

[illegible]

目录

EAPIL RENDER SDK..... 1

接口说明.....1

目录.....1

1.引言.....1

2.宏定义.....1

3.结构体说明.....1

4.回调函数说明.....2

5.接口说明.....2

    INITOPENGL.....2

    UNINITOPENGL.....2

    RENDER.....3

    ONBUTTONDOWN.....3

    ONBUTTONUP.....3

    ONMOUSEMOVE.....3

    ONMOUSEWHEEL.....4

    TRANSLATEVIDEOData.....4

    TRANSLATEIMAGEDATA.....4

    SETWINDOW.....4

    LOADTEMPLATE.....5

    SETTARGETFRAMERATE.....5

    SETPLAYERTYPE.....5

    SETROTATEVR.....5

    TRANSOPENGLFBOID.....6

    WHETHERSUPPORTIMAGE.....6

    RESETSTANDARDIZATION.....6

    MAKESTANDARDIZATION.....6

    ENABLERECORDSCREEN.....7

    DISABLERECORDSCREEN.....7

    SETRECORDSCREENCALLBACK.....7

    SAVESCREENIMAGE.....7

    SETBALLPOSRange.....8

    GETSDKVERSION.....8

    SETEXPIREKEY.....8

## 1.引言

本文档主要为易瞳全景渲染器的 SDK 的操作接口作说明，给内部开发人员提供参考。

名称	EAPIL RENDER SDK
功能	音视频播放库
SDK 头文件	c.h
Lib 文件	EPRenderSDK.lib / EPRenderSDK.a
动态库文件	EPRenderSDK.dll / EPRenderSDK.so
其他相关头文件	无
其他依赖库文件	无
编译环境	CMAKE + VISUAL STUDIO/CMAKE
支持系统	Xp/vista/win 8/win 10/Linux/mac
支持系统位数	32bit/64bit

## 2.宏定义

宏	值	说明
ERR_RENDER_SUCCEED	0	执行成功
ERR_RENDER_FAIL	-1	执行失败
ERR_RENDER_INVALIDPARAM	0x90000001	输入参数非法
ERR_RENDER_NOMEMORY	0x90000002	系统内存不足
ERR_RENDER_NOINIT	0x90000003	没有初始化
ERR_RENDER_NOSUPPORT	0x90000004	硬件平台不支持
ERR_RENDER_ORDER	0x90000005	SDK 接口的调用次序错误

## 3.结构体说明

名称: PlayerType

说明: 枚举播放特效类型

原型:

```
typedef enum
{
    RENDERBALL,           //普通球显示
    RENDERSMALLPLANET,    //小星球显示
    RENDERWIDESCREEEN,    //宽屏显示
    RENDERVR,             //VR 显示
    RENDERBALLORTHO,      //正交显示
    RENDERBALLTWOSCREEN,   //双分屏显示
    RENDERBALLFOURSCREEN, //四分屏显示
    RENDERBALLTHREESCREEN, //三分屏显示
}
```

```
}PlayerType;
```

名称: `TemplateType`

说明: 模板数据格式类型

原型:

```
typedef enum
{
    CAMERAEN = 0,    //从相机读取加密模板
    LOCAL,           //从本地读取加密或者非加密文件
    STRINGEN,        //读取加密字符串
    STRING,          //读取非加密字符串
    REMOTE,           //从远程获取带加密模板 (url)
    STRINGENEXT1     //新格式模板
}TemplateType;
```

## 4.回调函数说明

原型:

```
typedef void(*RecordScreenCallBack)(unsigned char *rgbData, int width, int height);
```

功能: 录屏回调函数。

返回值: 无。

参数:

参数	类型	说明
rgbData	unsigned char *	当前显示帧数据指针 (RGB32 格式)
width	int	数据宽
height	int	数据高

备注: 无。

## 5.接口说明

### InitOpenGL

原型: `void InitOpenGL()`

功能: 一次初始化完所有渲染特效的 shader、vbo、fbo 等 gl 渲染上下文相关的模块。

返回值: 无。

参数: 无。

备注: 此接口与 `UnInitOpenGL` 配对使用。

### UnInitOpenGL

原型: `void UnInitOpenGL()`

功能: 释放 gl 渲染上下文相关模块。

返回值: 无。

参数：无。

备注：只有当 gl 渲染上下文变化时才需要调用此函数。

## Render

原型：bool Render()

功能：渲染解码数据。

返回值：

成功：true，失败：false。

参数：无。

备注：无。

## OnMouseDown

原型：void OnMouseDown(int x, int y)

功能：响应鼠标左键点击事件。

返回值：无。

参数：

参数	类型	说明
x	int	鼠标位置 x 坐标
y	int	鼠标位置 y 坐标

## OnButtonUp

原型：void OnButtonUp()

功能：响应鼠标左键弹起事件。

返回值：无。

参数：无。

备注：无。

## OnMouseMove

原型：void OnMouseMove(int x, int y)

功能：响应鼠标移动事件。

返回值：无。

参数：

参数	类型	说明
x	int	鼠标位置 x 坐标
y	int	鼠标位置 y 坐标

备注：无。

## OnMouseWheel

原型: void OnMouseWheel(short delta)

功能: 响应鼠标滚轮事件。

返回值: 无。

参数:

参数	类型	说明
delta	short	鼠标滑轮滑动值

备注: 无

## TranslateVideoData

原型:

void TranslateVideoData(unsigned char \* yuvData, int width, int height);

功能: 将解码出的 yuv420 数据传入到渲染器中。

返回值: 无。

参数:

参数	类型	说明
yuvData	unsigned char *	传入数据的指针
width	int	数据宽
height	int	数据高

备注:

传入的数据将备份到渲染器中。

## TranslateImageData

原型: void TranslateImageData(unsigned char \* rgbaData, int width, int height)

功能: 将解码出的 rgb 数据传入到渲染器中。

返回值: 无。

参数:

参数	类型	说明
RgbData	unsigned char *	传入数据的指针
width	int	数据宽
height	int	数据高

备注: 由于 rgb 数据占用内存大, 建议只有当显示图片 (只需要调用一次) 的时候调用此接口。

## SetWindow

原型: void SetWindow(int width, int height)

功能: 设置渲染窗口大小。

返回值: 无。

参数:

参数	类型	说明
width	int	窗口宽
height	int	窗口高

备注: 渲染窗口有变化的时候需调用此接口。

### LoadTemplate

原型: void LoadTemplate(std::string path, TemplateType type)

功能: 加载模板。

返回值: 无。

参数:

参数	类型	说明
path	string	模板路径文件名或者字符串数据
type	TemplateType	模板数据格式类型

备注: 无。

### SetTargetFrameRate

原型: void SetTargetFrameRate(float rate)

功能: 设置渲染帧率

返回值: 无

参数:

参数	类型	说明
rate	float	帧率

备注: 无

### SetPlayerType

原型: void SetPlayerType(PlayerType type)

功能: 设置渲染类型

返回值: 无。

参数:

参数	类型	说明
type	PlayerType	渲染类型 (详见结构体说明)

备注: 无。

### SetRotateVR

原型: void SetRotateVR(float rotateX, float rotateY, float rotateZ)

功能: VR 模式下设置绕轴旋转角度(°)

返回值：无。

参数：

参数	类型	说明
rotateX	float	绕 X 轴旋转角度
rotateY	float	绕 Y 轴旋转角度
rotateZ	float	绕 Z 轴旋转角度

备注：坐标系为：Y 轴向上 X 轴向右 Z 轴朝外的右手坐标系。

## TransOpenGLFboID

原型：void TransOpenGLFboID(int id)

功能：设置当前渲染画布（fbo）的 id。

返回值：无。

参数：

参数	类型	说明
Id	int	帧缓冲对象 id

备注：如果当前 gl 渲染目标为帧缓冲对象，则需要调用此接口。

## WhetherSupportImage

原型：bool WhetherSupportImage(std::string name)

功能：判断图片是否是渲染器支持的格式。

返回值：支持：true，不支持：false。

参数：

参数	类型	说明
name	string	图片资源路径名

备注：无

## ResetStandardization

原型：void ResetStandardization()

功能：重置标定信息。

返回值：无。

参数：无。

备注：无。

## MakeStandardization

原型：void MakeStandardization()

功能：将当前球的旋转状态设置为标准（初始）状态。

返回值：无。

参数：无。



备注：无。

### EnableRecordScreen

原型：void EnableRecordScreen()

功能：开启录屏回掉。

返回值：无。

参数：无。

备注：无。

### DisableRecordScreen

原型：void DisableRecordScreen()

功能：禁用录屏回掉

返回值：无。

参数：无。

备注：无。

### SetRecordScreenCallBack

原型：void SetRecordScreenCallBack(RecordScreenCallBack callBack)

功能：设置录屏回掉函数。

返回值：无。

参数：

参数	类型	说明
callBack	RecordScreenCallBack	录屏回调函数指针

备注：无。

### SaveScreenImage

原型：int SaveScreenImage(std::string fileName, bool wideScreen = true)

功能：截屏。

返回值：0 成功 -1 失败。

参数：

参数	类型	说明
fileName	std::string	截屏文件存储路径名
wideScreen	bool	true:截取宽屏图 false:截取当前屏幕图

备注：无。

## SetBallPosRange

原型: `void SetBallPosRange(float minDisToEye, float maxDisToEye, float initDis)`

功能: 初始化球体的比例大小。

返回值: 无。

参数:

参数	类型	说明
minDisToEye	float	观察点与球的最小距离 (默认值 0.48)
maxDisToEye	float	观察点与球的最大距离 (默认值 3.0)
initDis	float	观察点与球的实际距离 (默认值 2.5)

备注: 无。

## GetSDKVersion

原型: `std::string GetSDKVersion()`

功能: 获取渲染器版本信息。

返回值: 渲染器版本信息。

参数: 无。

备注: 无。

## SetExpireKey

原型: `void SetExpireKey(char * key)`

功能: 设置渲染 SDK key。

返回值: 无。

参数:

参数	类型	说明
key	char *	渲染器 key 字符串

备注: 无。