# Complete Code (IOS)

## 1. Game Scene

## Interface:

```
#import "cocos2d.h"
#import "SneakyJoystick.h"
#import "SneakyJoystickSkinnedBase.h"
#import "SneakyButton.h"
#import "SneakyButtonSkinnedBase.h"
#import "SimpleAudioEngine.h"
#import "GameOverLayer.h"
#import "GameStartLayer.h"

@interface GameScene : CCScene
-(id) initWithCharacter:(int)mode ;
@end

@interface GameLayer : CCLayerColor
{
   CCSprite *addedGamePictureFrame;
   SneakyJoystick *leftJoystick;

   SneakyButton *shootButton;

   CCLabelTTF * timeLabel;
   CCLabelTTF * highScoreLabel;
   NSInteger highScoreCount;
   NSInteger timeCount;
   CCSprite* rightHand,
         * leftHand;
   BOOL finishedGame;
   BOOL allowToCompareHighScore;
}
@property(nonatomic, assign)NSInteger timeCount;
@property(nonatomic, assign)NSInteger highScoreCount;
@property(nonatomic, retain)NSMutableArray* myScoreArray;
@property(nonatomic, assign)BOOL finishedGame;
@property(nonatomic, assign)BOOL allowToCompareHighScore;

-
(void)spriteSheetAnimationPunch:(CCSprite*)sprite:(NSString*)spriteName:(NSString*)spriteFirstFrame:(float
)animationDelay:(NSInteger)animationFrameCount:(float)scale:(NSInteger)x:(NSInteger)y;
-(void)countDown: (ccTime)dt;
-(void)addScore: (ccTime)dt;
-(void)addGamePictures:(NSInteger)characterFrameCount:(NSString *)spriteName:(float)x:(float)y:(float)scale;
-(void)initJoystick;
-(void)initShootButton;
-(void)update:(ccTime)deltaTime;
-(void)saveScore:(NSInteger)score;
@end

@interface GamePigLayer : GameLayer
```

```
-(void)changeThePiggyPics: (ccTime)dt;
@end

@interface GameFrogLayer : GameLayer
-(void)changeTheFrogPics: (ccTime)dt;
@end
```

# Implementation:

```
#import "GameScene.h"

@implementation GameScene
//the character loading depends on the mode
//mode == 1 : loading picture related to piggy
//mode == 2 : loading picture related to frog

- (id) initWithCharacter:(int)mode {
    self = [super init];
    if (self != nil) {

        //load pictures
        [[CCSpriteFrameCache sharedSpriteFrameCache]addSpriteFramesWithFile:@"gamePictures.plist"];
        CCSpriteBatchNode* gameSpriteSheet = [CCSpriteBatchNode batchNodeWithFile:@"gamePictures.png"];
        [self addChild:gameSpriteSheet];

        //load joystick
        [[CCSpriteFrameCache sharedSpriteFrameCache]addSpriteFramesWithFile:@"analog_stick.plist"];
        CCSpriteBatchNode* gameContorllerPic = [CCSpriteBatchNode
batchNodeWithFile:@"analog_stick.png"];
        [self addChild:gameContorllerPic];

        //load dpadstick
        [[CCSpriteFrameCache sharedSpriteFrameCache]addSpriteFramesWithFile:@"dpad_buttons.plist"];
        CCSpriteBatchNode* dpadControllerPic = [CCSpriteBatchNode
batchNodeWithFile:@"dpad_buttons.png"];
        [self addChild:dpadControllerPic];

        if(mode == 1)
        {
            //loading piggy
            [self addChild:[GamePigLayer node]];

            //which character did u choose

        }

        if(mode == 2)
        {
            //loading frog
            [self addChild:[GameFrogLayer node]];
        }
    }
```

```
    return self;
}

-(void)dealloc
{
        [super dealloc];
}
@end

@implementation GameLayer
@synthesize timeCount;
@synthesize highScoreCount;
@synthesize finishedGame,allowToCompareHighScore;

-(void)initJoystick {
    SneakyJoystickSkinnedBase *joystickBase = [[[SneakyJoystickSkinnedBase alloc] init] autorelease];
    joystickBase.backgroundSprite = [CCSprite spriteWithSpriteFrameName:@"analog_pad.png"];
    joystickBase.thumbSprite = [CCSprite spriteWithSpriteFrameName:@"analog_nub.png"];
    joystickBase.joystick = [[SneakyJoystick alloc] initWithRect: CGRectMake(0, 0, 128, 128)];
    joystickBase.position = ccp(55, 55);
    joystickBase.scale = 0.4f;
    [self addChild:joystickBase z:5];
    leftJoystick = [joystickBase.joystick retain];
}

-(void)initShootButton {
    CGRect shootButtonDimensions = CGRectMake(0, 0, 64, 64);
    CGPoint shootButtonPosition = ccp(270,55);

    SneakyButtonSkinnedBase *shootButtonBase = [[[SneakyButtonSkinnedBase alloc] init] autorelease];
    shootButtonBase.position = shootButtonPosition;
    shootButtonBase.defaultSprite = [CCSprite spriteWithSpriteFrameName:@"a_button_up.png"];
    shootButtonBase.activatedSprite = [CCSprite spriteWithSpriteFrameName:@"a_button_down.png"];
    shootButtonBase.pressSprite = [CCSprite spriteWithSpriteFrameName:@"a_button_down.png"];

    shootButtonBase.button = [[SneakyButton alloc] initWithRect:shootButtonDimensions];
    shootButtonBase.scale = 0.7f;
    shootButton = [shootButtonBase.button retain];
    shootButton.isToggleable = NO;
    [self addChild:shootButtonBase z:5];
}

-(void) update:(ccTime)deltaTime {
    CGPoint scaledVelocity = ccpMult(leftJoystick.velocity, 240);

    if((scaledVelocity.x < 0) && (shootButton.active == YES))
    {
        //hide the first pic
        leftHand.visible = false;

        [[SimpleAudioEngine sharedEngine] playEffect:@"PUNCH.wav"];
        [self spriteSheetAnimationPunch:leftHand:
                                @"BoxingGloveLeft%d.png":
```

```
                                @"BoxingGloveLeft1.png":
                                0.1f:
                                11:
                                0.8f:
                                120:
                                200];
        highScoreCount ++;
    }

    if((scaledVelocity.x > 0) && (shootButton.active == YES))
    {
        //hide the first pic
        rightHand.visible = false;

        [[SimpleAudioEngine sharedEngine] playEffect:@"PUNCH.wav"];
        [self spriteSheetAnimationPunch:rightHand:
                                @"BoxingGlove%d.png":
                                @"BoxingGlove1.png":
                                0.1f:
                                11:
                                0.8f:
                                200:
                                200];
        highScoreCount ++;
    }
}

-(id)init {
    self=[super initWithColor:ccc4(200, 200, 200, 255)];
    if (self!=nil) {
    }
    return self;
}

-
(void)spriteSheetAnimationPunch:(CCSprite*)sprite:(NSString*)spriteName:(NSString*)spriteFirstFrame:(float
)animationDelay:(NSInteger)animationFrameCount:(float)scale:(NSInteger)x:(NSInteger)y{

    NSMutableArray *animFrames = [NSMutableArray array];
    [animFrames removeAllObjects];

    for(int i=1; i<animationFrameCount;i++)
    {
        [animFrames addObject:[[CCSpriteFrameCache sharedSpriteFrameCache]spriteFrameByName:[NSString
stringWithFormat:spriteName,i]]];
    }

    CCAnimation* animation = [CCAnimation animationWithFrames:animFrames delay:animationDelay];
    sprite = [CCSprite spriteWithSpriteFrameName:spriteFirstFrame];
    sprite.position = ccp(x,y);
    [sprite setScale:scale];

    id action = [CCRepeat actionWithAction:[CCAnimate actionWithAnimation:animation
```

```
restoreOriginalFrame:NO] times:1];
    id actions = [CCSequence actions:action,[CCCallFuncN actionWithTarget:self
                                                                 selector:@selector(removeFrameActions:)],
                                                                nil];
    [sprite runAction:actions];
    [self addChild:sprite z:2];
}

-(void)removeFrameActions:(id)sender
{
    CCSprite* sprite = (CCSprite*)sender;
    [sprite removeFromParentAndCleanup:YES];

    leftHand.visible = true;
    rightHand.visible = true;
}

-(void)countDown:(ccTime)dt
{
    if (timeCount != 0)
    {
        [self removeChild:timeLabel cleanup:YES];
        timeCount = timeCount - 1;
        timeLabel = [CCLabelTTF labelWithString:[NSString stringWithFormat:@"Time: %d",timeCount]
dimensions:CGSizeMake(100, 100) alignment:CCTextAlignmentCenter fontName:@"Arial" fontSize:20];
        [timeLabel setColor:ccRED];
        [timeLabel setPosition:CGPointMake(280, 420)];
        [self addChild:timeLabel z:1];
    }
    else{
        //finish the game and decide where to go
        finishedGame = true;
        [self unscheduleUpdate];
        [self unschedule:@selector(countDown:)];
        [self unschedule:@selector(changeThePiggyPics:)];
        [self unschedule:@selector(changeTheFrogPics:)];
        [self unschedule:@selector(addScore:)];

        //set to true
        allowToCompareHighScore = true;
        NSUserDefaults *setting = [NSUserDefaults standardUserDefaults];
        [setting setObject:[NSNumber numberWithInteger:allowToCompareHighScore]
forKey:@"compareHighScore"];

        [self saveScore:highScoreCount];
        [self addChild:[GameOverLayer node] z:10];
    }
}

//add/compare and save the score
-(void)saveScore:(NSInteger)score
{
    NSUserDefaults *setting = [NSUserDefaults standardUserDefaults];
```

```
    [setting setObject:[NSNumber numberWithInteger:score] forKey:@"highScore"];
}

-(void)addScore: (ccTime)dt
{
   [self removeChild:highScoreLabel cleanup:YES];
   highScoreLabel = [CCLabelTTF labelWithString:[NSString stringWithFormat:@"Score:
%d",highScoreCount] dimensions:CGSizeMake(100, 100) alignment:CCTextAlignmentCenter
fontName:@"Arial" fontSize:20];
   [highScoreLabel setColor:ccRED];
   [highScoreLabel setPosition:CGPointMake(70, 420)];
   [self addChild:highScoreLabel z:1];
}

-(void)addGamePictures:(NSInteger)characterFrameCount:(NSString *)spriteName:(float)x:(float)y:(float)scale
{
   //characterFrameCount == use to change the sprite when time comes
   //spriteName == call with count
   addedGamePictureFrame = [CCSprite spriteWithSpriteFrameName:[NSString stringWithFormat:
                                          spriteName, characterFrameCount]];
   addedGamePictureFrame.scale = scale;
   addedGamePictureFrame.position = ccp(x, y);
   [self addChild:addedGamePictureFrame];
}

- (void) dealloc
{
   [_myScoreArray release];
        [super dealloc];
}
@end

@implementation GamePigLayer
-(id)init
{
   if((self = [super initWithColor:ccc4(0, 0, 0, 255)]))
   {
      [[[CCDirector sharedDirector] openGLView] setMultipleTouchEnabled:YES];

      allowToCompareHighScore = false;
      [[NSUserDefaults standardUserDefaults]setBool:allowToCompareHighScore
forKey:@"compareHighScore"];

      //music
      [self initJoystick];
      [self initShootButton];
      [[SimpleAudioEngine sharedEngine] preloadEffect:@"PUNCH.wav"];
      [self scheduleUpdate];

      //boxing pictures
      rightHand = [CCSprite spriteWithSpriteFrameName: @"BoxingGlove1.png"];
      rightHand.position = ccp(200,180);
      [rightHand setScale:0.8f];
```

```
        [self addChild:rightHand z:2];

        leftHand = [CCSprite spriteWithSpriteFrameName: @"BoxingGloveLeft1.png"];
        leftHand.position = ccp(120,180);
        [leftHand setScale:0.8f];
        [self addChild:leftHand z:2];

        //add pause layer
        [self addChild:[GameStartLayer node] z:10];

        //add character pictures
        [self addGamePictures:0 :@"piggy%d.png" :170 :250: 1.0f];

        //timer
        //game finished when count to zero
        timeCount = 20;
        timeLabel = [CCLabelTTF labelWithString:[NSString stringWithFormat:@"Time: %d",timeCount]
dimensions:CGSizeMake(100, 100) alignment:CCTextAlignmentCenter fontName:@"Arial" fontSize:20];
        [timeLabel setColor:ccRED];
        [timeLabel setPosition:CGPointMake(280, 420)];
        [self addChild:timeLabel z:1];

        //high score
        highScoreCount = 0;
        highScoreLabel = [CCLabelTTF labelWithString:[NSString stringWithFormat:@"Score:
%d",highScoreCount] dimensions:CGSizeMake(100, 100) alignment:CCTextAlignmentCenter
fontName:@"Arial" fontSize:20];
        [highScoreLabel setColor:ccRED];
        [highScoreLabel setPosition:CGPointMake(70, 420)];
        [self addChild:highScoreLabel z:1];

        [self schedule:@selector(countDown:) interval:1.0];
        [self schedule:@selector(addScore:)  interval:1.0];
        [self schedule:@selector(changeThePiggyPics:) interval:1.0];
    }
    return self;
}

//change the picture of piggy
-(void)changeThePiggyPics: (ccTime)dt
{
    if(highScoreCount > 10)
    {
        [self removeChild:addedGamePictureFrame cleanup:YES];
        [self addGamePictures:1 :@"piggy%d.png" :170 :250: 1.0f];
    }

    if(highScoreCount > 50)
    {
        [self removeChild:addedGamePictureFrame cleanup:YES];
        [self addGamePictures:2 :@"piggy%d.png" :170 :250: 1.0f];
    }
```

```
    if(highScoreCount > 80)
    {
        [self removeChild:addedGamePictureFrame cleanup:YES];
        [self addGamePictures:3 :@"piggy%d.png" :170 :250: 1.0f];
    }

    if(highScoreCount > 100)
    {
        [self removeChild:addedGamePictureFrame cleanup:YES];
        [self addGamePictures:4 :@"piggy%d.png" :170 :250: 1.0f];
    }

    if(highScoreCount > 200)
    {
        [self removeChild:addedGamePictureFrame cleanup:YES];
        [self addGamePictures:5 :@"piggy%d.png" :170 :250: 1.0f];
    }
}
@end

@implementation GameFrogLayer
-(id)init
{
    if((self = [super initWithColor:ccc4(255, 160, 0, 255)]))
    {
        [[[CCDirector sharedDirector] openGLView] setMultipleTouchEnabled:YES];

        allowToCompareHighScore = false;
        [[NSUserDefaults standardUserDefaults]setBool:allowToCompareHighScore
forKey:@"compareHighScore"];

        //music
        [self initJoystick];
        [self initShootButton];
        [[SimpleAudioEngine sharedEngine] preloadEffect:@"PUNCH.wav"];
        [self scheduleUpdate];

        //boxing pictures
        rightHand = [CCSprite spriteWithSpriteFrameName: @"BoxingGlove1.png"];
        rightHand.position = ccp(200,180);
        [rightHand setScale:0.8f];
        [self addChild:rightHand z:2];

        leftHand = [CCSprite spriteWithSpriteFrameName: @"BoxingGloveLeft1.png"];
        leftHand.position = ccp(120,180);
        [leftHand setScale:0.8f];
        [self addChild:leftHand z:2];

        //add pause layer
        [self addChild:[GameStartLayer node] z:10];

        //add character pictures
        [self addGamePictures:0 :@"frog%d.png" :170 :250: 0.8f];
```

```
        //timer
        //game finished when count to zero
        timeCount = 20;
        timeLabel = [CCLabelTTF labelWithString:[NSString stringWithFormat:@"Time: %d",timeCount]
dimensions:CGSizeMake(100, 100) alignment:CCTextAlignmentCenter fontName:@"Arial" fontSize:20];
        [timeLabel setColor:ccRED];
        [timeLabel setPosition:CGPointMake(280, 420)];
        [self addChild:timeLabel z:1];

        //high score
        highScoreCount = 0;
        highScoreLabel = [CCLabelTTF labelWithString:[NSString stringWithFormat:@"Score:
%d",highScoreCount] dimensions:CGSizeMake(100, 100) alignment:CCTextAlignmentCenter
fontName:@"Arial" fontSize:20];
        [highScoreLabel setColor:ccRED];
        [highScoreLabel setPosition:CGPointMake(40, 420)];
        [self addChild:highScoreLabel z:1];


        [self schedule:@selector(countDown:) interval:1.0];
        [self schedule:@selector(addScore:) interval:1.0];
        [self schedule:@selector(changeTheFrogPics:) interval:1.0];
    }
    return self;
}

//change the picture of piggy
-(void)changeTheFrogPics: (ccTime)dt
{
    if(highScoreCount > 10)
    {
        [self removeChild:addedGamePictureFrame cleanup:YES];
        [self addGamePictures:1 :@"frog%d.png" :170 :250: 0.8f];
    }

    if(highScoreCount > 50)
    {
        [self removeChild:addedGamePictureFrame cleanup:YES];
        [self addGamePictures:2 :@"frog%d.png" :170 :250: 0.8f];
    }

    if(highScoreCount > 80)
    {
        [self removeChild:addedGamePictureFrame cleanup:YES];
        [self addGamePictures:3 :@"frog%d.png" :170 :250: 0.8f];
    }

    if(highScoreCount > 100)
    {
        [self removeChild:addedGamePictureFrame cleanup:YES];
        [self addGamePictures:4 :@"frog%d.png" :170 :250: 0.8f];
    }
```

```
    if(highScoreCount > 200)
    {
        [self removeChild:addedGamePictureFrame cleanup:YES];
        [self addGamePictures:5 :@"frog%d.png" :170 :250: 0.8f];
    }
}
@end
```

# 2. Main Menu Scene Interface:

```
#import "cocos2d.h"
#import "HighScoreScene.h"
#import "CreditScene.h"
#import "OptionScene.h"
#import "GameScene.h"

@interface MainMenuScene : CCScene
{

}
@end

@interface MainMenuLayer : CCLayer
{

}
-(void)showCharacterPick;
@end
```

# Implementation:

```
#import "MainMenuScene.h"

@implementation MainMenuScene

- (id) init {
    self = [super init];
    if (self != nil) {

        [self addChild:[MainMenuLayer node]];

    }
    return self;
}

-(void)dealloc
{
        [super dealloc];
}
```

```
@end

@implementation MainMenuLayer
- (id) init {
   if ((self = [super init])) {

          isTouchEnabled_ = YES;

      //play the music
      NSUserDefaults *usrDef = [NSUserDefaults standardUserDefaults];
      if([usrDef boolForKey:@"music"] == YES)
      {
         [[SimpleAudioEngine sharedEngine]playBackgroundMusic:@"Sonata.mp3" loop:YES];
         [[SimpleAudioEngine sharedEngine]setEffectsVolume:0.5f];
      }

      //set background for main menu
          CCSprite * background = [CCSprite spriteWithSpriteFrameName:@"MainMenu.png"];
      [background setPosition:ccp(160,240)];
          [self addChild:background];

      //font
          CCLabelBMFont * newgameLabel = [CCLabelBMFont labelWithString:@"NEW GAME"
                                              fntFile:@"hud_font.fnt"];
      CCLabelBMFont * highScoreLabel = [CCLabelBMFont labelWithString:@"HIGH SCORES"
                                               fntFile:@"hud_font.fnt"];
      CCLabelBMFont * creditLabel = [CCLabelBMFont labelWithString:@"CREDIT"
                                             fntFile:@"hud_font.fnt"];
      CCLabelBMFont * optionalLabel = [CCLabelBMFont labelWithString:@"OPTIONS"
                                               fntFile:@"hud_font.fnt"];
          [newgameLabel setColor:ccRED];
          [highScoreLabel setColor:ccRED];
      [creditLabel setColor:ccRED];
          [optionalLabel setColor:ccRED];

      //item label
          CCMenuItemLabel * newgame = [CCMenuItemLabel itemWithLabel:newgameLabel target:
                           self selector:@selector(newGame:)];
          CCMenuItemLabel * highscore = [CCMenuItemLabel itemWithLabel:highScoreLabel target:
                            self selector:@selector(highscore:)];
      CCMenuItemLabel * credit = [CCMenuItemLabel itemWithLabel:creditLabel target:
                          self selector:@selector(credit:)];
          CCMenuItemLabel * option = [CCMenuItemLabel itemWithLabel:optionalLabel target:
                          self selector:@selector(option:)];

      //menu
          CCMenu * menu = [CCMenu menuWithItems:newgame,highscore,option,credit,nil];
          [menu alignItemsVerticallyWithPadding:5];
          [self addChild:menu];
          [menu setPosition:ccp(480,160)];

          [newgame runAction:[CCSequence actions:
```

```
                                                [CCEaseOut actionWithAction:[CCMoveBy
actionWithDuration:1 position:ccp(-240,0)]  rate:2],
                                                [CCRepeat actionWithAction:[CCSequence actions:[CCScaleTo
actionWithDuration:1 scale:1.3],[CCScaleTo actionWithDuration:1 scale:1],nil] times:9000],
                                                nil]];
        [highscore runAction:[CCSequence actions:
                    [CCDelayTime actionWithDuration:1.0],[CCEaseOut actionWithAction:[CCMoveBy
actionWithDuration:1 position:ccp(-240,0)]  rate:2],
                    [CCRepeat actionWithAction:[CCSequence actions:[CCScaleTo actionWithDuration:1
scale:1.3],[CCScaleTo actionWithDuration:1 scale:1],nil] times:9000],
                    nil]];
      [credit runAction:[CCSequence actions:
                    [CCDelayTime actionWithDuration:2.0],[CCEaseOut actionWithAction:[CCMoveBy
actionWithDuration:1 position:ccp(-240,0)]  rate:2],
                    [CCRepeat actionWithAction:[CCSequence actions:[CCScaleTo actionWithDuration:1
scale:1.3],[CCScaleTo actionWithDuration:1 scale:1],nil] times:9000],
                    nil]];
        [option runAction:[CCSequence actions:
                    [CCDelayTime actionWithDuration:1.5],[CCEaseOut actionWithAction:[CCMoveBy
actionWithDuration:1 position:ccp(-240,0)]  rate:2],
                    [CCRepeat actionWithAction:[CCSequence actions:[CCScaleTo actionWithDuration:1
scale:1.3],[CCScaleTo actionWithDuration:1 scale:1],nil] times:9000],
                    nil]];

    }
    return self;
}

-(void)newGame:(id)sender
{
        [self showCharacterPick];
}

-(void)showCharacterPick
{
    //character pick layer z:1
        ccColor4B c = {0,0,0,180};
        CCLayerColor * characterPick = [CCLayerColor layerWithColor:c];
        [self addChild:characterPick z:1];

        CCMenuItemSprite * piggy = [CCMenuItemSprite itemFromNormalSprite:[CCSprite
                                spriteWithSpriteFrameName:@"Piggy_selected.png"] selectedSprite:[CCSprite
spriteWithSpriteFrameName:@"Piggy_selected.png"] target:self selector:@selector(selectMode:)];

        CCMenuItemSprite * frog = [CCMenuItemSprite itemFromNormalSprite:[CCSprite
                                  spriteWithSpriteFrameName:@"Frog_normal.png"]selectedSprite:[CCSprite
spriteWithSpriteFrameName:@"Frog_normal.png"] target:self selector:@selector(selectMode:)];

        [piggy setTag:1];
        [frog setTag:2];

        CCMenu * characterPickMenu = [CCMenu menuWithItems:piggy,frog,nil];
        [characterPickMenu alignItemsVerticallyWithPadding:10];
```

```
        [characterPick addChild:characterPickMenu];
}

-(void)selectMode:(CCMenuItemImage *)btn
{
        int mode = btn.tag;
        GameScene * gs = [[[GameScene alloc]initWithCharacter:mode]autorelease];
        [[CCDirector sharedDirector]replaceScene:gs];
}

-(void)highscore:(id)sender
{
        HighScoreScene * hs = [HighScoreScene node];
        [[CCDirector sharedDirector]replaceScene:hs];
}

-(void)credit:(id)sender
{
   CreditScene *cs = [CreditScene node];
   [[CCDirector sharedDirector]replaceScene:cs];
}

-(void)option:(id)sender
{
        OptionScene *os = [OptionScene node];
   [[CCDirector sharedDirector]replaceScene:os];
}

-(void)dealloc
{
        [super dealloc];
}
@end
```

# 3. HighScore Scene Interface:

```
#import <Foundation/Foundation.h>
#import "cocos2d.h"
#import "MainMenuScene.h"

@interface HighScoreScene : CCScene {

}
@end

@interface HighScoreLayer : CCLayer{
   NSMutableArray* _myScoreArray;
   NSInteger score;
}
@property(nonatomic,assign)NSInteger score;
@property(nonatomic,retain)NSMutableArray* myScoreArray;
```

```
-(NSInteger)getHighScore:(NSString*)keys;
-(BOOL)getCharacterChoose;
-(void)goBack:(id)sender;
@end
```

# Implementation:

```
#import "HighScoreScene.h"


@implementation HighScoreScene
- (id) init {
    self = [super init];
    if (self != nil) {

        [self addChild:[HighScoreLayer node]];

    }
    return self;
}

-(void)dealloc
{
        [super dealloc];
}
@end

@implementation HighScoreLayer
@synthesize myScoreArray = _myScoreArray;
@synthesize score;

- (id) init {
    if ((self = [super init])) {

        //background
        CCSprite *background = [[CCSprite alloc]initWithSpriteFrameName:@"HighScore_background.png"];
        [background setPosition:ccp(160,240)];
            [self addChild:background];

        //go back button
        CCMenuItemSprite * goback = [CCMenuItemSprite itemFromNormalSprite:[CCSprite
spriteWithSpriteFrameName:@"options_goback.png"] selectedSprite:[CCSprite
spriteWithSpriteFrameName:@"options_goback.png"] target:self
selector:@selector(goBack:)];

        [goback setPosition:ccp(160,30)];

        //menu
            CCMenu * menu = [CCMenu menuWithItems:goback,nil];
            [self addChild:menu];
            [menu setPosition:ccp(0,0)];
```

```
        //high score board
        [self highScoreBoard];
    }
    return self;
}

-(void)highScoreBoard{

    //compare with the score
    if([self getCharacterChoose] == true){
        if([self getHighScore:@"highScore"] > [self getHighScore:@"ScoreNoOne"])
        {
            NSUserDefaults *setting = [NSUserDefaults standardUserDefaults];
            [setting setObject:[NSNumber numberWithInteger:[self getHighScore:@"ScoreNoTwo"]]
forKey:@"ScoreNoThree"];
            [setting setObject:[NSNumber numberWithInteger:[self getHighScore:@"ScoreNoOne"]]
forKey:@"ScoreNoTwo"];
            [setting setObject:[NSNumber numberWithInteger:[self getHighScore:@"highScore"]]
forKey:@"ScoreNoOne"];
        }

        if(([self getHighScore:@"highScore"] < [self getHighScore:@"ScoreNoOne"])&&
            ([self getHighScore:@"highScore"] > [self getHighScore:@"ScoreNoTwo"]))
        {
            NSUserDefaults *setting = [NSUserDefaults standardUserDefaults];
            [setting setObject:[NSNumber numberWithInteger:[self getHighScore:@"ScoreNoTwo"]]
forKey:@"ScoreNoThree"];
            [setting setObject:[NSNumber numberWithInteger:[self getHighScore:@"highScore"]]
forKey:@"ScoreNoTwo"];
        }

        if(([self getHighScore:@"highScore"] < [self getHighScore:@"ScoreNoOne"])&&
            ([self getHighScore:@"highScore"] < [self getHighScore:@"ScoreNoTwo"])&&
            ([self getHighScore:@"highScore"] > [self getHighScore:@"ScoreNoThree"]))
        {
            NSUserDefaults *setting = [NSUserDefaults standardUserDefaults];
            [setting setObject:[NSNumber numberWithInteger:[self getHighScore:@"highScore"]]
forKey:@"ScoreNoThree"];
        }
    }


    CCLabelTTF* highScoreThree = [CCLabelTTF labelWithString:
                                  [NSString stringWithFormat:@"%@ _____ %d",
                                  @"No. 3",
                                  [self getHighScore:@"ScoreNoThree"]]
                                  dimensions:CGSizeMake(250, 100)
                                  alignment:CCTextAlignmentLeft
                                  fontName:@"Arial"
                                  fontSize:20];
    [highScoreThree setColor:ccRED];
    [highScoreThree setPosition:CGPointMake(150,250)];
    [self addChild:highScoreThree z:1];
```

```
    CCLabelTTF* highScoreTwo = [CCLabelTTF labelWithString:
                                [NSString stringWithFormat:@"%@ _____ %d",
                                @"No. 2",
                                [self getHighScore:@"ScoreNoTwo"]]
                                dimensions:CGSizeMake(250, 100)
                                alignment:CCTextAlignmentLeft
                                fontName:@"Arial"
                                fontSize:20];
    [highScoreTwo setColor:ccRED];
    [highScoreTwo setPosition:CGPointMake(150,300)];
    [self addChild:highScoreTwo z:1];

    CCLabelTTF* highScoreOne = [CCLabelTTF labelWithString:
                                [NSString stringWithFormat:@"%@ _____ %d",
                                @"No. 1",
                                [self getHighScore:@"ScoreNoOne"]]
                                dimensions:CGSizeMake(250, 100)
                                alignment:CCTextAlignmentLeft
                                fontName:@"Arial"
                                fontSize:20];
    [highScoreOne setColor:ccRED];
    [highScoreOne setPosition:CGPointMake(150,350)];
    [self addChild:highScoreOne z:1];
}

-(BOOL)getCharacterChoose{
    return [[NSUserDefaults standardUserDefaults]boolForKey:@"compareHighScore"];
}

-(NSInteger)getHighScore:(NSString*)keys{
    return [[NSUserDefaults standardUserDefaults]integerForKey:keys];
}

-(void)goBack:(id)sender
{
        MainMenuScene * gs = [MainMenuScene node];
        [[CCDirector sharedDirector]replaceScene:gs];
}

-(void)dealloc
{
    [_myScoreArray release];
        [super dealloc];
}

@end
```

# 4. Credit Scene:

```
#import <Foundation/Foundation.h>
#import "cocos2d.h"
```

```
#import "MainMenuScene.h"

@interface CreditScene : CCScene {

}

@end

@interface CreditLayer : CCLayer{

}

@end
```

# implementation:

```
#import "CreditScene.h"


@implementation CreditScene
- (id) init {
    self = [super init];
    if (self != nil) {

        [self addChild:[CreditLayer node]];

    }
    return self;
}

-(void)dealloc
{
        [super dealloc];
}
@end

@implementation CreditLayer

- (id) init {
    if ((self = [super init])) {

        isTouchEnabled_ = true;

        //background
        CCSprite *background = [[CCSprite alloc]initWithSpriteFrameName:@"credit_background.png"];
        [background setPosition:ccp(160,240)];
            [self addChild:background];

        //go back button
        CCMenuItemSprite * goback = [CCMenuItemImage itemFromNormalSprite:[CCSprite
spriteWithSpriteFrameName:@"options_goback.png"] selectedSprite:[CCSprite
spriteWithSpriteFrameName:@"options_goback.png"] target:self selector:@selector(goBack:)];
```

```
        [goback setPosition:ccp(160,30)];

        //menu
            CCMenu * menu = [CCMenu menuWithItems:goback,nil];
            [self addChild:menu z:2];
            [menu setPosition:ccp(0,0)];

    }
    return self;
}

-(void)goBack:(id)sender
{
        MainMenuScene * gs = [MainMenuScene node];
        [[CCDirector sharedDirector]replaceScene:gs];
}

-(void)dealloc
{
        [super dealloc];
}

@end
```

# 5. Option Scene Interface:

```
#import <Foundation/Foundation.h>
#import "cocos2d.h"
#import "MainMenuScene.h"
#import "SimpleAudioEngine.h"

@interface OptionScene : CCScene {

}

@end


@interface OptionLayer : CCLayer {

}

@end
```

# Implementation:

```
#import "OptionScene.h"
```

```
@implementation OptionScene

- (id) init {
    self = [super init];
    if (self != nil) {

        [self addChild:[OptionLayer node]];

    }
    return self;
}

-(void)dealloc
{
        [super dealloc];
}

@end

@implementation OptionLayer
- (id) init {
    if ((self = [super init])) {

            isTouchEnabled_ = YES;

        //background of option
            CCSprite * background = [CCSprite spriteWithSpriteFrameName:@"options_background.png"];
            [background setPosition:ccp(160,240)];
            [self addChild:background];

        //music and sound label
            CCLabelBMFont * musicLabel = [CCLabelBMFont labelWithString:@"MUSIC"
                                                fntFile:@"hud_font.fnt"];
            [musicLabel setColor:ccRED];
            [self addChild:musicLabel];
            [musicLabel setPosition:ccp(80,400)];

            CCLabelBMFont * soundLabel = [CCLabelBMFont labelWithString:@"SOUND"
                                                fntFile:@"hud_font.fnt"];
            [soundLabel setColor:ccRED];
            [self addChild:soundLabel];
            [soundLabel setPosition:ccp(80,300)];

            //option check and uncheck for music and sound
            CCMenuItemSprite * unchecked1 = [CCMenuItemSprite itemFromNormalSprite:[CCSprite
spriteWithSpriteFrameName:@"options_check.png"] selectedSprite:[CCSprite
spriteWithSpriteFrameName:@"options_check_d.png"]];

            CCMenuItemSprite * checked1 = [CCMenuItemSprite itemFromNormalSprite:[CCSprite
spriteWithSpriteFrameName:@"options_check_d.png"] selectedSprite:[CCSprite
spriteWithSpriteFrameName:@"options_check.png"]];
```

```
        CCMenuItemToggle * music = [CCMenuItemToggle itemWithTarget:self
selector:@selector(changeMusic:) items:unchecked1,checked1,nil];

        CCMenuItemSprite * unchecked2 = [CCMenuItemSprite itemFromNormalSprite:[CCSprite
spriteWithSpriteFrameName:@"options_check.png"] selectedSprite:[CCSprite
spriteWithSpriteFrameName:@"options_check_d.png"]];

        CCMenuItemSprite * checked2 = [CCMenuItemSprite itemFromNormalSprite:[CCSprite
spriteWithSpriteFrameName:@"options_check_d.png"] selectedSprite:[CCSprite
spriteWithSpriteFrameName:@"options_check.png"]];

        CCMenuItemToggle * sound = [CCMenuItemToggle itemWithTarget:self
selector:@selector(changeSound:) items:unchecked2,checked2,nil];

        CCMenuItemSprite * goback = [CCMenuItemSprite itemFromNormalSprite:[CCSprite
spriteWithSpriteFrameName:@"options_goback.png"] selectedSprite:[CCSprite
spriteWithSpriteFrameName:@"options_goback.png"] target:self selector:@selector(goBack:)];

        [music setPosition:ccp(220,410)];
        [sound setPosition:ccp(220,310)];
        [goback setPosition:ccp(160,30)];

        CCMenu * menu = [CCMenu menuWithItems:music,sound,goback,nil];
        [self addChild:menu];
        [menu setPosition:ccp(0,0)];

    //remember the selection for player
        NSUserDefaults *usrDef = [NSUserDefaults standardUserDefaults];
        if([usrDef boolForKey:@"sound"] == YES)
                sound.selectedIndex = 1;

        if([usrDef boolForKey:@"music"] == YES)
                music.selectedIndex = 1;
   }
   return self;
}

-(void)changeSound:(CCMenuItemToggle *)sender
{
        NSUserDefaults *usrDef = [NSUserDefaults standardUserDefaults];

        if(sender.selectedIndex ==1)
   {
      [SimpleAudioEngine sharedEngine].effectsVolume = 0.5;
        [usrDef setBool:YES forKey:@"sound"];
   }
        if(sender.selectedIndex ==0)
   {
      [SimpleAudioEngine sharedEngine].effectsVolume = 0;
        [usrDef setBool:NO forKey:@"sound"];
   }
}
```

```
-(void)changeMusic:(CCMenuItemToggle *)sender
{
        NSUserDefaults *usrDef = [NSUserDefaults standardUserDefaults];

        if(sender.selectedIndex ==1)
   {
      [SimpleAudioEngine sharedEngine].backgroundMusicVolume = 0.5;
         [usrDef setBool:YES forKey:@"music"];
   }
        if(sender.selectedIndex ==0)
   {
      [SimpleAudioEngine sharedEngine].backgroundMusicVolume = 0;
         [usrDef setBool:NO forKey:@"music"];
   }
}

-(void)goBack:(id)sender
{
        MainMenuScene * gs = [MainMenuScene node];
        [[CCDirector sharedDirector]replaceScene:gs];
}

-(void)dealloc
{
        [super dealloc];
}
@end
```

# 6. Splash Scene:
# Interface:

```
#import <Foundation/Foundation.h>
#import "cocos2d.h"
#import "MainMenuScene.h"

@interface SplashScene : CCScene {
   //parents of splashLayer
}
@end

@interface SplashLayer : CCLayer {
   //add sprites

}

-(void)fadeAndShow:(NSMutableArray *)images;
-(void) cFadeAndShow:(id)sender data:(void*)data;
-(void)remove:(CCSprite *)s;
@end
```

# implementation:

```
#import "SplashScene.h"


@implementation SplashScene
-(id)init
{
    self = [super init];
    if(self != nil)
    {
        //add layer into scene as a child
        [self addChild:[SplashLayer node]];
    }
    return self;
}

-(void)dealloc
{
    [super dealloc];
}
@end

//splash layer
@implementation SplashLayer
-(id)init
{
    if(self = [super init])
    {
        isTouchEnabled_ = YES;

        //import the spritesheet
        [[CCSpriteFrameCache sharedSpriteFrameCache]addSpriteFramesWithFile:@"mainMenu.plist"];
        CCSpriteBatchNode* menuSpriteSheet = [CCSpriteBatchNode batchNodeWithFile:@"mainMenu.png"];
        [self addChild:menuSpriteSheet];

        NSMutableArray * splashImages = [[NSMutableArray alloc]init];

        //right now only has one splash pic
        for(int i=1;i<3;i++)
        {
            CCSprite * splashImage = [CCSprite spriteWithSpriteFrameName:[NSString
stringWithFormat:@"mySplash%d.png",i]];
            CGSize size = [[CCDirector sharedDirector] winSize];
            [splashImage setPosition:ccp(size.width/2, size.height/2)];
            [splashImage setScale:1.0f];
            [self addChild:splashImage];

            if(i!=1)
                [splashImage setOpacity:0];
            [splashImages addObject: splashImage];
        }
        [self fadeAndShow:splashImages];
```

```
    }
    return self;
}

-(void)fadeAndShow:(NSMutableArray *)images
{
    if([images count]<=1)
    {
        [images release];
        [[CCDirector sharedDirector]replaceScene:[CCTransitionFade transitionWithDuration:3
scene:[MainMenuScene node]withColor:ccBLACK]];
    }
    else {
        CCSprite * actual = (CCSprite *)[images objectAtIndex:0];
        [images removeObjectAtIndex:0];
        CCSprite * next = (CCSprite *)[images objectAtIndex:0];
        [actual runAction:[CCSequence actions:[CCDelayTime actionWithDuration:3], [CCFadeOut
actionWithDuration:2],[CCCallFuncN actionWithTarget:self selector:@selector(remove:)],nil]];

        //cccallFunND is to call array again to make sure whether we have image in array
        [next runAction:[CCSequence actions:[CCDelayTime actionWithDuration:3], [CCFadeIn
actionWithDuration:2],[CCDelayTime actionWithDuration:3],[CCCallFuncND actionWithTarget:self
selector:@selector(cFadeAndShow:data:) data:images],nil]];
    }
}
-(void) cFadeAndShow:(id)sender data:(void*)data
{
    NSMutableArray * images = (NSMutableArray *)data;
    [self fadeAndShow:images];
}

-(void)remove:(CCSprite *)s
{
    [s.parent removeChild:s cleanup:YES];
}

-(void)dealloc
{

    [super dealloc];
}
@end
```

# 7. Game Start Layer:
# Interface:

```
#import <Foundation/Foundation.h>
#import "cocos2d.h"
#import "GameScene.h"

@interface GameStartLayer : CCLayer {

}
```

```
@end
```

# Implementation:

```
#import "GameStartLayer.h"

@implementation GameStartLayer
-(id)init
{
   if(self = [super init])
   {
      self.isTouchEnabled = YES;

      CCSprite* pauseView = [CCSprite spriteWithSpriteFrameName:@"StartGame.png"];
      [pauseView setPosition:ccp(320/2,480/2)];
      [pauseView setScale:0.8f];
      [self addChild:pauseView];

      //pause the game --- put into a function --
      [self PauseGame];
   }
   return self;
}

-(void)ccTouchesBegan:(NSSet *)touches withEvent:(UIEvent *)event
{
   [[CCDirector sharedDirector]resume];
   [self.parent removeChild:self cleanup:YES];
}

-(void)PauseGame
{
   [[CCDirector sharedDirector]pause];
}

- (void) dealloc
{
   [super dealloc];
}
@end
```

# 8. GameOver Layer:
# Interface:

```
#import <Foundation/Foundation.h>
#import "cocos2d.h"
#import "MainMenuScene.h"

@interface GameOverLayer : CCLayer
```

```
{

}
@end
```

# implementation:

```
#import "GameOverLayer.h"


@implementation GameOverLayer
-(id)init
{
    if(self = [super init])
    {
        self.isTouchEnabled = YES;

        CCSprite* GameOverView = [CCSprite spriteWithSpriteFrameName:@"GameOver.png"];
        [GameOverView setPosition:ccp(320/2,480/2)];
        [GameOverView setScale:0.8f];
        [self addChild:GameOverView];

    }
    return self;
}

-(void)ccTouchesBegan:(NSSet *)touches withEvent:(UIEvent *)event
{
    [self GameOver];
}

-(void)GameOver
{
    //touch to jump back to main menu
    MainMenuScene * Ms = [MainMenuScene node];
        [[CCDirector sharedDirector]replaceScene:Ms];
}

- (void) dealloc
{
    [super dealloc];
}
@end
```