

# 1. FEATURE

- A. Movie Textures are animated Textures that are created from a video file(supported Android/IOS)
- B. Support 3D Sound(Mobile-Only)
- C. Support default feature [Play, stop, Pause, Resume, SeekTo(Mobile-Only)]

# 2. REQUIREMENT

# **IOS** : OpenGL ES 2.0 or OpenGL ES 3.0(Unity5 Metal Graphics Level not supported)

# **Android** : OpenGL ES 2.0 or higher, Android 4.0 or higher

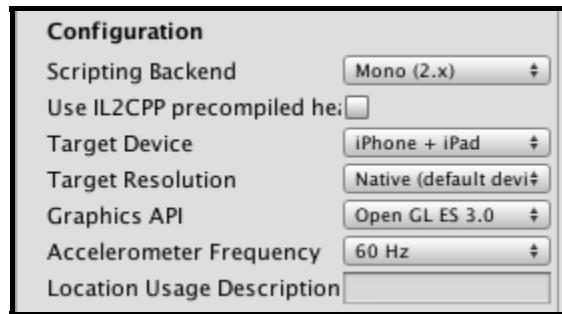
# **PC, Mac & Linus Standalone** : Unity 3D Pro

# **Supported Movie Formats**

- Android : [Android Formats Link](#)
- IOS : [IOS Formats Link](#)
- Android And IOS : mp4( Video:MPEG-4, Audio:AAC )

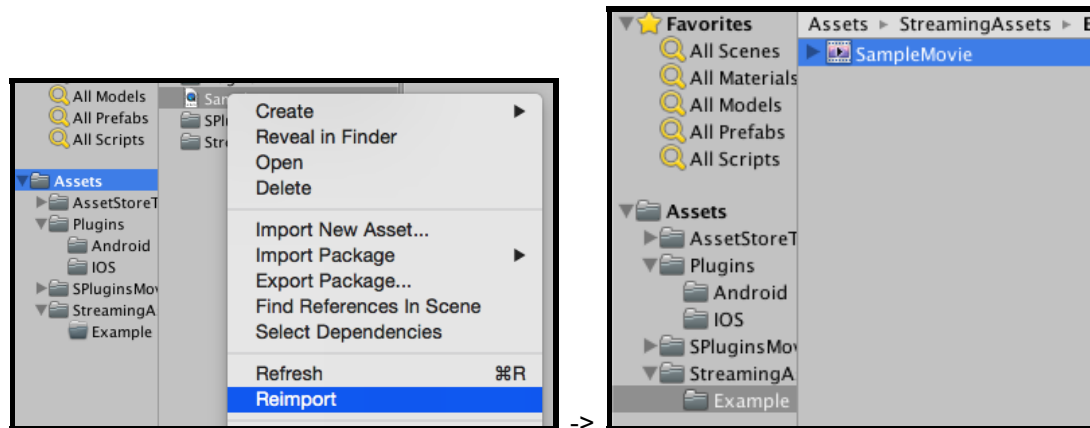
### 3. UNITY5 SETTINGS

# IOS : File->Build Settings->IOS Player Settings -> Other Settings -> Graphics Level -> Change Open GL ES 2.0 or Open GL ES 3.0



# PC,Mac Standalone : [Assets/StreamingAssets/Example/SampleMovie.mp4] file move to [Assets] folder-> Right click -> Click Reimport menu ->

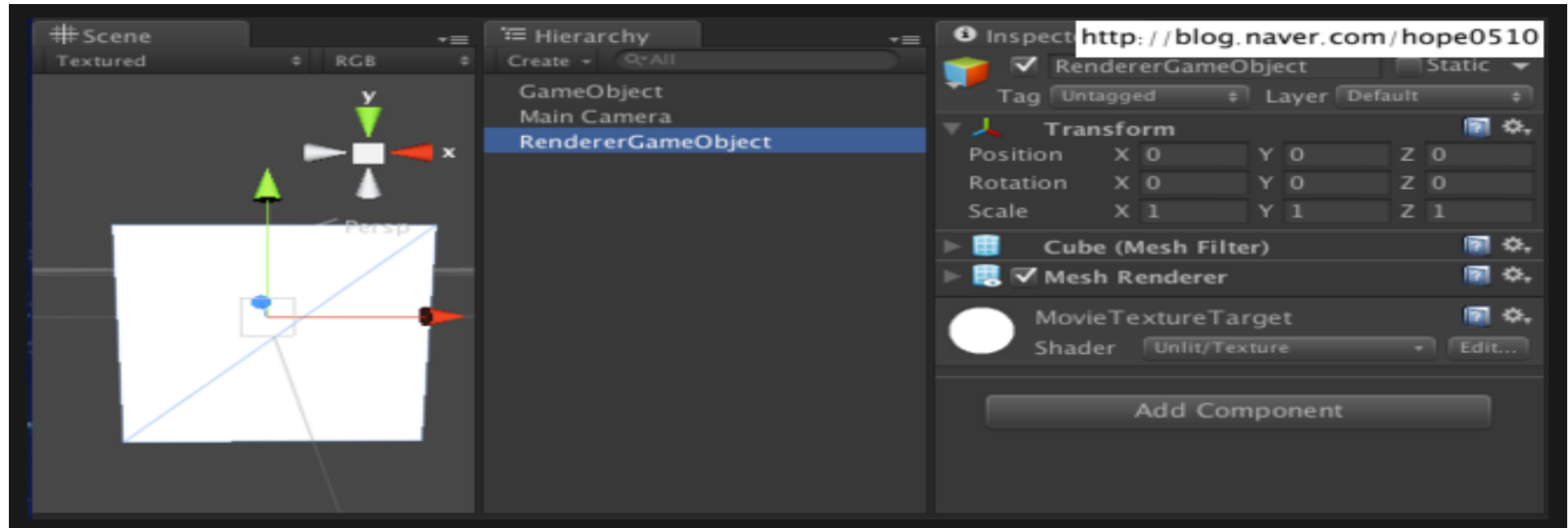
[Assets/SampleMovie.mp4] file move to [Assets/StreamingAssets] or [Assets/StreamingAssets/Example]folder



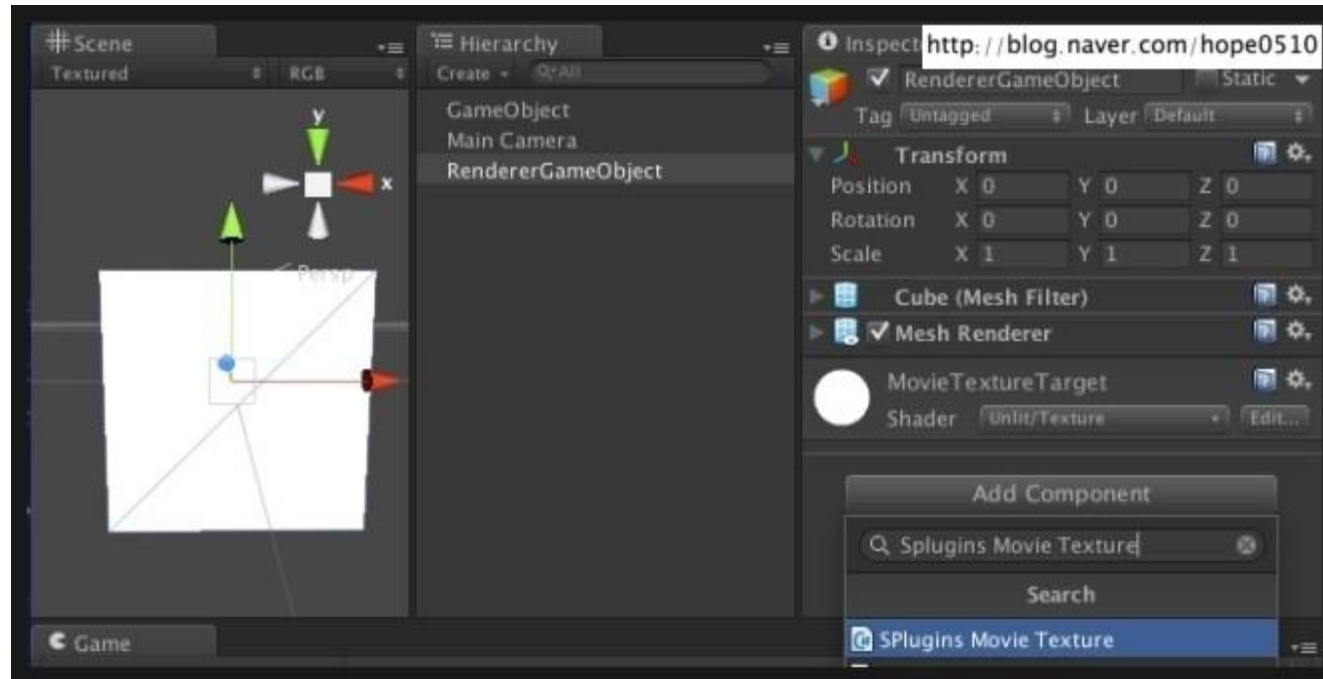
## 4. HOW TO USE

### A. Add Component

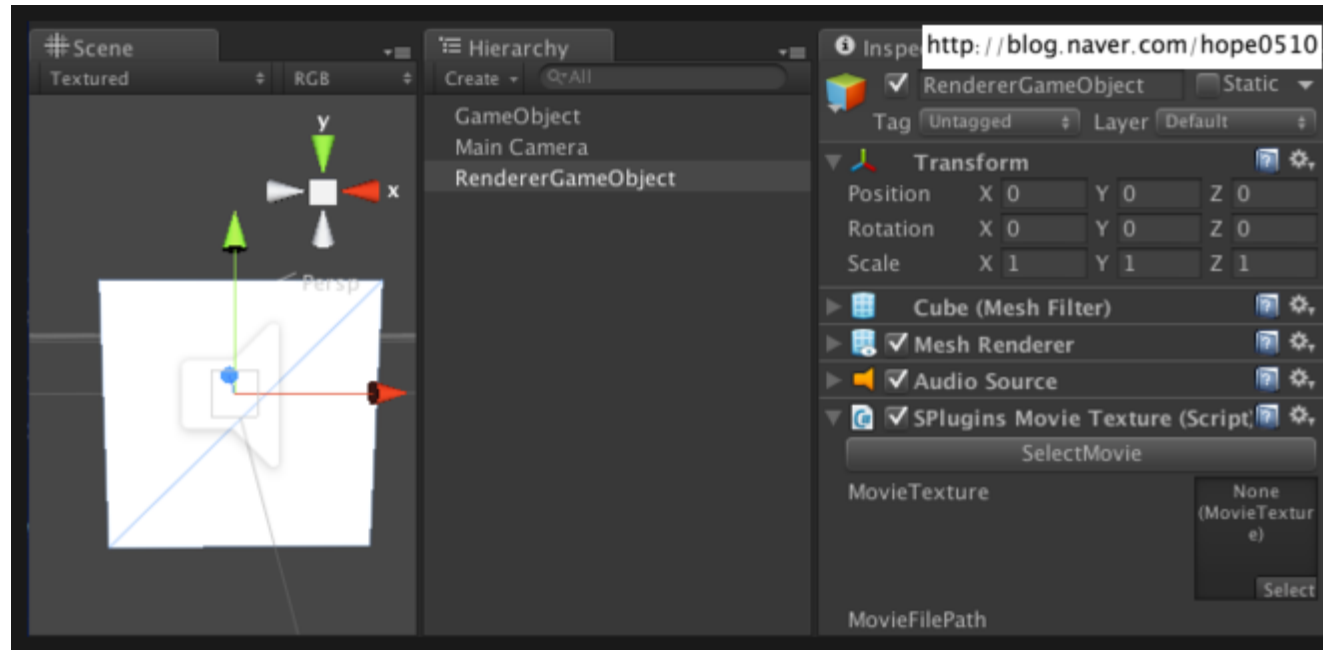
- i. Create a Mesh Renderer Object



- ii. Add Splugins Movie Texture (Component)

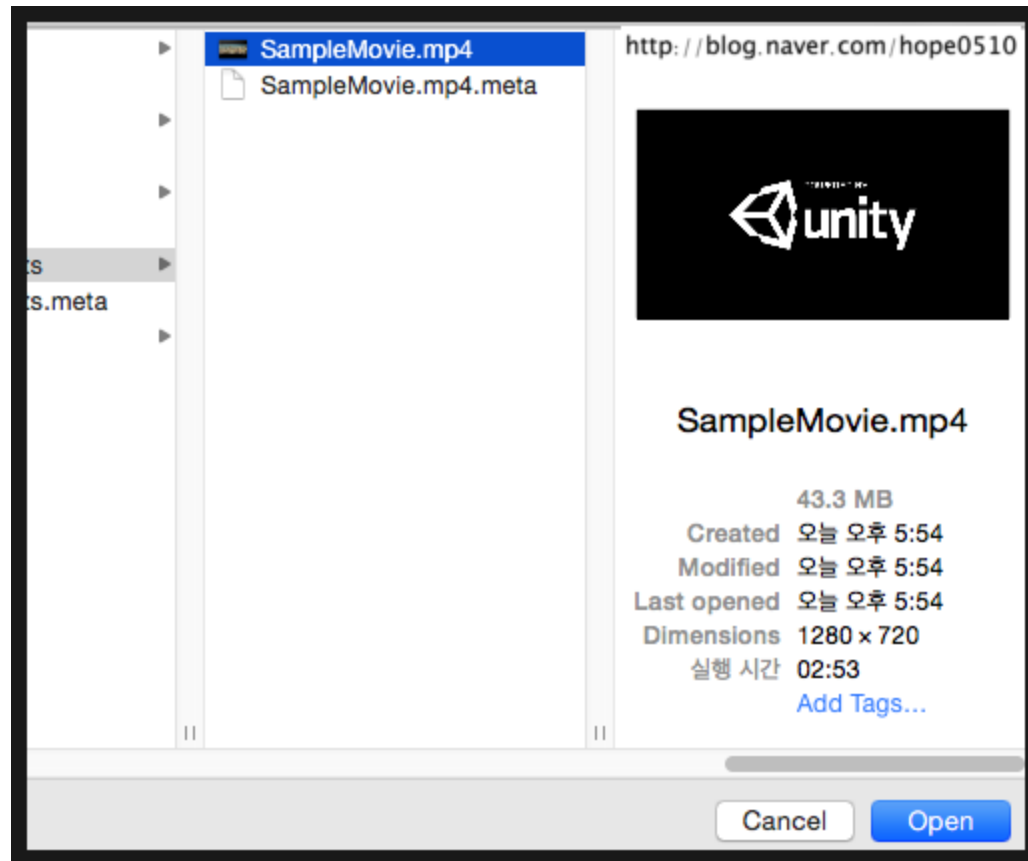


- iii. Click the SelectMovie Button In SPluginsMovieTexture Inspector window

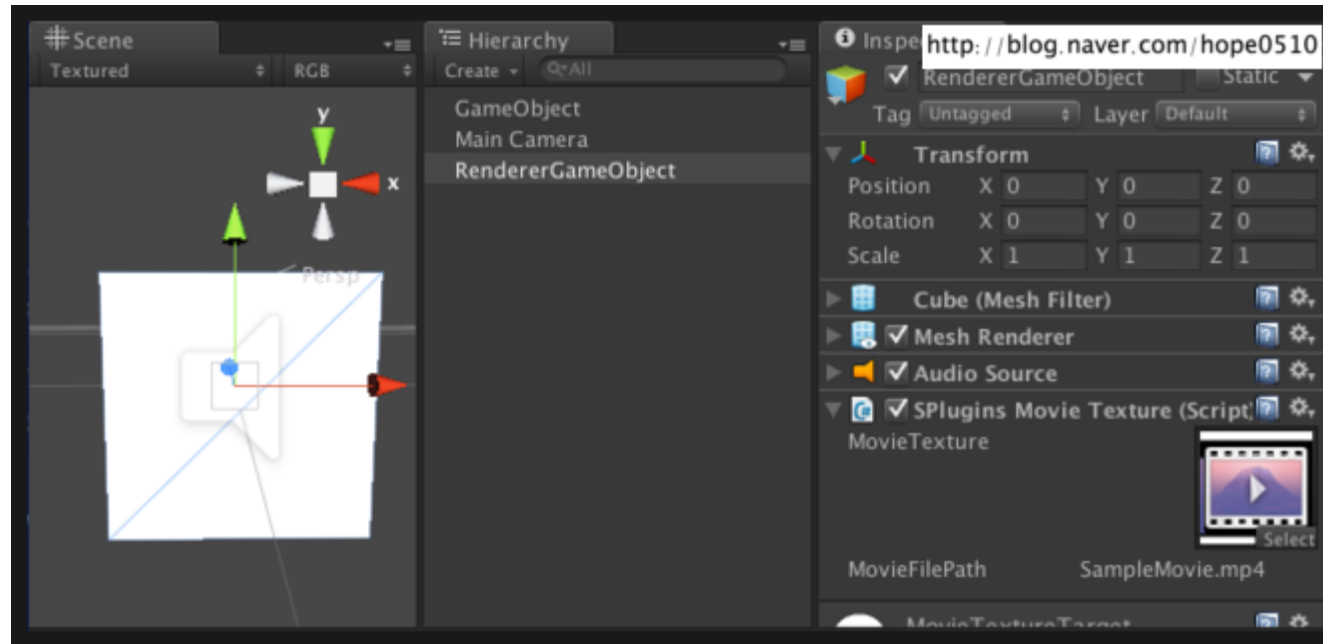


iv. Select Movie file

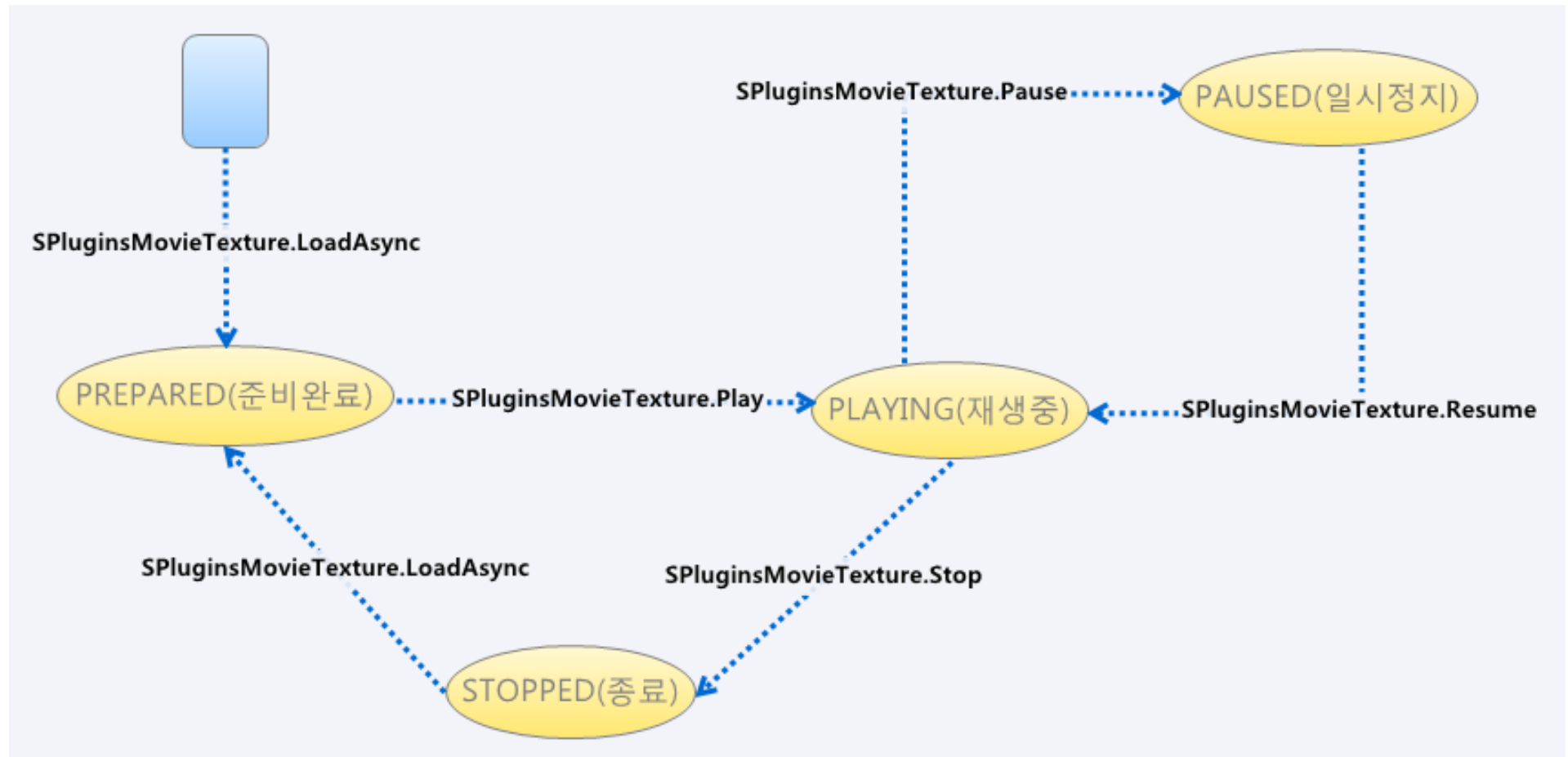
- The movie file location should exist in the [Assets/StreamingAssets] folder



v. Completed



## B. Use API





i. Load – SPluginsMovieTexture.LoadAsync

```
public class SampleMain : MonoBehaviour
{
    public SPluginsMovieTexture movieTexture;
    void Load()
    {
        movieTexture.LoadAsync(HandleOnLoadCompleted);
    }
    void HandleOnLoadCompleted(SPlugins.MovieTexture.AbstractMovieTextureComponent.ResultType resultType_,
        string resultDescription_)
    {
        if(SPlugins.MovieTexture.AbstractMovieTextureComponent.ResultType.SUCCESS == resultType_)
        {
            /*Load Complete Success*/
        }
        else
        {
            /*Load Failed*/
            Debug.LogError(string.Format("Result: {0} - {1}", resultType_.ToString(), resultDescription_));
        }
    }
}
```

ii. Play - SPluginsMovieTexture.Play

```
public class SampleMain : MonoBehaviour
{
    public SPluginsMovieTexture movieTexture;
    void Play()
    {
        movieTexture.Play(HandleOnPlayCompleted);
    }
    void HandleOnPlayCompleted(SPlugins.MovieTexture.AbstractMovieTextureComponent.CompletedType completedType_)
    {
        if(SPlugins.MovieTexture.AbstractMovieTextureComponent.CompletedType.PLAYBACK_COMPLETION == completedType_)
        {
            if (null != movieTexture)
            {
                movieTexture.Stop();
            }
        }
    }
}
```

iii. Stop - SPluginsMovieTexture.Stop

```
public class SampleMain : MonoBehaviour
{
    public SPluginsMovieTexture movieTexture;
    void Stop()
    {
        movieTexture.Stop();
    }
}
```

iv. Pause - SPluginsMovieTexture.Pause

```
public class SampleMain : MonoBehaviour
{
    public SPluginsMovieTexture movieTexture;
    void Pause()
    {
        movieTexture.Pause();
    }
}
```

v. Resume - SPluginsMovieTexture.Resume

```
public class SampleMain : MonoBehaviour
{
    public SPluginsMovieTexture movieTexture;
    void Resume()
    {
        movieTexture.Resume();
    }
}
```

vi. SeekTo - SPluginsMovieTexture.SeekTo

```
public class SampleMain : MonoBehaviour
{
    public SPluginsMovieTexture movieTexture;
    void SeekTo(int seekTimeInMilliseconds)
    {
        movieTexture.SeekTo(seekTimeInMilliseconds);
    }
}
```

vii. Loop - SPluginsMovieTexture.SetLooping

```
public class SampleMain : MonoBehaviour
{
    public SPluginsMovieTexture movieTexture;
    void SetLooping(bool loop_)
    {
        movieTexture.SetLooping(loop_);
    }
}
```

viii. Volume - SPluginsMovieTexture.SetVolume

```
public class SampleMain : MonoBehaviour
{
    public SPluginsMovieTexture movieTexture;
    void SetVolume(float normalizedVolume_)
    {
        movieTexture.SetVolume(normalizedVolume_);
    }
}
```

- ix. 3d sound on/off - SPluginsMovieTexture.Enable3DSound

```
public class SampleMain : MonoBehaviour
{
    public SPluginsMovieTexture movieTexture;
    void Enable3DSound(bool enable_)
    {
        movieTexture.Enable3DSound(enable_);
    }
}
```

- x. 3d sound min distance - SPluginsMovieTexture.Set3DSoundMinDistance

```
public class SampleMain : MonoBehaviour
{
    public SPluginsMovieTexture movieTexture;
    void Set3DSoundMinDistance(float distance_)
    {
        movieTexture.Set3DSoundMinDistance(distance_);
    }
}
```

xi. 3d sound max distance - SPluginsMovieTexture.Set3DSoundMaxDistance

```
public class SampleMain : MonoBehaviour
{
    public SPluginsMovieTexture movieTexture;
    void Set3DSoundMaxDistance(float distance_)
    {
        movieTexture.Set3DSoundMaxDistance(distance_);
    }
}
```

# Need more Help?

Visit <http://blog.naver.com/hope0510>

Or email us at

[sunityplugins@gmail.com](mailto:sunityplugins@gmail.com)



# Release Notes

## **Version1.1(20150331)**

Fix a bug played only sound on IOS devices

Add LoadAsync function( LoadAsync(string path, ...) );