SpriteKit in Xcode 7

David Corrigan

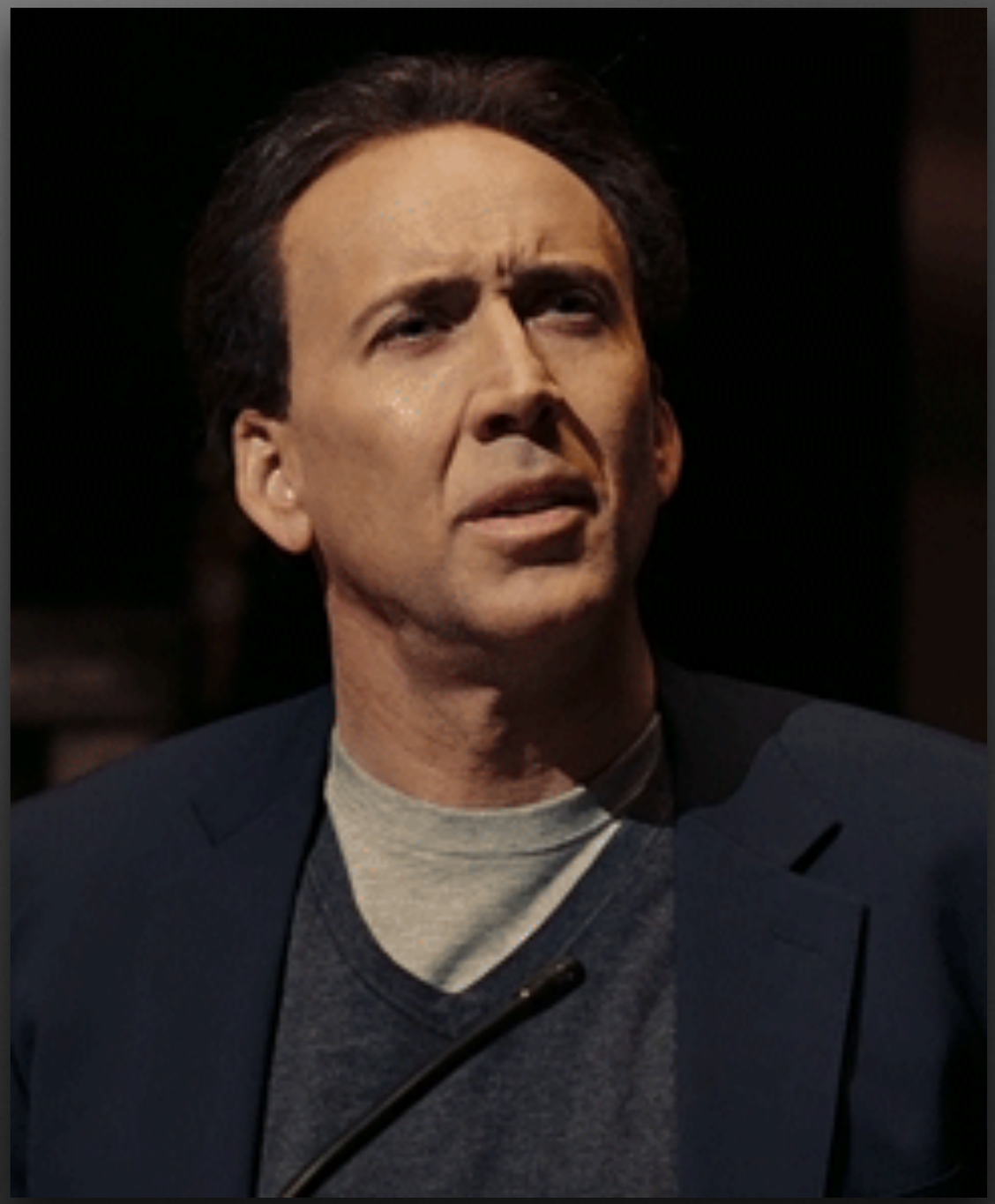
www.DigitalHeroGames.com

What is SpriteKit exactly?

- SpriteKit is a 2D Game Engine. It is a core Apple framework included in Xcode.
- It simplifies the complexity of drawing, animating, etc.
- Most game developers use a Game Engine and do not write their own. The tools are already there. Use them.
- Also great for making multimedia apps.

So where is the UI designer?

- There typically isn't a visual way to create your game.
- Most of the game is created in code.
- Draw it out on paper or on a story board and then start coding.
- Or just make it up as you go.

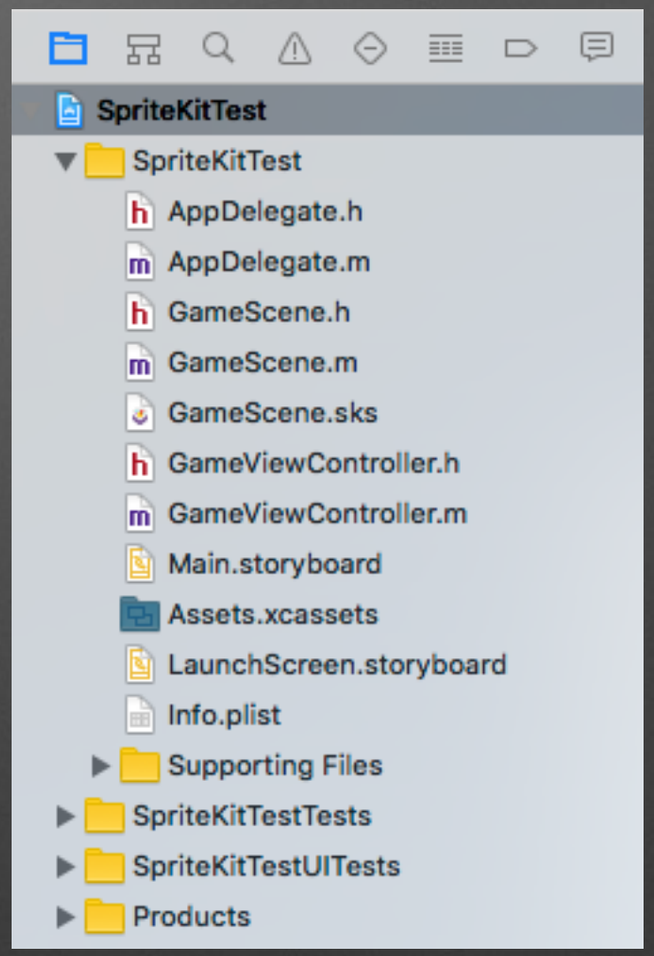


What are these files in the SpriteKit Project?

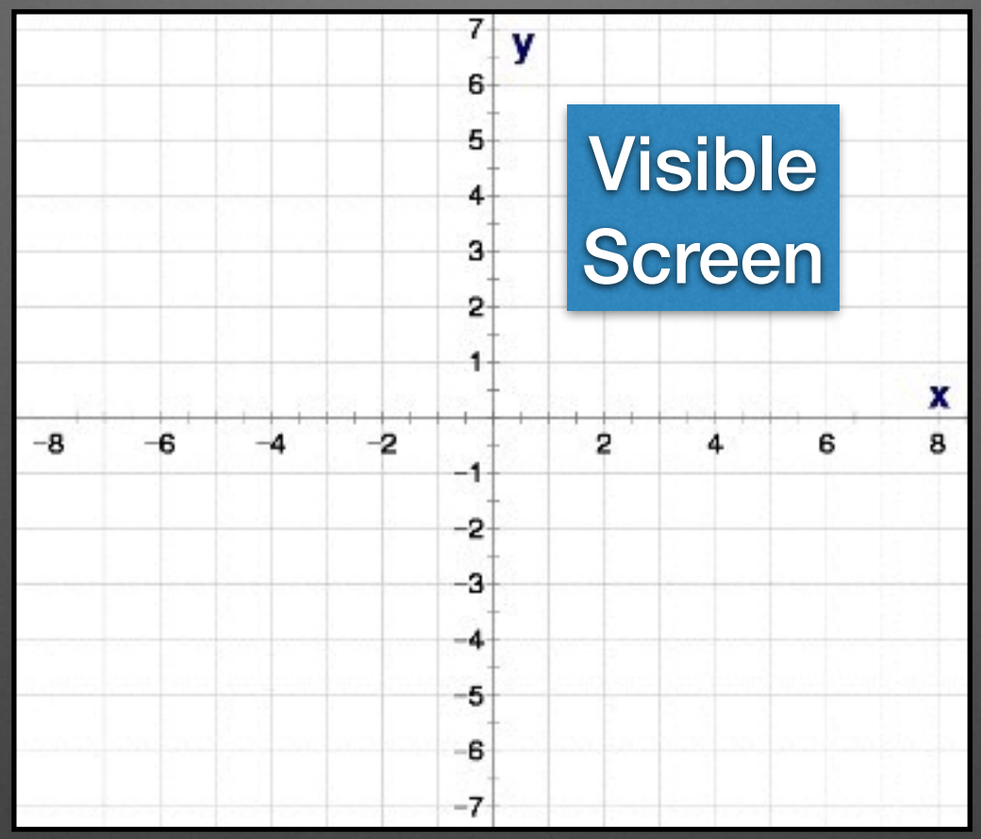
- GameScene - Basically this is your game. Think of it like a level or a stage that gets performed on.

- GameViewController - Like the Director of a movie. Responsible for loading a GameScene.

- AppDelegate - Responsible for starting and stopping your app. You don’t need to do much here.



You place and move graphics by their x,y screen coordinates



z-index is sometimes used for depth or layers on the screen so you can place an image on top of another one.

Don’t worry this is the extent of our math lesson.

The Main Game Loop

- The Main Game Loop or -(void)update function in your GameScene is where most of your game logic lives.
- FPS (Frames per Second) is how many times that Main Game Loop gets executes and updates the screen per second.
- Your game logic and screen draw calls happen every time the Main Game Loop is executed.
- 60 FPS is the gold standard. That allows the screen to update fast enough to look smooth and responsive.
- Doing too much in your Main Game Loop can slow down the entire game.

```
-(void)update:(CFTimeInterval)currentTime {
    /* Called before each frame is rendered */


    //check to see if we need to create a new asteroid
    if (asteroidCount <= 10 && !waitingOnAsteroid)
    {
        waitingOnAsteroid = YES;

        int xTime = [self getRandomNumberBetween:1 to:5];

        [self performSelector:@selector(createAsteroid) wi

    }

    //check to see if we need to move the spaceship
    if(isShipMovingRight){

        if(spaceship.position.x < self.size.width - spaces

            spaceship.position = CGPointMake(spaceship.pos
        }

    } else if (isShipMovingLeft){

        if(spaceship.position.x > spaceship.size.width/2){

            spaceship.position = CGPointMake(spaceship.pos
        }
}
```

Let’s write some code

Download the sample project

www.DigitalHeroGames.com/SpriteKitSample

Find me on Twitter @UberArcade

Free Artwork at: http://opengameart.org