify(e.detail))  
}
```

- 参数说明

无

回调

- 示例

```
{}
```

- 回调说明：

无

播放器被销毁

方法名

onVideoDestroy

用法

- 用法如下：

```
onVideoDestroy(e) {  
    console.log("onVideoDestroy:" + JSON.stringify(e.detail))  
}
```

- 参数说明

无

回调

- 示例

```
{}
```

- 回调说明：
无

播放出错

方法名

onVideoError

用法

- 用法如下：

```
onVideoError(e) {  
    console.log("onVideoError:" + JSON.stringify(e.detail))  
}
```

- 参数说明

无

回调

- 示例

```
{}
```

- 回调说明：

无