1 功能概述

云课堂场景模块位于 `PolyvLiveCloudClassScene` 文件夹之下，页面实现为 **PLVLCCloudClassViewController**

云课堂有直播、回放两种场景模式，两种模式的UI界面高度共用，功能模块封装成多个UI区域（AreaView），在页面中通过简单地布局组合即可完成集成。

直播场景模式支持的功能有：媒体（播放器、PPT、浮窗、竖屏皮肤）、页面菜单、连麦、互动应用、横屏聊天室、横屏皮肤。 回放场景模式支持的功能有：媒体（播放器、PPT、浮窗、竖屏皮肤）、页面菜单、横屏聊天室、横屏皮肤。

各个UI功能模块的区域类如下：

- 媒体区域：PLVLCMediaAreaView
- 连麦区域：PLVLCLinkMicAreaView
- 菜单区域：PLVLCLivePageMenuAreaView
- 横屏聊天区域：PLVLCChatLandscapeView
- 横屏皮肤区域：PLVLCLiveRoomPlayerSkinView
- 互动应用区域：PLVInteractView

2 初始化对象、数据

主要初始化获取登录频道信息和设置、监听聊天室socekt。

```
- (instancetype)init {
    self = [super init];
    if (self) {
        [[PLVRoomDataManager sharedManager] addDelegate:self delegateQueue:dispatch_get_main_queue()];
        PLVRoomData *roomData = [PLVRoomDataManager sharedManager].roomData;
        PLVRoomUser *roomUser = roomData.roomUser;

        /// Socket 登录管理
        PLVSocketUserType userType = roomUser.viewerType == PLVRoomUserTypeStudent ? PLVSocketUserTypeStudent :
PLVSocketUserTypeSlice;
        [[PLVSocketManager sharedManager] loginWithChannelId:roomData.channelId
                                                    viewerId:roomUser.viewerId
                                                  viewerName:roomUser.viewerName
                                                   avatarUrl:roomUser.viewerAvatar actor:nil userType:userType];

        socketDelegateQueue = dispatch_get_global_queue(0, DISPATCH_QUEUE_PRIORITY_DEFAULT);
        [[PLVSocketManager sharedManager] addDelegate:self delegateQueue:socketDelegateQueue];

        // 启动聊天室管理器
        [[PLVLCChatroomViewModel sharedViewModel] setup];
        [[PLVLCChatroomViewModel sharedViewModel] addDelegate:self delegateQueue:dispatch_get_main_queue()];
    }
    return self;
}
```

- `PLVRoomDataManager` 和 `PLVRoomData` 具体实现可参考7_2 核心common-数据
- `PLVSocketManager` 具体实现可参考7_4 核心common-聊天室

3 初始化配置（添加页面旋转、互动通知）

```
- (void)setupModule{
    // 通用的 配置
    [[NSNotificationCenter defaultCenter] addObserver:self
                                             selector:@selector(interfaceOrientationDidChange:)
                                                 name:UIApplicationDidChangeStatusBarOrientationNotification
object:nil];

    PLVRoomData *roomData = [PLVRoomDataManager sharedManager].roomData;
    if (roomData.videoType == PLVChannelVideoType_Live) { // 视频类型为 直播
        /// 监听事件
        [[NSNotificationCenter defaultCenter] addObserver:self selector:@selector(notificationForOpenBulletin:)
name:PLVLCChatroomOpenBulletinNotification object:nil];

    } else if (roomData.videoType == PLVChannelVideoType_Playback){ // 视频类型为 直播回放

    }
}
```

4 初始化页面UI

```
- (void)setupUI {
    /// 注意：1. 此处不建议将共同拥有的图层，提炼在 if 判断外，来做“代码简化”
    ///          因为此处涉及到添加顺序，而影响图层顺序。放置在 if 内，能更加准确地配置图层顺序，也更清晰地预览图层顺序。
    ///      2. 懒加载过程中(即Getter)，已增加判断，若场景不匹配，将创建失败并返回nil

    PLVRoomData *roomData = [PLVRoomDataManager sharedManager].roomData;
    if (roomData.videoType == PLVChannelVideoType_Live) { // 视频类型为 直播
        /// 初始化直播的UI

    }else if (roomData.videoType == PLVChannelVideoType_Playback){ // 视频类型为 直播回放
        /// 初始化回放的UI
    }
}
```

根据当前是直播或回放，将选择不同的AreaView控件组合进行初始化。

5 设置AreaView代理回调

由于各个功能之间需要进行通信，例如播放器的直播开始和直播结束状态会影响到媒体的显示和隐藏，因此每个布局都提供了外部可以使用的回调，方便各个功能模块进行通信。

详细接口介绍请参考各个功能模块的具体介绍文章。

回调设置和处理请参考页面中的*设置AreaView回调*方法块。

6 处理屏幕旋转

云课堂场景的直播和回放均支持横竖屏观看，需要在页面中做对应的代码处理。

参考代码块：*屏幕旋转处理*

```
- (void)viewWillLayoutSubviews {
    [super viewWillLayoutSubviews];
    [self updateUI];
}

- (void)interfaceOrientationDidChange:(NSNotification *)notification {
    BOOL fullScreen = [UIScreen mainScreen].bounds.size.width > [UIScreen mainScreen].bounds.size.height;
    self.fullScreenDifferent = (self.currentLandscape != fullScreen);
    self.currentLandscape = fullScreen;

    // 全屏播放器皮肤 liveRoomSkinView 的弹幕按钮 danmuButton 为显示状态，且为非选中状态，且当前为横屏时，才显示弹幕
    BOOL danmuEnable = !self.liveRoomSkinView.danmuButton.selected && !self.liveRoomSkinView.danmuButton.hidden;
    [self.mediaAreaView showDanmu:fullScreen && danmuEnable];

    // 调用setStatusBarHidden后状态栏旋转横屏不自动隐藏
    [[UIApplication sharedApplication] setStatusBarHidden:fullScreen];
}
```