**ijkplayer自定义播放器教程**

**作者：arthur.Liu**

[制作ijkplayer静态库 2](#_Toc522289611)

[相关知识点 2](#_Toc522289612)

[typedef 2](#_Toc522289613)

[UITapGestureRecognizer 2](#_Toc522289614)

[synchronized 3](#_Toc522289615)

[dispatch\_once 3](#_Toc522289616)

[Block 4](#_Toc522289617)

[BlockSlider 4](#_Toc522289618)

[BlockButton 5](#_Toc522289619)

[单例 6](#_Toc522289620)

[自定义播放器控件 12](#_Toc522289621)

[播放/暂停 12](#_Toc522289622)

[大小屏切换 13](#_Toc522289623)

[播放进度/拖动进度 14](#_Toc522289624)

[播放时间/总时间 14](#_Toc522289625)

[播放控件定时显示 15](#_Toc522289626)

# 制作ijkplayer静态库

参考：<https://www.jianshu.com/p/6990355cd1df>

# 相关知识点

## typedef

作用：给类型起别名（给已知的类型起别名）。常用于简化复杂类型，变量类型意义化等。

typedef double NSTimeInterval; //给double取别名为NSTimeInterval（变量类型意义化）

typedef NSTimeInterval MyTime; //给NSTimeInterval取别名为MyTime

typedef char \* MyString; //给char \*取别名为MyString

typedef struct Person

{

char \*name

}MyPerson; //给Person结构体取别名为MyPerson。使用:MyPerson p = {"jack"};

typedef enum Gender

{

Man,

Woman

}MyGender; //给Gender枚举取别名为MyGender。使用:MyGender g = Man;

typedef void(^MyBlock) (int a,int b); //给block取别名MyBlock

typedef int(\*MyFunction) (int a,int b); //给指向函数的指针取别名MyFunction

## UITapGestureRecognizer

## synchronized

英 ['sɪŋkrənaɪzd]

代表这个方法加锁, 相当于不管哪一个线程（例如线程A），运行到这个方法时,都要检查有没有其它线程例如B正在用这个方法，有的话要等正在使用synchronized方法的线程B运行完这个方法后再运行此线程A,没有的话,直接运行。它包括两种用法：synchronized 方法和 synchronized 块

使用synchronized方法实现：

static id obj = nil;

+(instancetype)shareInstance

{

@synchronized(self) {

if (!obj) {

obj = [[SingletonObj alloc] init];

}

}

return obj;

}

## dispatch\_once

*dispatch\_once*的作用正如其名：对于某个任务执行一次，且只执行一次。 *dispatch\_once*函数有两个参数，第一个参数*predicate*用来保证执行一次，第二个参数是要执行一次的任务block。

|  |  |
| --- | --- |
| 1  2  3  4 | *static* *dispatch\_once\_t* *predicate;*  *dispatch\_once(&predicate, ^{*  *// some one-time task*  *});* |

*dispatch\_once*被广泛使用在单例、缓存等代码中，用以保证在初始化时执行一次某任务。

使用dispatch\_once方法实现：

static id obj = nil;

+(instancetype)shareInstance

{

static dispatch\_once\_t onceToken;

dispatch\_once(&onceToken, ^{

obj = [[SingletonObj alloc] init];

});

return obj;

}

## Block

Objective-C 中 Block 有三种类型：

NSStackBlock 存储于栈区

NSGlobalBlock 存储于程序数据区

NSMallocBlock 存储于堆区

## BlockSlider

#import <UIKit/UIKit.h>

typedef void(^sBlock)(UISlider \* slider);

@interface BlockSlider : UISlider

@property (nonatomic, copy) sBlock block;

@end

#import "BlockSlider.h"

@implementation BlockSlider

- (instancetype)initWithFrame:(CGRect)frame

{

self = [super initWithFrame:frame];

if (self) {

[self addTarget:self action:@selector(doAction:) forControlEvents:UIControlEventValueChanged];

}

return self;

}

- (void)doAction:(UISlider \*)slider {

self.block(slider);

}

@end

## BlockButton

#import <UIKit/UIKit.h>

typedef void(^Block)(UIButton \* button);

@interface BlockButton : UIButton

@property (nonatomic, copy) Block block;

@end

#import "BlockButton.h"

@implementation BlockButton

- (instancetype)initWithFrame:(CGRect)frame

{

self = [super initWithFrame:frame];

if (self) {

[self addTarget:self action:@selector(doAction:) forControlEvents:UIControlEventTouchUpInside];

}

return self;

}

- (void)doAction:(UIButton \*)button {

self.block(button);

}

@end

## 单例

#import <UIKit/UIKit.h>

@interface PMSingleton : UIViewController

+ (PMSingleton \*)shareInstance;

@end

#define SCREEN\_WIDTH ([[UIScreen mainScreen] bounds].size.width)

#define SCREEN\_HEIGHT ([[UIScreen mainScreen] bounds].size.height)

#import "PMSingleton.h"

#import "IJKMediaFramework/IJKMediaFramework.h"

#import "BlockButton.h"

#import "BlockSlider.h"

@interface PMSingleton ()

@property (atomic, retain) id <IJKMediaPlayback> player;

@property (weak, nonatomic) UIView \*playView;

@property (weak, nonatomic) UIButton \*playBtn;

@property (nonatomic) UILabel \*nowlabel;

@property (nonatomic)NSTimer\* timer;

@end

@implementation PMSingleton

static PMSingleton \* pm\_singleton = nil;

#define SCREEN\_WIDTH ([[UIScreen mainScreen] bounds].size.width)

#define SCREEN\_HEIGHT ([[UIScreen mainScreen] bounds].size.height)

#import "PMSingleton.h"

#import "IJKMediaFramework/IJKMediaFramework.h"

#import "BlockButton.h"

#import "BlockSlider.h"

@interface PMSingleton ()

@property (atomic, retain) id <IJKMediaPlayback> player;

@property (weak, nonatomic) UIView \*playView;

@property (weak, nonatomic) UIButton \*playBtn;

@property (nonatomic) UILabel \*nowlabel;

@property (nonatomic)NSTimer\* timer;

@property (nonatomic)UIView \*bottomview;

@end

@implementation PMSingleton

static PMSingleton \* pm\_singleton = nil;

#pragma mark 单例模式

+(PMSingleton \*)shareInstance{

static dispatch\_once\_t onceToken;

dispatch\_once(&onceToken, ^{

pm\_singleton = [[super allocWithZone:nil] init];

});

return pm\_singleton;

}

+(id)allocWithZone:(NSZone \*)zone{

return [PMSingleton shareInstance];

}

-(id)copyWithZone:(NSZone \*)zone{

return [PMSingleton shareInstance];

}

-(id)mutableCopyWithZone:(NSZone \*)zone{

return [PMSingleton shareInstance];

}

#pragma mark 界面布局

- (void)viewDidLoad {

[super viewDidLoad];

// Do any additional setup after loading the view.

//

self.view.frame=CGRectMake(0, 0, self.view.frame.size.width, self.view.frame.size.width/2);

NSURL \*url = [NSURL URLWithString:@"rtmp://localhost:1935/myapp/wdqk.mp4"];

\_player = [[IJKFFMoviePlayerController alloc] initWithContentURL:url withOptions:nil];

UIView \*playerView = [\_player view];

playerView.frame = CGRectMake(0, 0, self.view.frame.size.width, self.view.frame.size.width/2);

[self.view addSubview:playerView];

[self.player prepareToPlay];

[self.player play];

[self createbottomview];

self.timer= [NSTimer scheduledTimerWithTimeInterval:1 target:self selector:@selector(updatetime) userInfo:nil repeats:YES];

UITapGestureRecognizer \*tapGesturRecognizer=[[UITapGestureRecognizer alloc]initWithTarget:self action:@selector(tapAction:)];

[playerView addGestureRecognizer:tapGesturRecognizer];

[self performSelector:@selector(hidebottom) withObject:nil afterDelay:4];

}

-(void)updatetime

{

self.nowlabel.text =[self TimeformatFromSeconds:self.player.currentPlaybackTime];

}

-(void)tapAction:(id)tap

{

self.bottomview.hidden=NO;

[self performSelector:@selector(hidebottom) withObject:nil afterDelay:4];

NSLog(@"点击了playerView");

}

-(void)hidebottom

{

self.bottomview.hidden=YES;

}

-(void)createbottomview

{

self.bottomview=[[UIView alloc]initWithFrame:CGRectMake(0, self.view.frame.size.height-50, self.view.frame.size.width, 50)];

self.bottomview.backgroundColor=[UIColor colorWithPatternImage:[UIImage imageNamed:@"pbg.png"]];

self.bottomview.layer.cornerRadius = 8;

self.bottomview.layer.masksToBounds = YES;

//设置边框及边框颜色

self.bottomview.layer.borderWidth = 3;

self.bottomview.layer.borderColor =[ [UIColor grayColor] CGColor];

BlockButton \*startbutton=[[BlockButton alloc]initWithFrame:CGRectMake(0, 0, 50, 50)];

startbutton.block = ^(UIButton \* button) {

NSLog(@"%@",button);

if (![self.player isPlaying]) {

[self.player play];

[button setTitle:@"Pause" forState:UIControlStateNormal];

[button setImage:[UIImage imageNamed:@"pause.png"] forState:UIControlStateNormal];

self.timer=[NSTimer scheduledTimerWithTimeInterval:1 target:self selector:@selector(updatetime) userInfo:nil repeats:YES];

}else {

[self.player pause];

[button setTitle:@"Play" forState:UIControlStateNormal];

[button setImage:[UIImage imageNamed:@"start.png"] forState:UIControlStateNormal];

[self.timer invalidate];

}

};

BlockSlider \*progress=[[BlockSlider alloc]initWithFrame:CGRectMake(50, 10, self.view.frame.size.width-100, 30)];

progress.block=^(UISlider \* slider) {

NSLog(@"now--- %f",slider.value);

self.player.currentPlaybackTime = slider.value\* self.player.duration;

self.nowlabel.text =[self TimeformatFromSeconds:self.player.currentPlaybackTime];

};

BlockButton \*allscreembutton=[[BlockButton alloc]initWithFrame:CGRectMake(self.view.frame.size.width-50, 0, 50, 50)];

[startbutton setImage:[UIImage imageNamed:@"start.png"] forState:UIControlStateNormal];

[allscreembutton setImage:[UIImage imageNamed:@"allscreen.png"] forState:UIControlStateNormal];

static BOOL isallscreen=NO;

allscreembutton.block = ^(UIButton \* button) {

self.player.view.center=CGPointMake(SCREEN\_WIDTH/2, SCREEN\_HEIGHT/2);//找到中心

if(!isallscreen){

isallscreen=YES;

CGAffineTransform transform = CGAffineTransformMakeRotation(M\_PI/2);

[self.player.view setTransform:transform];

[self.bottomview setTransform:transform];

self.view.frame=CGRectMake(0, 0, SCREEN\_WIDTH, SCREEN\_HEIGHT);

self.player.view.frame=CGRectMake(0, 0, SCREEN\_WIDTH, SCREEN\_HEIGHT);

self.bottomview.center=CGPointMake(50/2, SCREEN\_HEIGHT/2);

}else

{

isallscreen=NO;

// CGAffineTransform transform = CGAffineTransformMakeRotation(2\*M\_PI);

[self.player.view setTransform: CGAffineTransformIdentity];

[self.bottomview setTransform:CGAffineTransformIdentity];

self.view.frame=CGRectMake(0, 0, SCREEN\_WIDTH, SCREEN\_WIDTH/2);

self.player.view.frame=CGRectMake(0, 0, SCREEN\_WIDTH, SCREEN\_WIDTH/2);

self.bottomview.center=CGPointMake(SCREEN\_WIDTH/2, SCREEN\_WIDTH/2-50/2);

}

};

UILabel \*timelabel=[[UILabel alloc]initWithFrame:CGRectMake(self.view.frame.size.width-50-55, 25, 120, 20)];

timelabel.font=[UIFont systemFontOfSize:12];

timelabel.textColor=[UIColor lightGrayColor];

timelabel.text =[NSString stringWithFormat:@"%@",[self TimeformatFromSeconds:self.player.duration]];

self.nowlabel=[[UILabel alloc]initWithFrame:CGRectMake(self.view.frame.size.width-50-55, 5, 120, 20)];

self.nowlabel.font=[UIFont systemFontOfSize:12];

self.nowlabel.textColor=[UIColor lightGrayColor];

self.nowlabel.text=@"00:06:56";

[self.bottomview addSubview:startbutton];

[self.bottomview addSubview:self.nowlabel];

[self.bottomview addSubview:timelabel];

[self.bottomview addSubview:progress];

[self.bottomview addSubview:allscreembutton];

[self.view addSubview:self.bottomview];

}

#pragma mark-公共方法

- (NSString\*)TimeformatFromSeconds:(NSInteger)seconds

{

//format of hour

NSString \*str\_hour = [NSString stringWithFormat:@"%02ld",seconds/3600];

//format of minute

NSString \*str\_minute = [NSString stringWithFormat:@"%02ld",(seconds%3600)/60];

//format of second

NSString \*str\_second = [NSString stringWithFormat:@"%02ld",seconds%60];

//format of time

NSString \*format\_time = [NSString stringWithFormat:@"%@:%@:%@",str\_hour,str\_minute,str\_second];

return format\_time;

}

-(IBAction)clickedAction:(id)sender {

if (![\_player isPlaying]) {

[self.player play];

[self.playBtn setTitle:@"Pause" forState:UIControlStateNormal];

}else {

[self.player pause];

[self.playBtn setTitle:@"Play" forState:UIControlStateNormal];

}

}

- (void)didReceiveMemoryWarning {

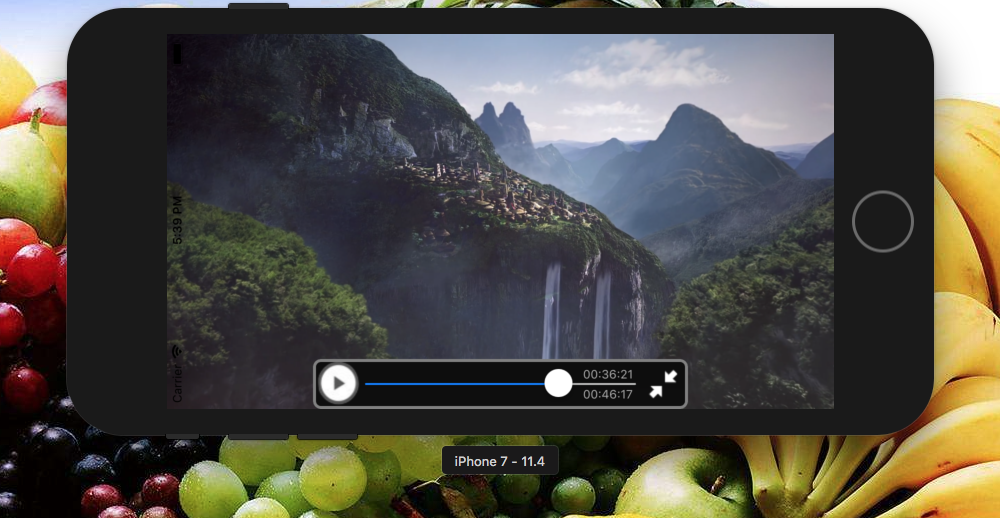
[super didReceiveMemoryWarning];

// Dispose of any resources that can be recreated.

}

@end

# 自定义播放器控件



## 播放/暂停

BlockButton \*startbutton=[[BlockButton alloc]initWithFrame:CGRectMake(0, 0, 50, 50)];

startbutton.block = ^(UIButton \* button) {

NSLog(@"%@",button);

if (![self.player isPlaying]) {

[self.player play];

[button setTitle:@"Pause" forState:UIControlStateNormal];

[button setImage:[UIImage imageNamed:@"pause.png"] forState:UIControlStateNormal];

self.timer=[NSTimer scheduledTimerWithTimeInterval:1 target:self selector:@selector(updatetime) userInfo:nil repeats:YES];

}else {

[self.player pause];

[button setTitle:@"Play" forState:UIControlStateNormal];

[button setImage:[UIImage imageNamed:@"start.png"] forState:UIControlStateNormal];

[self.timer invalidate];

}

};

## 大小屏切换

BlockButton \*allscreembutton=[[BlockButton alloc]initWithFrame:CGRectMake(self.view.frame.size.width-50, 0, 50, 50)];

[startbutton setImage:[UIImage imageNamed:@"start.png"] forState:UIControlStateNormal];

[allscreembutton setImage:[UIImage imageNamed:@"allscreen.png"] forState:UIControlStateNormal];

static BOOL isallscreen=NO;

allscreembutton.block = ^(UIButton \* button) {

self.player.view.center=CGPointMake(SCREEN\_WIDTH/2, SCREEN\_HEIGHT/2);//找到中心

if(!isallscreen){

isallscreen=YES;

CGAffineTransform transform = CGAffineTransformMakeRotation(M\_PI/2);

[self.player.view setTransform:transform];

[bottomview setTransform:transform];

self.view.frame=CGRectMake(0, 0, SCREEN\_WIDTH, SCREEN\_HEIGHT);

self.player.view.frame=CGRectMake(0, 0, SCREEN\_WIDTH, SCREEN\_HEIGHT);

bottomview.center=CGPointMake(50/2, SCREEN\_HEIGHT/2);

}else

{

isallscreen=NO;

// CGAffineTransform transform = CGAffineTransformMakeRotation(2\*M\_PI);

[self.player.view setTransform: CGAffineTransformIdentity];

[bottomview setTransform:CGAffineTransformIdentity];

self.view.frame=CGRectMake(0, 0, SCREEN\_WIDTH, SCREEN\_WIDTH/2);

self.player.view.frame=CGRectMake(0, 0, SCREEN\_WIDTH, SCREEN\_WIDTH/2);

bottomview.center=CGPointMake(SCREEN\_WIDTH/2, SCREEN\_WIDTH/2-50/2);

}

};

## 播放进度/拖动进度

BlockSlider \*progress=[[BlockSlider alloc]initWithFrame:CGRectMake(50, 10, self.view.frame.size.width-100, 30)];

progress.block=^(UISlider \* slider) {

NSLog(@"now--- %f",slider.value);

self.player.currentPlaybackTime = slider.value\* self.player.duration;

self.nowlabel.text =[self TimeformatFromSeconds:self.player.currentPlaybackTime];

};

#pragma mark-公共方法

- (NSString\*)TimeformatFromSeconds:(NSInteger)seconds

{

//format of hour

NSString \*str\_hour = [NSString stringWithFormat:@"%02ld",seconds/3600];

//format of minute

NSString \*str\_minute = [NSString stringWithFormat:@"%02ld",(seconds%3600)/60];

//format of second

NSString \*str\_second = [NSString stringWithFormat:@"%02ld",seconds%60];

//format of time

NSString \*format\_time = [NSString stringWithFormat:@"%@:%@:%@",str\_hour,str\_minute,str\_second];

return format\_time;

}

## 播放时间/总时间

timelabel.text =[NSString stringWithFormat:@"%@",[self TimeformatFromSeconds:self.player.duration]];

self.nowlabel.text =[self TimeformatFromSeconds:self.player.currentPlaybackTime];

## 播放控件定时显示

UITapGestureRecognizer \*tapGesturRecognizer=[[UITapGestureRecognizer alloc]initWithTarget:self action:@selector(tapAction:)];

[playerView addGestureRecognizer:tapGesturRecognizer];

[self performSelector:@selector(hidebottom) withObject:nil afterDelay:4];

-(void)tapAction:(id)tap

{

self.bottomview.hidden=NO;

[self performSelector:@selector(hidebottom) withObject:nil afterDelay:4];

NSLog(@"点击了playerView");

}

-(void)hidebottom

{

self.bottomview.hidden=YES;

}