

- 1 功能概述
- 2 初始化及播放
- 3 播放器皮肤
 - 3.1 皮肤控制播放器
 - 3.2 皮肤代理回调
 - 3.3 竖屏皮肤 `PLVLCMediaPlayerSkinView`
 - 3.4 横屏皮肤 `PLVLCLiveRoomPlayerSkinView`
- 4 播放器更多
- 5 播放器画布

1 功能概述

视频播放是多场景项目中提供的基础功能，功能包括直播播放器和回放播放器，在云课堂场景模块中播放器主要功能代码在 `PLVLCMediaAreaView` 中，`PLVLCMediaAreaView` 集成7_3 核心common-播放器中 `PLVPlayerPresenter` 来实现的功能。

2 初始化及播放

代码如下：

```
/// PLVLCMediaAreaView.m

- (void)setupModule{
    if (self.videoType == PLVChannelVideoType_Live) {
        /// 初始化直播 模块
        self.playerPresenter = [[PLVPlayerPresenter alloc] initWithVideoType:PLVChannelVideoType_Live];
        self.playerPresenter.delegate = self;
        [self.playerPresenter setupPlayerWithDisplayView:self.canvasView.playerSuperview];

    }else if (self.videoType == PLVChannelVideoType_Playback){
        /// 初始化直播回放 模块
        self.playerPresenter = [[PLVPlayerPresenter alloc] initWithVideoType:PLVChannelVideoType_Playback];
        self.playerPresenter.delegate = self;
        [self.playerPresenter setupPlayerWithDisplayView:self.canvasView.playerSuperview];
    }
}
```

3 播放器皮肤

云课堂场景播放器部分功能是通过皮肤UI的代理回调来控制的，皮肤分为竖屏皮肤 `PLVLCMediaPlayerSkinView` 和横屏皮肤 `PLVLCLiveRoomPlayerSkinView`，两种皮肤都基于

PLVLCBasePlayerSkinView 扩展而来，横竖屏皮肤UI元素是一样的区别在于UI布局不同。

3.1 皮肤控制播放器

- 直播：刷新按钮、直播时长、全屏按钮
- 回放：播放器播放时间、总时间、进度条、全屏按钮

3.2 皮肤代理回调

横屏皮肤的代理全部定义在几类 PLVLCBasePlayerSkinView 中，具体回调到外部，功能业务不同在不同地方调用回调

- 点击返回

实现处理功能：

- 全屏时将房间改变成竖屏；
- 竖屏时退出房间；

```
/// @param currentFullScreen YES: 全屏、NO: 竖屏
/// - (void)plvLCBasePlayerSkinViewBackButtonClicked:(PLVLCBasePlayerSkinView *)skinView currentFullScreen:
(BOOL)currentFullScreen;
```

- 点击更多

实现处理功能：打开更多视图，更多功能参考[4 播放器更多](#)

```
/// - (void)plvLCBasePlayerSkinViewMoreButtonClicked:(PLVLCBasePlayerSkinView *)skinView;
```

- 点击播放、暂停

实现处理功能：播放、暂停播放器

```
/// @param wannaPlay YES: 播放、NO: 暂停
/// - (void)plvLCBasePlayerSkinViewPlayButtonClicked:(PLVLCBasePlayerSkinView *)skinView wannaPlay:
(BOOL)wannaPlay;
```

- 点击刷新

实现处理功能：重新加载直播视频内容

```
/// -(void)plvLCBasePlayerSkinViewRefreshButtonClicked:(PLVLCBasePlayerSkinView *)skinView;
```

- 拖动进度条

实现处理功能：改变回放播放器播放进度

```
/// @param currentSliderProgress 进度 (0 - 1浮点值 )  
/// - (void)plvLCBasePlayerSkinView:(PLVLCBasePlayerSkinView *)skinView sliderDragEnd:  
(CGFloat)currentSliderProgress;
```

3.3 竖屏皮肤 PLVLCMediaPlayerSkinView

PLVLCMediaPlayerSkinView 对象在 PLVLCMediaAreaView.m 中与 PLVPlayerPresenter 交互, 通过 PLVLCMediaAreaView 的代理与 PLVLCCLoudClassViewController 交互, 同步横屏皮肤UI状态。

3.4 横屏皮肤 PLVLCLiveRoomPlayerSkinView

PLVLCLiveRoomPlayerSkinView 对象在 PLVLCCLoudClassViewController.m 中与 PLVLCMediaAreaView 交互, 达到控制 PLVPlayerPresenter 和同步竖屏皮肤目的。

PLVLCBasePlayerSkinView、PLVLCMediaPlayerSkinView、PLVLCLiveRoomPlayerSkinView 可以在polyv demo项目中找到。

4 播放器更多

云课堂场景播放器切换视频或音频、视频质量（码率）和线路功能是在 PLVLCMediaAreaView.m 中通过 PLVPlayerPresenter 和UI封装类 PLVLCMediaMoreView 实现切换播放器质量线路功能。

- 初始化更多视图，同步播放器状态

```

/// -(void)updateMoreviewWithData {
    // 视频质量选项数据
    PLVLCMediaMoreModel * qualityModel = [PLVLCMediaMoreModel
modelWithOptionTitle:PLVLCMediaAreaView_Data_QualityOptionTitle
optionItemsArray:self.playerPresenter.codeRateNamesOptions];
    [qualityModel setSelectedIndexWithOptionItemString:self.playerPresenter.currentCodeRate];

    // 线路选项数据
    NSMutableArray * routeArray = [[NSMutableArray alloc] init];
    for (int i = 1; i <= self.playerPresenter.lineNum; i++) {
        NSString * route = [NSString stringWithFormat:@"线路%d",i];
        [routeArray addObject:route];
    }
    PLVLCMediaMoreModel * routeModel = [PLVLCMediaMoreModel
modelWithOptionTitle:PLVLCMediaAreaView_Data_RouteOptionTitle optionItemsArray:routeArray
selectedIndex:self.playerPresenter.currentLineIndex];

    // 整合数据
    NSMutableArray * modelArray = [[NSMutableArray alloc] init];
    if (qualityModel) { [modelArray addObject:qualityModel]; }
    if (routeModel) { [modelArray addObject:routeModel]; }

    // 更新 moreView
    [self.moreView updateTableViewWithDataArrayByMatchModel:modelArray];
}

```

- 实现更多代理回调，控制播放器

```

#pragma mark PLVLCMediaMoreViewDelegate
/// -(void)plvLCMediaMoreView:(PLVLCMediaMoreView *)moreView optionItemSelected:(PLVLCMediaMoreModel *)model{
    if ([model.optionTitle isEqualToString:PLVLCMediaAreaView_Data_ModeOptionTitle]) {
        // 用户点选了“模式”中的选项
        self.logoView.hidden = model.selectedIndex != 0;
        [self.canvasView switchTypeTo:(model.selectedIndex == 0 ? PLVLCMediaPlayerCanvasViewType_Video :
PLVLCMediaPlayerCanvasViewType_Audio)];
        [self.playerPresenter switchLiveToAudioMode:(model.selectedIndex == 0 ? NO : YES)];
    } else if ([model.optionTitle isEqualToString:PLVLCMediaAreaView_Data_QualityOptionTitle]) {
        // 用户点选了“视频质量”中的选项
        [self.playerPresenter switchLiveToCodeRate:model.currentSelectedItemString];
    } else if ([model.optionTitle isEqualToString:PLVLCMediaAreaView_Data_RouteOptionTitle]) {
        // 用户点选了“线路”中的选项
        [self.playerPresenter switchLiveToLineIndex:model.selectedIndex];
    } else if ([model.optionTitle isEqualToString:PLVLCMediaAreaView_Data_SpeedOptionTitle]) {
        // 用户点选了“倍速”中的选项
        CGFloat speed = [[model.currentSelectedItemString
substringToIndex:model.currentSelectedItemString.length - 1] floatValue];
        [self.playerPresenter switchLivePlaybackSpeedRate:speed];
    }
}

```

PLVLCMediaMoreView 可以在polyv demo项目找到。

5 播放器画布

云课堂场景播放器画布是根据房间状态(暂无直播、等待直播、直播中、直播断开、直播结束)和播放器模式(视频、音频)，显示相对应UI和通过代理回调控制播放器，此功能UI封装在 `PLVLCMediaPlayerCanvasView` 中，在 `PLVLCMediaAreaView.m` 中实例化和调用。

实现 `PLVLCMediaPlayerCanvasViewDelegate` 代理

在音频模式下，点击播放画面回调此方法，在 `PLVLCMediaPlayerCanvasView` 中处理回调，播放器切换为视频模式

```
- (void)plvLCMediaPlayerCanvasViewPlayCanvasButtonClicked:(PLVLCMediaPlayerCanvasView *)playerCanvasView;
```

`PLVLCMediaPlayerCanvasView` 可以在polyv demo项目中找到。