

December 4th, 2018  
6D.ai Beta SDK v0.18.0 Release Notes

Find additional documentation on the 6D.ai beta developer portal  
<https://dashboard.6d.ai>

## Introduction

Happy Tuesday! Today we are introducing SDK version beta 0.18.0, bringing Metal 2.0 support on iOS and making it the default graphics API. OpenGL ES 3.0 will still be supported for compatibility, for as long as it is supported on iOS. Since it was deprecated by Apple earlier this year, we recommend adopting Metal.

The sample apps (Unity and SceneKit) have been updated to run on Metal, but can still be configured to run on OpenGL ES with a simple configuration change. Upgrading from 0.17.0 will be easy and in most cases require no changes, e.g. our Unity C# scripts will auto-detect Metal or OpenGL ES and call the corresponding 6D.ai SDK initialization method.

## Changes

### New API methods for Metal

- `void SixDegreesSDK_Initialize(void)`  
Initializes the 6D.ai SDK to run with Metal 2.0, the default and recommended graphics API for iOS.
- `void* SixDegreesSDK_GetBackgroundTexture(void)`  
Returns the `MTLTexture` pointer of the camera background. Will return `NULL` if the SDK is not initialized or was initialized with OpenGL ES. Follow the snippets below to go from `void*` to `MTLTexture`.

#### Swift:

```
if let texPtr = SixDegreesSDK_GetBackgroundTexture() {  
    let mtlTexture =  
    Unmanaged<MTLTexture>.fromOpaque(texPtr).takeUnretainedValue()  
}
```

#### Objective-C:

```
void* texPtr = SixDegreesSDK_GetBackgroundTexture();  
if (texPtr != nil) {  
    id<MTLTexture> mtlTexture = (__bridge id<MTLTexture>)texPtr;  
}
```

## New API methods for OpenGL ES

- `void SixDegreesSDK_InitializeWithEAGL(void* eaglContext)`  
Initializes the 6D.ai SDK to run with OpenGL ES 3.0, identical to the former API call `void SixDegreesSDK_Initialize(void* eaglContext)` from earlier versions.
- `int SixDegreesSDK_GetEAGLBackgroundTexture(void)`  
Returns the OpenGL ES texture handle of the camera background, identical to the former API call `int SixDegreesSDK_GetBackgroundTexture(void)` from earlier versions. Will return 0 if the SDK is not initialized or was initialized with Metal.

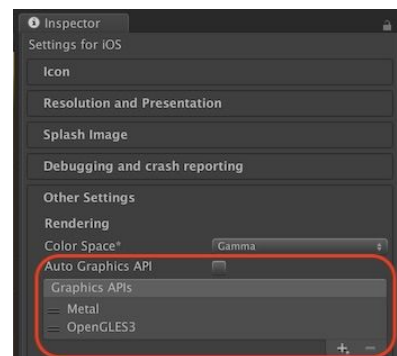
If you're upgrading from beta 0.17.0 and wish to continue using OpenGL ES, make sure your code calls those new API methods.

## Switching between graphics APIs in the Sample apps

### Unity:

Go to the Build Settings menu (⌘B) and click "Player Settings...". In the panel that opens, find "Settings for iOS" > "Other Settings".

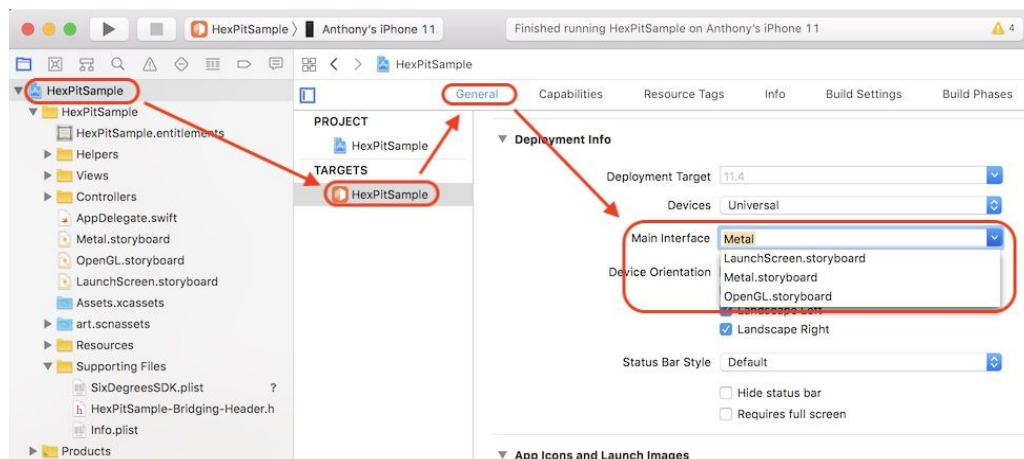
Ticking "Auto Graphics API" will select Metal; unticking it lets you manually choose between "**Metal**" and "**OpenGLES3**", whichever is on top will be the selected graphics API.



### SceneKit:

Open the HexPitSample project settings, select the HexPitSample target, and in the General tab find "Deployment Info" > "Main Interface".

Selecting "**Metal.storyboard**" or "**OpenGL.storyboard**" will instantiate a different SixDegreesView implementation for the namesake graphics API.



## Sample apps fixes

- Fixed an issue in SDMesh.cs (Unity Sample App) and MeshController.swift (Swift/SceneKit Sample App) where part of the update code was inefficient and slowing down the app after a few minutes.
- Fixed a crash in AutoNetworkManager.cs (Unity Sample App) when no action is attached to `OnConnectionEvent`.
- Made the animation of multiplayer camera position updates in the Drawing Sample / Photon Drawing Sample of the Unity Sample App a lot smoother by increasing the update sending rate.

## Hardware Requirements

As of SDK version 0.18.0, only the following iOS devices are supported:

Year	Supported iPhones <a href="#">Identify your iPhone model</a>	Supported iPads <a href="#">Identify your iPad model</a>
2018	iPhone XS iPhone XS Max iPhone XR	iPad Pro 12.9" (3rd Gen) iPad Pro 11" iPad (6th Gen)
2017	iPhone X iPhone 8 iPhone 8 Plus	iPad Pro 12.9" (2nd Gen) iPad Pro 10.5"
2016	iPhone 7 iPhone 7 Plus	iPad Pro 9.7"
2015		iPad Pro 12.9" (1st Gen)

**Not** supported:

- iPhone 6S (2015) and lower
- iPad 5th Gen (2017) and lower
- iPad Air family
- iPad Mini family

The SDK **will not initialize** on those devices, even if they support ARKit.

The API method `SixDegreesSDK_IsDeviceSupported()` should be used at runtime to detect if the device is supported by the SDK.