1 功能概述

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

2 核心类介绍

2.1 初始化及播放

```
 (instancetype)init {
    self = [super init];
    if (self) {
        /// 播放器
        self.playerPresenter = [[PLVPlayerPresenter alloc] initWithVideoType:[PLVRoomDataManager
sharedManager].roomData.videoType];
        self.playerPresenter.openAdv = YES;
        self.playerPresenter.delegate = self;
    }
    return self;
}

- (void)viewDidLoad {
    [super viewDidLoad];

    [self.view addSubview:self.backgroundView];
    [self.view addSubview:self.playerBackgroundView];
    [self.view addSubview:self.displayView];
    [self.view addSubview:self.audioAnimalView];
    [self.view addSubview:self.playButton];

    [self.playerPresenter setupPlayerWithDisplayView:self.displayView];
}

- (void)viewWillLayoutSubviews {
    /// ...
}
```

2.2 播放器控制方法

```
/// 播放直播/回放
- (void)play;

/// 暂停直播/回放
- (void)pause;

/// 静音 播放器
- (void)mute;

/// 取消静音 播放器
- (void)cancelMute;

#pragma mark 直播的API

/// 播放/刷新/重新加载直播
- (void)reload;

/// 切换线路
- (void)switchPlayLine:(NSUInteger)Line showHud:(BOOL)showHud;

/// 切换码率
- (void)switchPlayCodeRate:(NSString *)codeRate showHud:(BOOL)showHud;

/// 切换音频模式
- (void)switchAudioMode:(BOOL)audioMode;
```

3 播放器控制UI

带货场景的播放器控制UI包括屏幕点击控制和回放控制栏控件。

3.1 点击屏幕控制

带货场景的播放器支持双击暂停、单击继续播放功能，此功能是在`PLVECWatchRoomViewController`的view上加上了单、双击手势实现的，代码如下：

```
PLVECWatchRoomViewController.m

- (void)viewDidLoad {
    [super viewDidLoad];

    [self setupUI];
    [self setupData];

    [PLVECFloatingWindow sharedInstance].delegate = self;

    // 单击、双击手势控制播放器和UI
    UITapGestureRecognizer *doubleGestureRecognizer = [[UITapGestureRecognizer alloc] init];
    doubleGestureRecognizer.numberOfTapsRequired = 2;
    doubleGestureRecognizer.numberOfTouchesRequired = 1;
    doubleGestureRecognizer.delegate = self;
    [doubleGestureRecognizer addTarget:self action:@selector(tapAction:)];
    [self.view addGestureRecognizer:doubleGestureRecognizer];

    UITapGestureRecognizer *singleGestureRecognizer = [[UITapGestureRecognizer alloc] init];
    singleGestureRecognizer.numberOfTapsRequired = 1;
    singleGestureRecognizer.numberOfTouchesRequired = 1;
    singleGestureRecognizer.delegate = self;
    [singleGestureRecognizer addTarget:self action:@selector(tapAction:)];
    [self.view addGestureRecognizer:singleGestureRecognizer];

    [singleGestureRecognizer requireGestureRecognizerToFail:doubleGestureRecognizer];
}

- (void)tapAction:(UITapGestureRecognizer *)gestureRecognizer {
    /** 播放广告中，点击屏幕跳转广告链接 */
    if (self.playerVC.advPlaying) {
        if ([PLVFdUtil checkStringUseable:self.playerVC.advLinkUrl]) {
            [[UIApplication sharedApplication] openURL:[NSURL URLWithString:self.playerVC.advLinkUrl]];
        }
        return;
    }

    if (gestureRecognizer.numberOfTapsRequired == 1) {
        if (! self.playerVC.playing) {
            [self.playerVC play];
        }
    } else if (gestureRecognizer.numberOfTapsRequired == 2) {
        if (self.playerVC.playing) {
            [self.playerVC pause];
        } else {
            [self.playerVC play];
        }
    }
}
```

3.2 回放控制栏PLVECPlayerContolView

回放播放器控制栏是通过叠加在播放器布局上层封装的`PLVECPlayerContolView`来实现的。

代码如下：

```
PLVECPlayerContolView.m

- (void)playButtonAction:(UIButton *)sender {
    sender.selected = !sender.isSelected;
    if ([self.delegate respondsToSelector:@selector(playerContolView:switchPause:)]) {
        [self.delegate playerContolView:self switchPause:!sender.isSelected];
    }
}

- (void)sliderTouchDownAction:(UISlider *)sender {
    self.sliderDragging = YES;
}

- (void)sliderTouchEndAction:(UISlider *)sender {
    if (self.delegate && [self.delegate respondsToSelector:@selector(playerContolViewSeeking:)]) {
        [self.delegate playerContolViewSeeking:self];
    }
    self.sliderDragging = NO;
}

- (void)sliderValueChangedAction:(UISlider *)sender {
    if (self.duration == 0.0) {
        sender.value = 0.0;
    }
}
```

`PLVECPlayerContolView` 在 `PLVECPalybackHomePageView` 中实现并与其通讯。

代码如下：

```
PLVECPalybackHomePageView.m

- (void)playerContolView:(PLVECPlayerContolView *)playerContolView switchPause:(BOOL)pause {
    if ([self.delegate respondsToSelector:@selector(homePageView:switchPause:)]) {
        [self.delegate homePageView:self switchPause:pause];
    }
}

- (void)playerContolViewSeeking:(PLVECPlayerContolView *)playerContolView {
    NSTimeInterval interval = self.duration * playerContolView.progressSlider.value;

    // 拖动进度条后，同步当前进度时间
    [self updateDowloadProgress:0
                 playedProgress:playerContolView.progressSlider.value
            currentPlaybackTime:[PLVFdUtil secondsToString:interval]
                       duration:self.playerContolView.totalTimeLabel.text];

    if ([self.delegate respondsToSelector:@selector(homePageView:seekToTime:)]) {
        [self.delegate homePageView:self seekToTime:interval];
    }
}
```

`PLVECPalybackHomePageView` 在 `PLVECWatchRoomViewController` 中实现并与其通讯。

代码如下：

PLVECWatchRoomViewController.m

```
- (void)homePageView:(PLVECPalybackHomePageView *)homePageView switchPause:(BOOL)pause {
    /** 播放广告中，点击屏幕跳转广告链接 */
    if (self.playerVC.advPlaying) {
        if ([PLVFdUtil checkStringUseable:self.playerVC.advLinkUrl]) {
            [[UIApplication sharedApplication] openURL:[NSURL URLWithString:self.playerVC.advLinkUrl]];
        }
        return;
    }

    if (pause) {
        [self.playerVC pause];
    } else {
        [self.playerVC play];
    }
}

- (void)homePageView:(PLVECPalybackHomePageView *)homePageView seekToTime:(NSTimeInterval)time {
    [self.playerVC seek:time];
}

- (void)homePageView:(PLVECPalybackHomePageView *)homePageView switchSpeed:(CGFloat)speed {
    [self.playerVC speedRate:speed];
}

- (void)palyback_homePageView:(PLVECPalybackHomePageView *)homePageView receiveBulletinMessage:(NSString
*)content open:(BOOL)open {
    dispatch_async(dispatch_get_main_queue(), ^{
        if (open) {
            [self.liveDetailPageView addBulletinCardView:content];
        } else {
            [self.liveDetailPageView removeBulletinCardView];
        }
    });
}
```