



Unityはじめるよ ～AssetBundle基礎～

統合開発環境を内蔵したゲームエンジン
<http://japan.unity3d.com/>

※いろんな職業の方が見る資料なので説明を簡単にしている部分があります。正確には本来の意味と違いますが上記理由のためです。ご了承ください。
この資料内の一部の画像、一部の文章はUnity公式サイトから引用しています。

アセットバンドルって？

アセットバンドル は選択したアセットを 1つにまとめて Unity からエクスポートできるファイルです。

これらのファイルは専用の圧縮フォーマットを使用してアプリケーションからオンデマンドでロードできます。

これによりモデル、テクスチャ、オーディオクリップ、または使用されるシーンから切り離してシーン全体などのコンテンツをストリーミングすることができます。

公式ドキュメントから引用

<http://docs.unity3d.com/ja/current/Manual/AssetBundlesIntro.html>

アセットバンドルを使う利点

アセットバンドルを使うと、
アプリ本体とリソースを切り離すことができる。

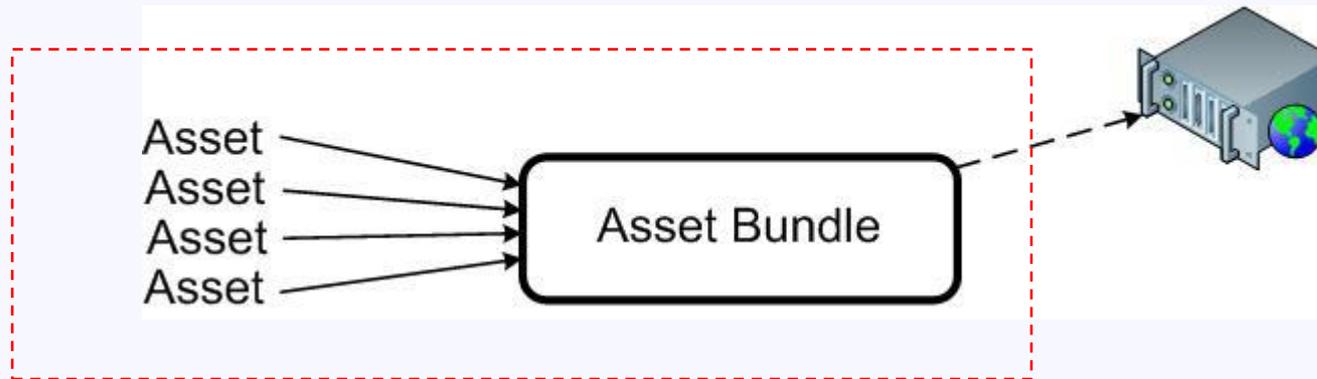
スマホアプリには特に有効

スマホアプリはアプリサイズが一定のサイズを超えていると
Wifiにつないでいる時にしかDLできない。

貴重なDL機会を失ってしまうことになりかねない。

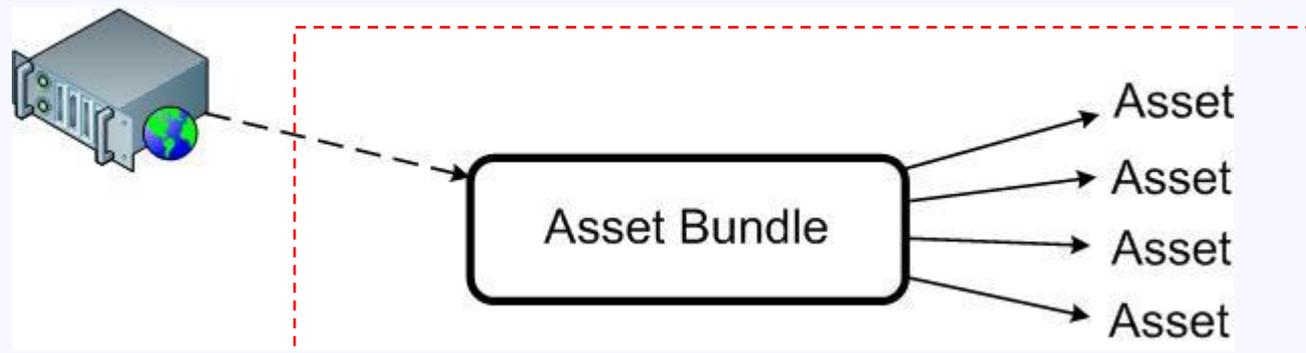
また、アプリ申請時は仮のデータだけど、
公開時までには本番データに差し替えたい時にも有効。

エディター上で、
AssetからAssetBundleを作ってサーバーにアップ



エディターでの作業

アプリ実行時に、
サーバーからAssetBundleを取得して、Assetを利用する。



アプリでの作業

アセットバンドル使い方

1. AssetBundle Manager
2. アセットバンドルの作り方
3. アセットバンドルのアップロード
4. アセットバンドルの利用方法

1. AssetBundle Manager



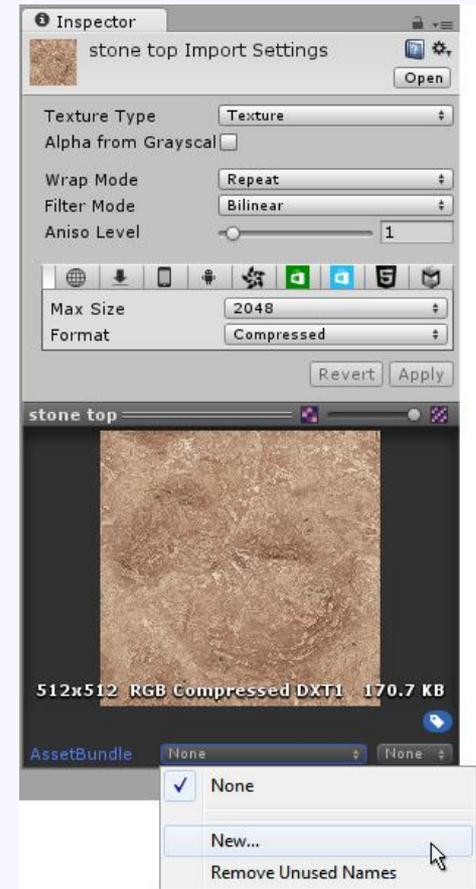
アセットバンドルを扱いやすくするユーティリティ

アセットバンドルを作ったり、使う場合、
ちょっと面倒な作業が必要となるのだが、
その面倒な部分を何とかしてくれるのが、
このAssetBundle Manager。

アセットストアからDLできます。
※UnityTechnologiesが提供！

2. アセットバンドルの作り方

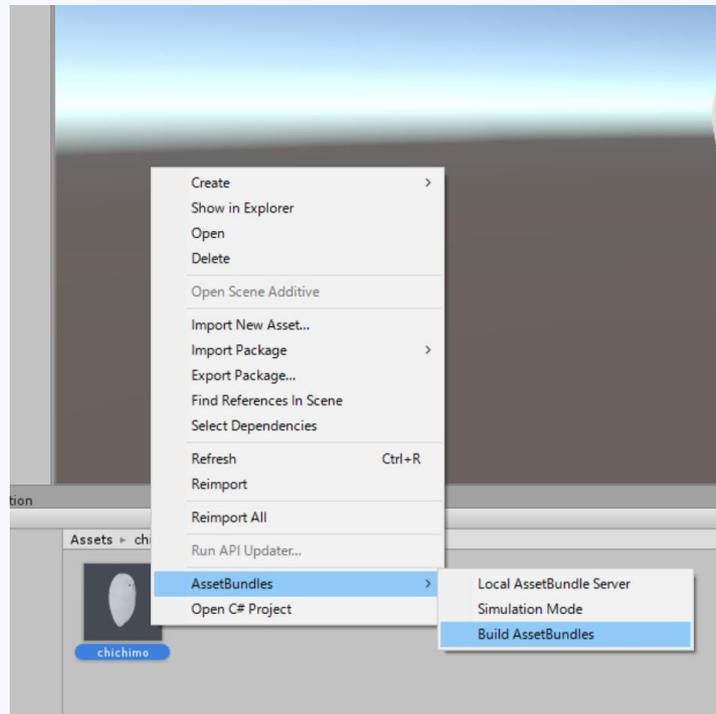
1. プロジェクトビューでアセットバンドル化したいリソースを選択し、インスペクタビューの最下部のアセットバンドルメニューから、Newを選んでバンドル名をつける。
※Prefabをアセットバンドル化することが多いのかな
2. 他にも同じバンドル内に含めたいリソースがあれば、同様の作業を行う。
但し、Newせずに定義済みのバンドル名から選択すること。



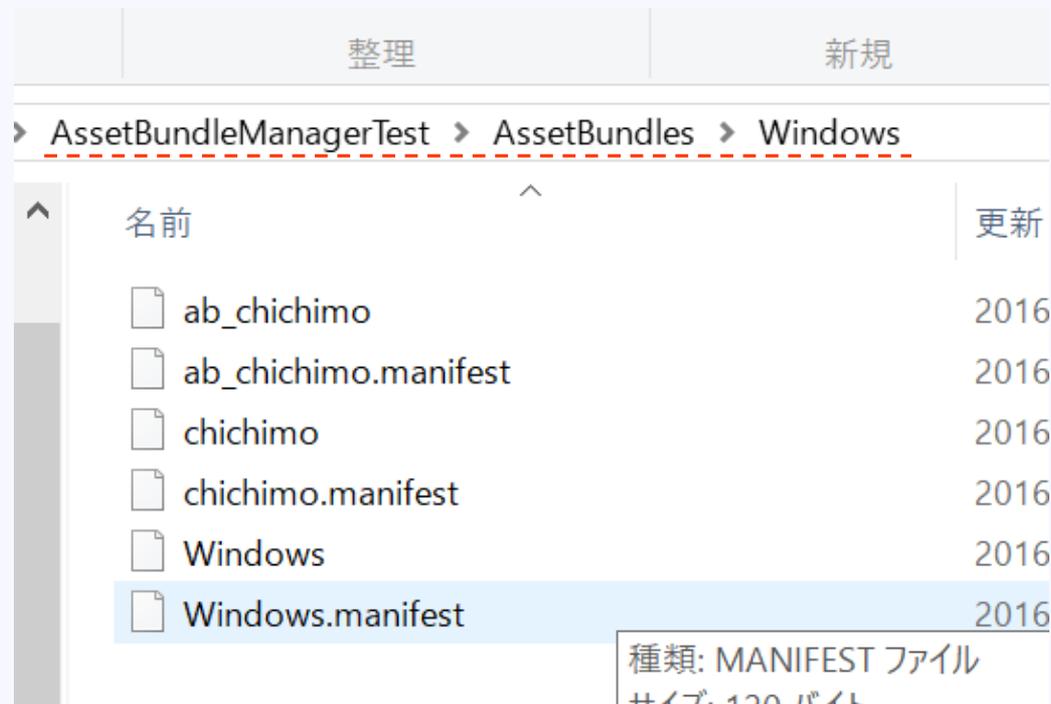
3. アセットバンドルと紐づけたリソースをプロジェクトビューで右クリックし、メニューの中から、

AssetBundles > Build AssetBundle

を選択。これでアセットバンドルの作成は完了。



4. Unityのプロジェクトフォルダ直下に「AssetBundles」というフォルダが作成され、その中にプラットフォーム毎のフォルダが作成され書き出される。(WindowsとかiOSとか)
ただし、書き出されるのは、現在のプラットフォームのみ。
別のプラットフォームのアセットバンドルを作成するには、SwitchPlatformしてからビルドしなおす必要がある。

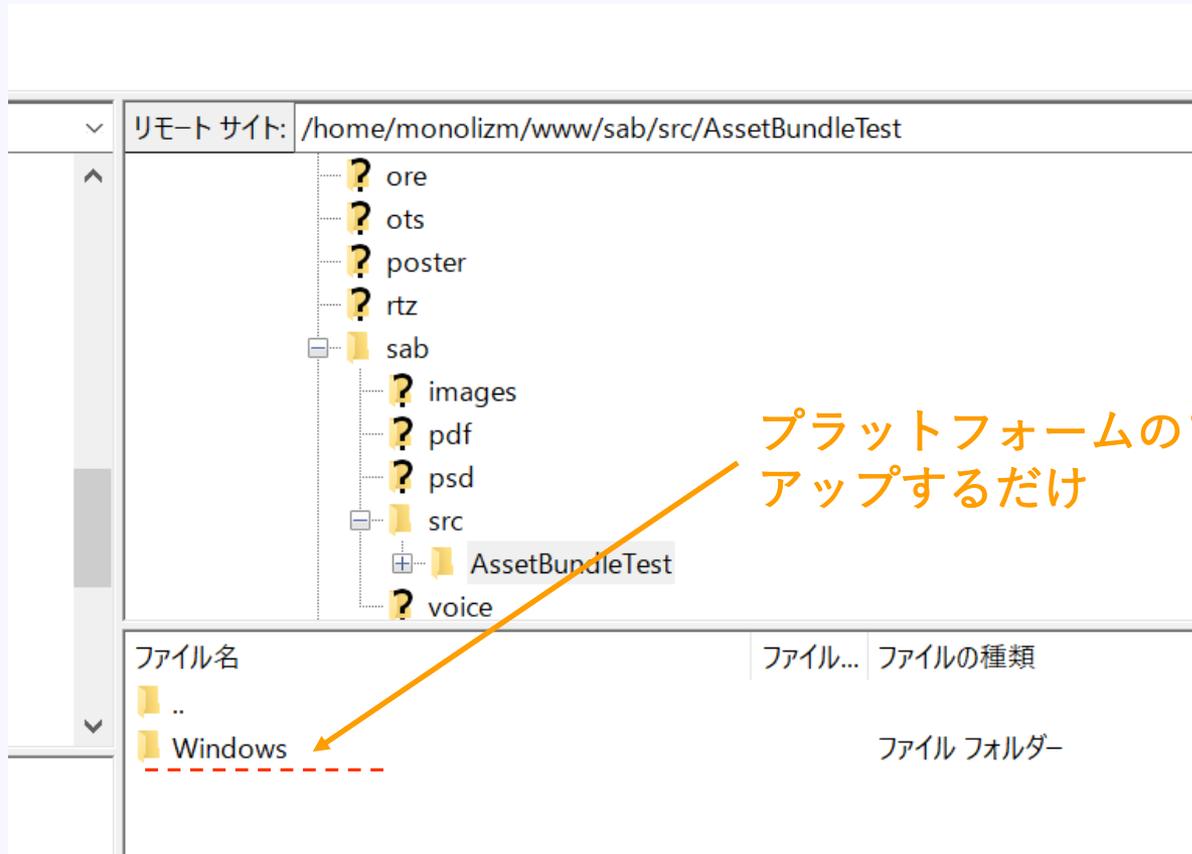


注意事項

- プラットフォームごとに用意する必要がある
- シーンファイルとその他のアセットは別のアセットバンドルにしなくてはならない。

3. アセットバンドルのアップロード

1. FTPクライアントソフトを使ってサーバーに、プラットフォームフォルダごとアップロードするだけ。

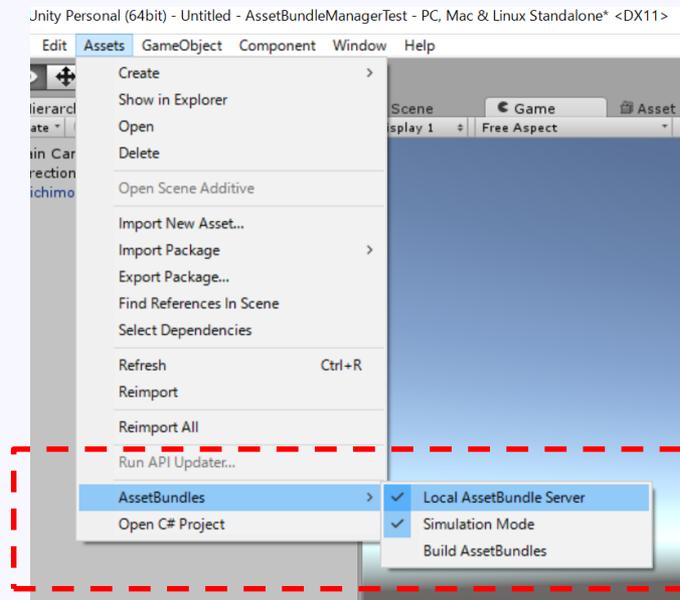


菅原も保坂も使ってるFTPクライアント「filezilla」！
WindowsもMacもあるよ。

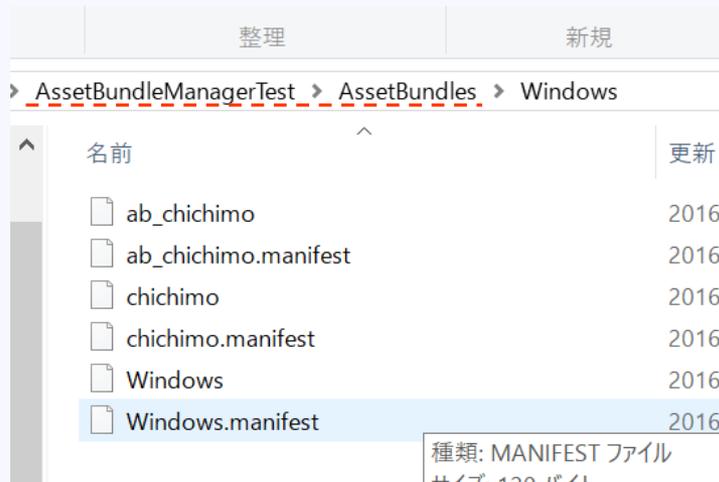
番外編. AssetBundle Managerにはサーバーをシミュレーションする機能がある。動作確認の際はこれを使えば、いちいちサーバーにデータをアップしなくて済むのだ。

メニュー > Assets > AssetBundle > Local AssetBundle Serverにチェックを入れれば使えるぞ。

つまり、チェックを入れるだけで、もう動作確認が可能な状態となるわけだ。

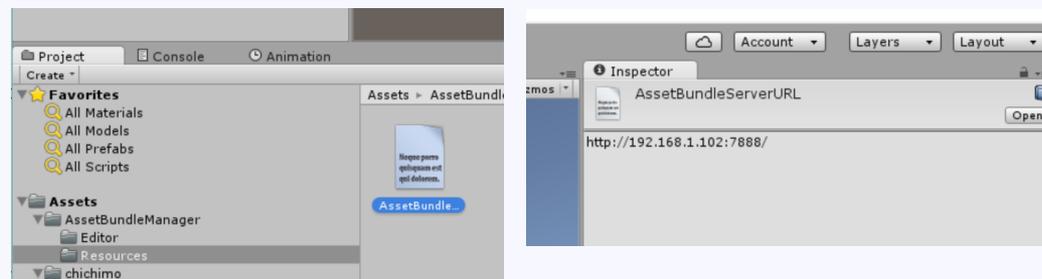


番外編. アクセス先は、
192.168.1.102:7888/
さっき作ったAssetBundles以下を見ることとなる。



赤点線の部分が、
192.168.1.102:7888/となる。
だからWindowsフォルダは
192.168.1.102:7888/Windows
となる。

ちなみにアクセス先は、
AssetBundleManager/Resources内の**AssetBundleServerURL**ってファイルに書いてある



4. アセットバンドルの利用方法

1. アセットバンドルをダウンロードするスクリプトを作る。

```
/// <summary>
/// アセットバンドルをロード。
/// </summary>
/// <returns>The asset bundle coroutine.</returns>
/// <param name="url">URL.</param>
public IEnumerator LoadAssetBundleCoroutine(string url)
{
    // アセットバンドルをダウンロード
    WWW www = WWW.LoadFromCacheOrDownload(url, 0);
    while (!www.isDone)
    {
        yield return null;
    }

    // キャッシュしておく
    m_assetBundleCache = www.assetBundle;

    // 開放
    www.Dispose();
}
```

通信で取得する処理

通信で取得したものを
キャッシュしておく
※通信処理は開放したいため

2. アセットバンドルから指定のデータを読み込むスクリプトを作る。

```
/// <summary>
/// アセットバンドルからGameObjectを取得
/// </summary>
/// <returns>The game object from asset bundle.</returns>
/// <param name="assetName">Asset name.</param>
public GameObject GetGameObjectFromAssetBundle(string assetName)
{
    try
    {
        // プレハブを取得してインスタンス化
        GameObject prefab = m_assetBundleCache.LoadAsset<GameObject>(assetName);
        return GameObject.Instantiate(prefab);
    }
    catch (NullReferenceException e)
    {
        Debug.Log("LoadAssetBundle::GetGameObjectFromAssetBundle [" + e.ToString() + "]);
        return null;
    }
}
```

キャッシュしといたアセットバンドルから
指定のファイルを取得

3. 今回は1, 2の処理を合わせたスクリプトを作りました。

```
using UnityEngine;
using System;
using System.Collections;

public class LoadAssetBundle : MonoBehaviour
{
    /// <summary>
    /// アセットバンドルのキャッシュ用.
    /// </summary>
    private AssetBundle m_assetBundleCache;

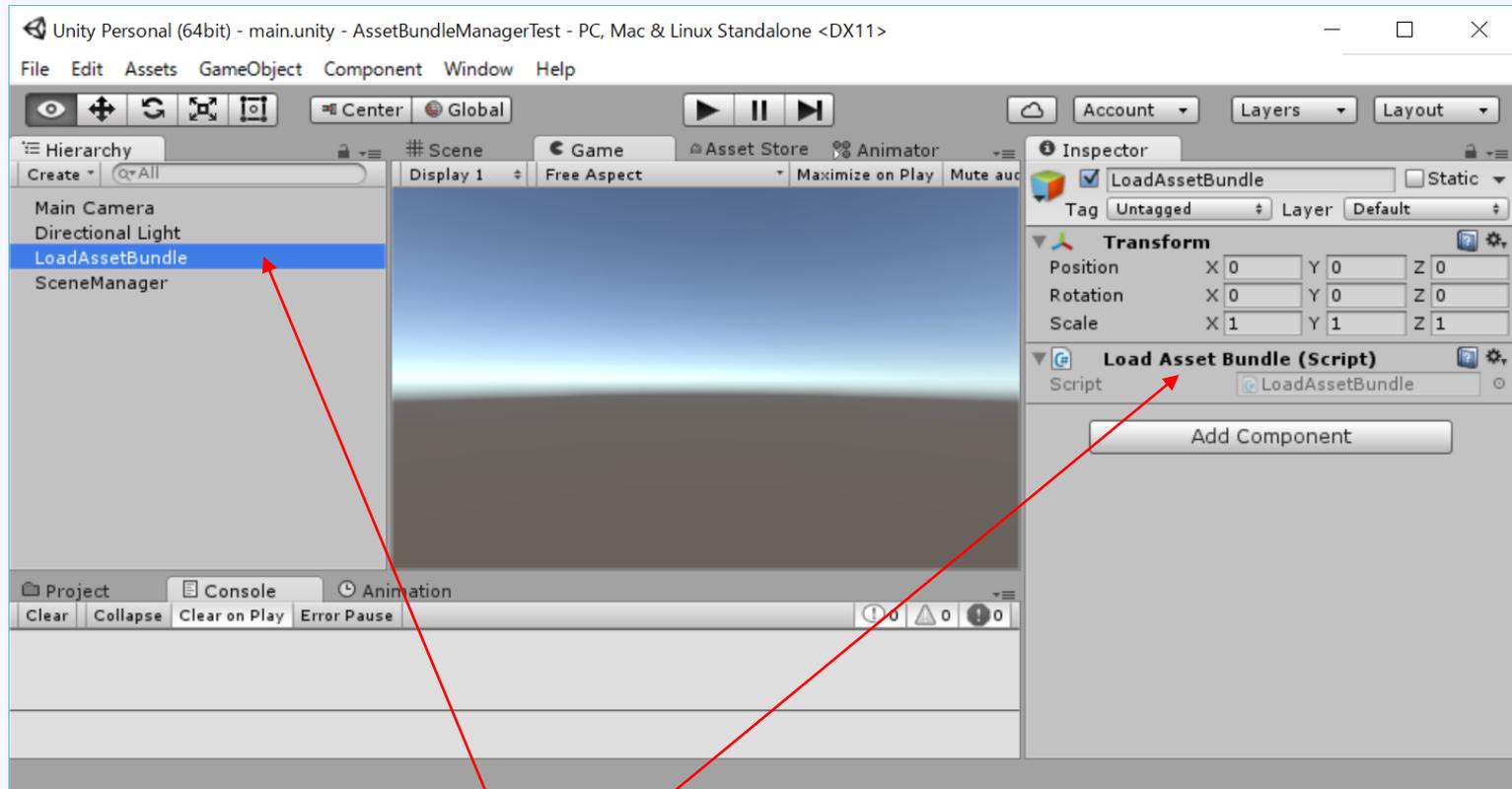
    /// <summary>
    /// アセットバンドルをロード.
    /// </summary>
    /// <returns>The asset bundle coroutine.</returns>
    /// <param name="url">URL.</param>
    public IEnumerator LoadAssetBundleCoroutine(string url)
    {
        // アセットバンドルをダウンロード
        WWW www = WWW.LoadFromCacheOrDownload(url, 0);
        while (!www.isDone)
        {
            yield return null;
        }

        // キャッシュしておく
        m_assetBundleCache = www.assetBundle;

        // 開放
        www.Dispose();
    }

    /// <summary>
    /// アセットバンドルからGameObjectを取得
    /// </summary>
    /// <returns>The game object from asset bundle.</returns>
    /// <param name="assetName">Asset name.</param>
    public GameObject GetGameObjectFromAssetBundle(string assetName)
    {
        try
        {
            // プレハブを取得してインスタンス化
            GameObject prefab = m_assetBundleCache.LoadAsset<GameObject>(assetName);
            return GameObject.Instantiate(prefab);
        }
        catch (NullReferenceException e)
        {
            Debug.Log("LoadAssetBundle::GetGameObjectFromAssetBundle [" + e.ToString() + "];");
            return null;
        }
    }
}
```

4. 3のスク립トをGameObjectにアタッチ



これね

5. さっきのスクリプトを使う側の処理。

```
using UnityEngine;
using System.Collections;

public class SceneManager : MonoBehaviour {

    /// <summary>
    /// アセットバンドルのURL.
    /// </summary>
    public string AssetBundleBaseURL = "http://192.168.1.102:7888/";

    /// <summary>
    /// アセットバンドルローダー
    /// </summary>
    public LoadAssetBundle m_loadAssetBundle;

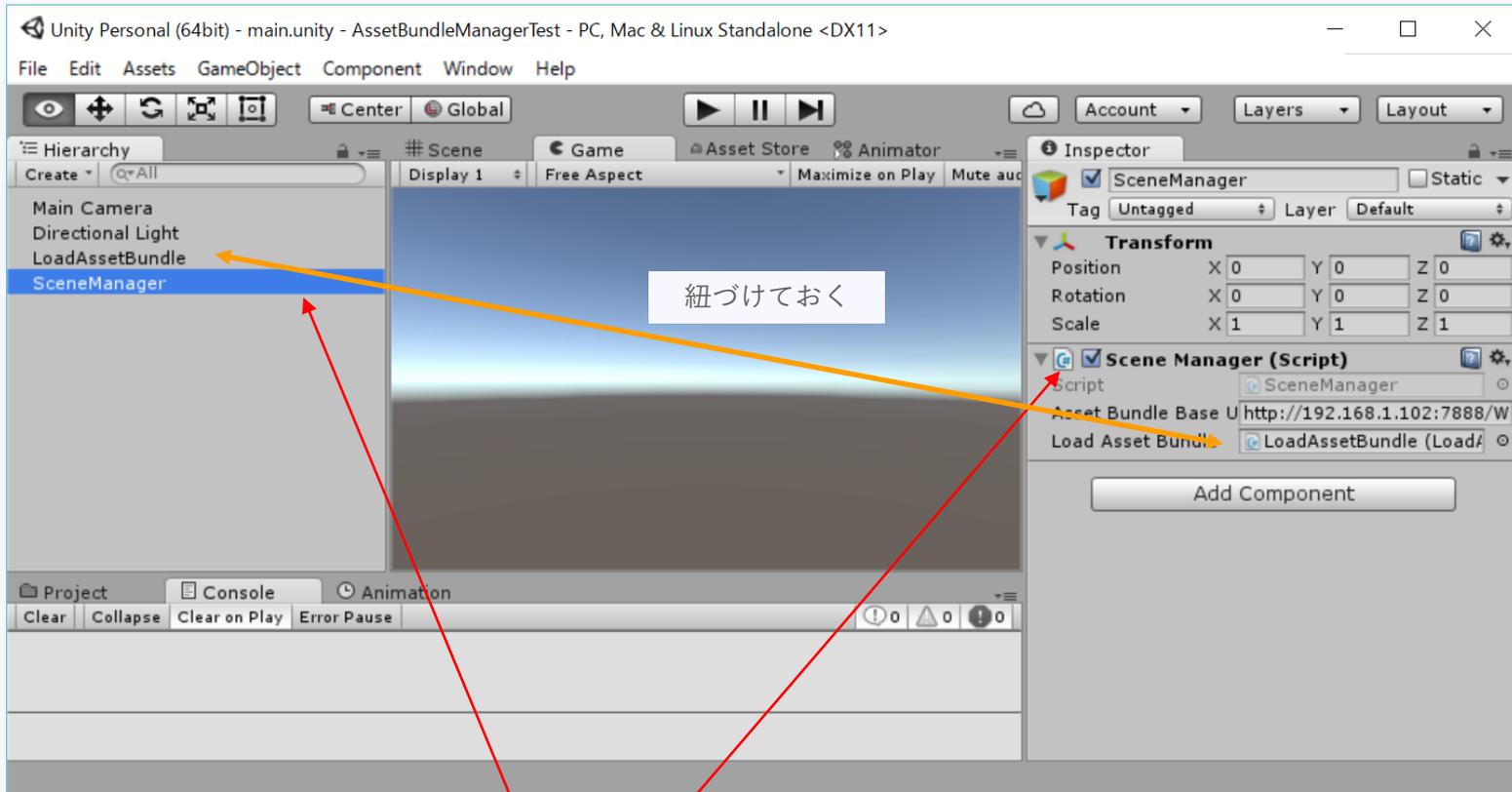
    /// <summary>
    /// アセットバンドルから取り出したGameObjectを入れるためのGameObject
    /// </summary>
    private GameObject m_gameObject;

    /// <summary>
    /// 初期化
    /// </summary>
    public IEnumerator Start() {
        // まずはアセットバンドルのダウンロード
        yield return StartCoroutine(m_loadAssetBundle.LoadAssetBundleCoroutine(AssetBundleBaseURL));

        // ダウンロード後にアセットバンドルから中身を取り出す
        m_gameObject = m_loadAssetBundle.GetGameObjectFromAssetBundle("Cube");
    }
}
```

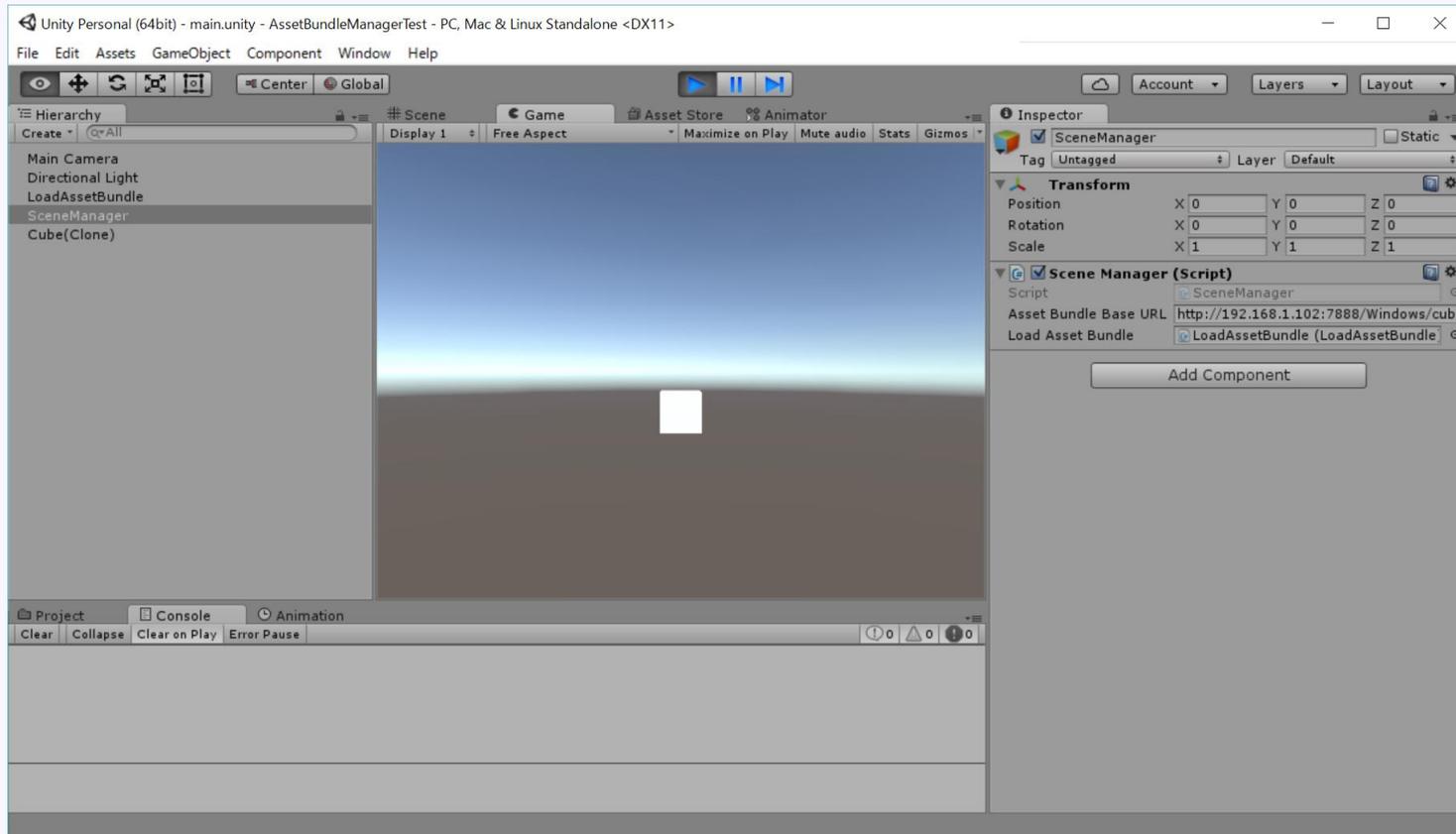
アセットバンドルをダウンロードして
使う部分は実質この2行だけ

6. 5のスク립トをGameObjectにアタッチ



これね

7. 実行してみる



実行結果

アセットバンドル「cube」からCubeというPrefabを取り出してインスタンス化した状態

■まとめ

アセットバンドルを使うと、リソースを外部に出せる。

メリット（特にスマホアプリ目線で）

- ・ **アプリ本体の容量削減**
- ・ **アプリ公開中にデータを差し替えられる**
- ・ **実機で動作確認時、再ビルドなしでデータ差し替え可能**

今回はシンプルな使い方の紹介だったけど、調べてみたらこれまた奥が深いこと…。

AssetBundle Managerの良いところや、AssetBundleを使うデメリットについてもあまり触れてないので、その辺りはまた今度。

■参考サイト

Unity公式マニュアル

<http://docs.unity3d.com/ja/current/Manual/AssetBundlesIntro.html>

Unityの中のスライド

<http://ssslide.com/www.slideshare.net/MakotoItoh/unite2015-assetbundle-final>

テラシュールブログ

<http://tsubakit1.hateblo.jp/entry/2015/12/16/233336>

<http://tsubakit1.hateblo.jp/entry/2016/04/01/184607>

Unity5系でのAsset Bundle使い方まとめ

<http://qiita.com/Ijoru/items/c8e002a09f1b9d533adb>

ご清聴ありがとうございました