Unity資源打包之Asset Bundle知識點整理

原創 月儿圆 2020-06-28 21:41

Asset Bundle的作用：

1.AssetBundle是一個壓縮包包含模型、貼圖、預製體、聲音、甚至整個場景，可以在遊戲運行的時候被加載；

2.AssetBundle自身保存着互相的依賴關係；

3.壓縮包可以使用LZMA和LZ4壓縮算法，減少包大小，更快的進行網絡傳輸；

4.把一些可以下載內容放在AssetBundle裏面，可以減少安裝包的大小；

什麼是AssetBundle

可以歸爲兩點：

1，它是一個存在於硬盤上的文件。可以稱之爲壓縮包。這個壓縮包可以認爲是一個文件夾，裏面包含了多個文件。這些文件可以分爲兩類：serialized file 和 resource files。（序列化文件和源文件）

serialized file：資源被打碎放在一個對象中，最後統一被寫進一個單獨的文件（只有一個）

resource files：某些二進制資源（圖片、聲音）被單獨保存，方便快速加載

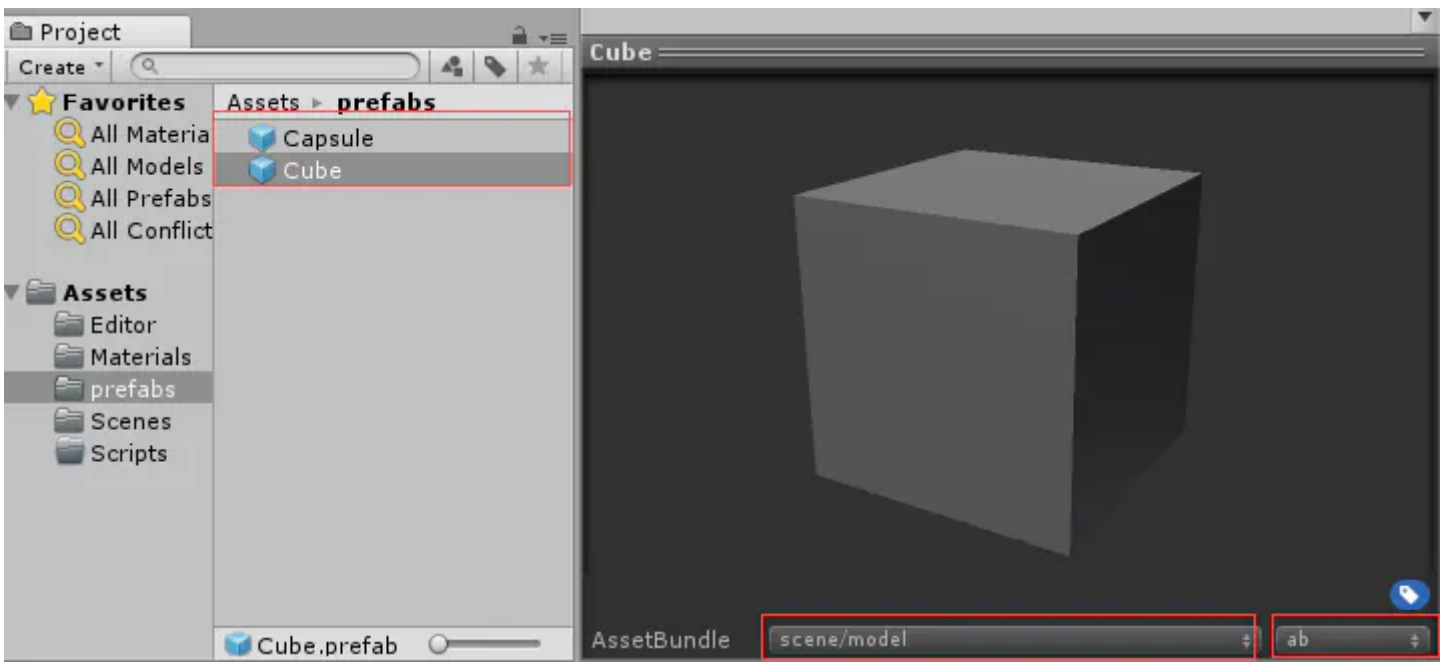
2，它是一個AssetBundle對象，我們可以通過代碼從一個特定的壓縮包加載出來的對象。這個對象包含了所有我們當初添加到這個壓縮包裏面的內容，我們可以通過這個對象加載出來使用。

Asset Bundle資源打包實例

無論是模型資源還是UI資源，最好是先把他們放在Prefab中，然後在做成Assetbundle。我們以模型來舉例，Assetbundle中可以放一個模型、也可以放多個模型，它是非常靈活了那麼最需要考慮的就是模型空間佔用的問題。

下面我們來實際操作下，首先隨便先創建兩個3D對象，Cube和Capsule,並將他們做成Prefab,然後去指定資源的AssetBundle屬性，這裏我將這兩個模型都打包成model.ab包

（xxxa/xxx）這裏xxxa會生成目錄，名字爲xxx ，後面的ab是後綴名，可自己制定。



Paste_Image.png

設置好屬性後，下面開始構建AssetBundle包，首先先創建一個文件夾命名Editor，這個文件夾是不會進行打

月 月儿圆

24小時熱門文章

WCF 學習研究 推薦blog

最新文章

Unity技巧分享之批量修改精靈貼圖命名

Unity技術分享之ARFoundation打包Xcode真機遇到的"objc-class-ref in UnityARKit.a"問題解決

Unity資源打包之Asset Bundle知識點整理

unity3d 關於AssetBundle的一些處理細節

Unity技術分享之動態設置腳本執行順序

最新評論文章

Spring Boot 統一參數校驗、統一異常、統一響應，這纔是優雅的處理方式！

winform input 輸入數值（可以小數，負數）

For 健康，還在糾結"喫什麼"？答案在這裏！——營養膳食的基礎準則

```
using UnityEditor;
using System.IO;

public class CreateAssetbundles  {

    [MenuItem("AssetsBundle/Build AssetBundles")]
     static void BuildAllAssetBundles()//進行打包
    {
        string dir = "AssetBundles";
        //判斷該目錄是否存在
        if (Directory.Exists(dir) == false)
        {
            Directory.CreateDirectory(dir);//在工程下創建AssetBundles目錄
        }
        //參數一爲打包到哪個路徑，參數二壓縮選項  參數三 平臺的目標
        BuildPipeline.BuildAssetBundles(dir,
BuildAssetBundleOptions.None,BuildTarget.StandaloneWindows64);
    }
}
```

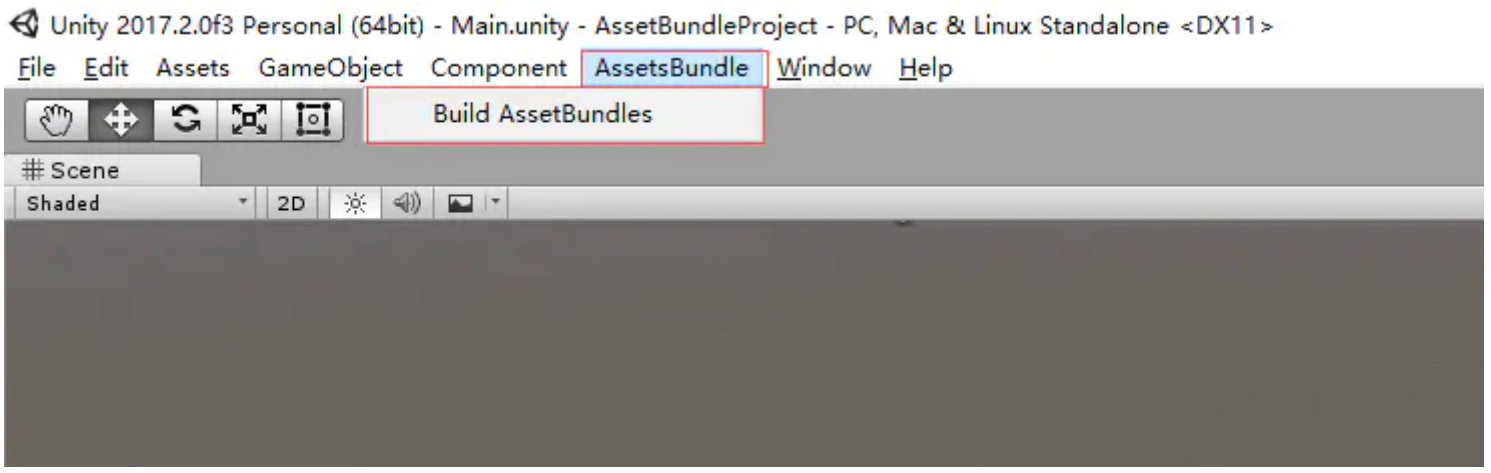
BuildAssetBundleOptions.None：使用LZMA算法壓縮，壓縮的包更小，但是加載時間更長。使用之前需要整體解壓。一旦被解壓，這個包會使用LZ4重新壓縮。使用資源的時候不需要整體解壓。在下載的時候可以使用LZMA算法，一旦它被下載了之後，它會使用LZ4算法保存到本地上。

BuildAssetBundleOptions.UncompressedAssetBundle：不壓縮，包大，加載快

BuildAssetBundleOptions.ChunkBasedCompression：使用LZ4壓縮，壓縮率沒有LZMA高，但是我們可以加載指定資源而不用解壓全部。

注意使用LZ4壓縮，可以獲得可以跟不壓縮想媲美的加載速度，而且比不壓縮文件要小。

然後回到Unity裏面點擊我們剛剛擴展出來的打包按鈕



Paste_Image.png

點擊後我們的模型就打包了出來，可以在工程的目錄下可以找到AssetBundles目錄，在AssetBundles下有個Scene文件夾裏面就是我們的打包文件了，後綴是.ab

Paste_Image.png

AssetBundle的加載

AssetBundle的加載有以下幾種方式，從內存加載使用LoadFromMemoryAsync，從本地文件加載可以使用LoadFromFile，從服務器上Web上加載可以使用UnityWbRequest。下面我們來看看這幾種加載的方式。

```
using UnityEngine;
using System.IO;
using System.Collections;
public class LoadFromFileExample : MonoBehaviour {

    IEnumerator  Start () {
        string path = "AssetBundles/scene/model.ab";
        //第一種加載AB的方式 LoadFromMemoryAsync
        //異步加載
        AssetBundleCreateRequest request =
AssetBundle.LoadFromMemoryAsync(File.ReadAllBytes(path));
        yield return request;
        AssetBundle ab = request.assetBundle;
        //同步方式
        //AssetBundle ab=  AssetBundle.LoadFromMemory(File.ReadAllBytes(path));

         //使用裏面的資源
        Object[] obj = ab.LoadAllAssets<GameObject>();//加載出來放入數組中
        // 創建出來
        foreach (Object o in obj)
        {
            Instantiate(o);
        }
    }
}
```

第二種方式(LoadFromFile)從本地加載

```
using UnityEngine;
using System.Collections;

public class LoadFromFileExample : MonoBehaviour {

    IEnumerator  Start () {
        string path = "AssetBundles/scene/model.ab";
        //第二種加載方式 LoadFromFile
        //異步加載
        AssetBundleCreateRequest request = AssetBundle.LoadFromFileAsync(path);
        yield return request;
        AssetBundle ab = request.assetBundle;
        //同步加載
        //AssetBundle ab = AssetBundle.LoadFromFile(path);

        //使用裏面的資源
      Object[] obj = ab.LoadAllAssets<GameObject>();//加載出來放入數組中
        // 創建出來
        foreach (Object o in obj)
        {
            Instantiate(o);
        }
    }
}
```

第三種方式(UnityWbRequest)從服務器或者本地加載

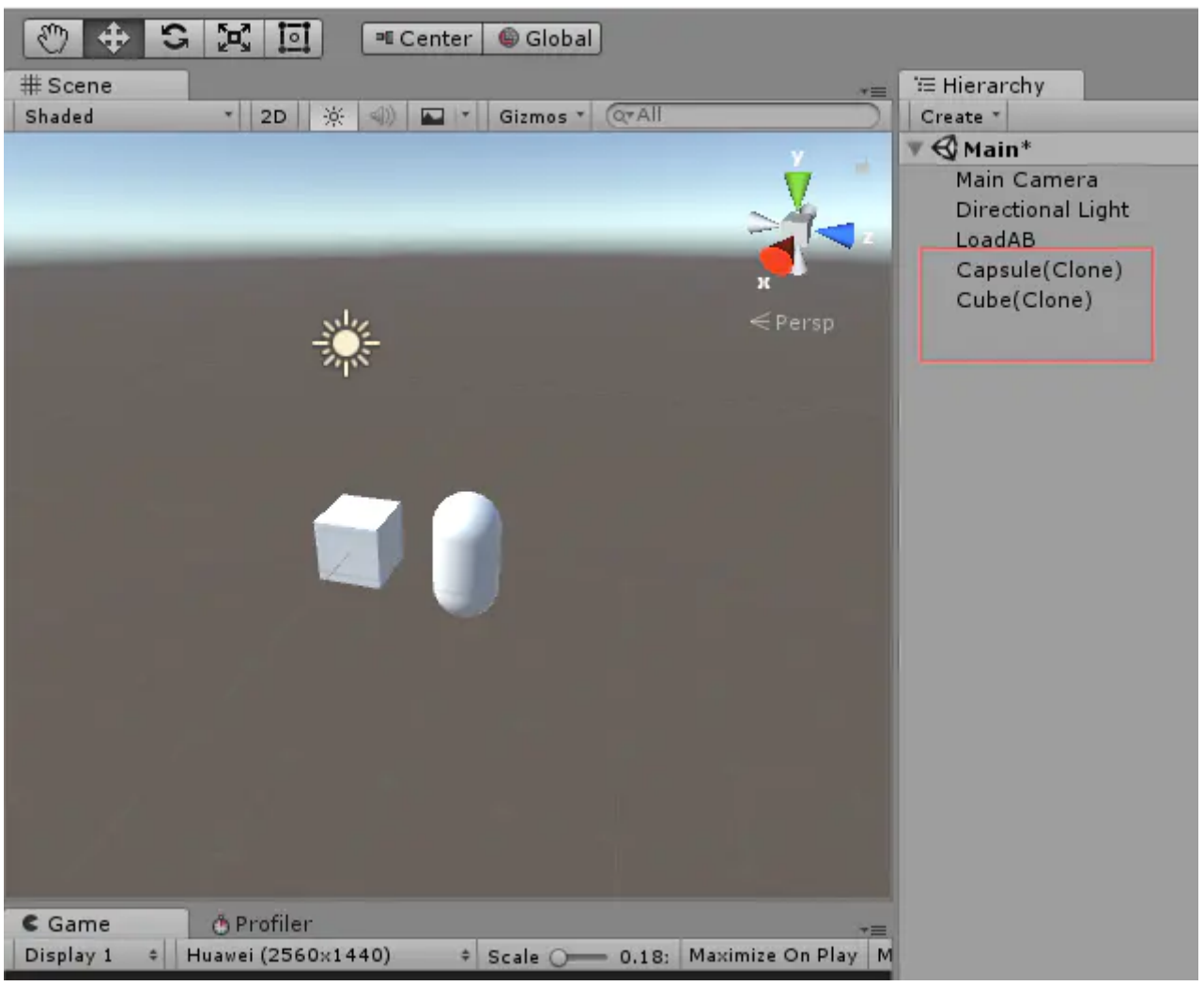
```
using UnityEngine;
using System.Collections;
using UnityEngine.Networking;

public class LoadFromFileExample : MonoBehaviour {

    IEnumerator  Start () {
        //第三種加載方式   使用UnityWbRequest  服務器加載使用http本地加載使用file
        //string uri =
@"file:///C:\Users\Administrator\Desktop\AssetBundleProject\AssetBundles\model.ab";
        string uri = @"http://localhost/AssetBundles\model.ab";
        UnityWebRequest request = UnityWebRequest.GetAssetBundle(uri);
        yield return request.Send();
        AssetBundle ab = DownloadHandlerAssetBundle.GetContent(request);

        //使用裏面的資源
      Object[] obj = ab.LoadAllAssets<GameObject>();//加載出來放入數組中
        // 創建出來
        foreach (Object o in obj)
        {
            Instantiate(o);
        }
    }
}
```

這樣我們Model包裏面的資源就加載出來並創建在場景裏了。這時候運行Unity就可以看到兩個模型都各自創建了出來



Paste_Image.png

當然也可以創建指定的資源例如

```
    AssetBundle ab=AssetBundle.LoadFromFile("AssetBundles/scene/model.ab");
      GameObject go = ab.LoadAsset<GameObject>("Cube");
      Instantiate(go)
```

這樣就實現了Asset Bundle資源的加載了

1，把經常更新的資源放在一個單獨的包裹面，跟不經常更新的包分離

2，把需要同時加載的資源放在一個包裹面

3，可以把其他包共享的資源放在一個單獨的包裹面

4，把一些需要同時加載的小資源打包成一個包

5，如果對於一個同一個資源有兩個版本，可以考慮通過後綴來區分 v1 v2 v3 unity3dv1 unity3dv2

1，邏輯實體分組

a,一個UI界面或者所有UI界面一個包（這個界面裏面的貼圖和佈局信息一個包）

b,一個角色或者所有角色一個包（這個角色裏面的模型和動畫一個包）

c,所有的場景所共享的部分一個包（包括貼圖和模型）

2，按照類型分組

所有聲音資源打成一個包，所有shader打成一個包，所有模型打成一個包，所有材質打成一個包

3，按照使用分組

把在某一時間內使用的所有資源打成一個包。可以按照關卡分，一個關卡所需要的所有資源包括角色、貼圖、聲音等打成一個包。也可以按照場景分，一個場景所需要的資源一個包

依賴打包

意思就是例如有兩個模型使用的都是同一個材質和貼圖，那麼模型和材質貼圖之間就是依賴關係。如果我們這兩個模型都單獨打包出來那麼就會打包出兩份材質和貼圖，這樣包就會變大，那麼我們如何解決呢，這裏Unity裏面自帶有一種方式，那就是首先先把所依賴的材質和貼圖單獨打包到一個文件夾中，然後再分別打包兩個需要依賴這個材質和貼圖的模型。這樣Unity就會去查找這個材質貼圖，發現這個材質和貼圖已經打包了出來，那麼它就不會去重複的打包材質和貼圖了，這樣就大大減小了包的大小

Paste_Image.png

上面一個是直接兩個模型分別打包出來可以看到材質和貼圖都分別打包了出來，分別都是63KB,而下面的是先將材質和貼圖打包出來是62KB再將兩個2KB的模型打包出來，總共也才64KB。

Paste_Image.png

這就是Unity自帶的依賴打包.
但是加載的時候模型、材質和貼圖都要進行去加載，不然就會出現財政的丟失

```
using UnityEngine;

public class LoadFromFileExample : MonoBehaviour {

    void Start () {
        AssetBundle ab = AssetBundle.LoadFromFile("AssetBundles/cube.ab");
        AssetBundle abShare = AssetBundle.LoadFromFile("AssetBundles/share.ab");
        //GameObject go = ab.LoadAsset<GameObject>("Cube");
        //Instantiate(go);
        Object[] obj = ab.LoadAllAssets<GameObject>();//加載出來放入數組中
        //創建出來
        foreach (Object o in obj)
        {
            Instantiate(o);
        }
    }
}
```

*使用**AssetBundleManifest**獲取所有的包*

而AssetBundles.manifest是一個文本文件，裏面有一些包的信息，下面可以打開看看

Paste_Image.png

所以下面我們就可以去讀取到AssetBundles然後獲取裏面所有的AssetBundle包

在Start方法裏面寫入

```
    AssetBundle manifesAB = AssetBundle.LoadFromFile("AssetBundles/AssetBundles");
       AssetBundleManifest manifest= manifesAB.LoadAsset<AssetBundleManifest>
("AssetBundleManifest");
       foreach (string name in manifest.GetAllAssetBundles())
       {
           print(name);
       }
```

這時候運行Unity就可以看到所有包都完整的輸出出來了。

Paste_Image.png

利用Manifest加載某個包所依賴的包

```
       AssetBundle manifesAB = AssetBundle.LoadFromFile("AssetBundles/AssetBundles");
       AssetBundleManifest manifest= manifesAB.LoadAsset<AssetBundleManifest>
("AssetBundleManifest");
       foreach (string name in manifest.GetAllAssetBundles())
       {
           print(name);
       }
       string []strs=manifest.GetAllDependencies("Cube.ab");
       foreach (var name in strs)
       {
           AssetBundle.LoadFromFile("AssetBundles/"+name);
       }
```

運行後可以看到所依賴的Share包的資源也加載了出來

Paste_Image.png

AssetBundle的卸載

卸載有兩個方面

1，減少內存使用

2，有可能導致丟失

所以什麼時候去卸載資源

AssetBundle.Unload(true)卸載所有資源，即使有資源被使用着

（1，在關卡切換、場景切換的時候

（2，資源沒被調用的時候

AssetBundle.Unload(false)卸載所有沒用被使用的資源

個別資源怎麼卸載

(1，通過 Resources.UnloadUnusedAssets.

(2，場景切換的時候

文件校驗

個算法去生成這個文件的校驗碼，然後拿這個值和A傳輸給我的校驗碼比對，如果一樣說明這個文件是完整的，如果不一樣那麼就重新傳輸。下面是幾個算法生成的校驗值

CRC MD5 SHA1

相同點：

CRC、MD5、SHA1都是通過對數據進行計算，來生成一個校驗值，該校驗值用來校驗數據的完整性。

不同點：

1. 算法不同。CRC採用多項式除法，MD5和SHA1使用的是替換、輪轉等方法；
2. 校驗值的長度不同。CRC校驗位的長度跟其多項式有關係，一般爲16位或32位；MD5是16個字節（128位）；SHA1是20個字節（160位）；
3. 校驗值的稱呼不同。CRC一般叫做CRC值；MD5和SHA1一般叫做哈希值（Hash）或散列值；
4. 安全性不同。這裏的安全性是指檢錯的能力，即數據的錯誤能通過校驗位檢測出來。CRC的安全性跟多項式有很大關係，相對於MD5和SHA1要弱很多；MD5的安全性很高，不過大概在04年的時候被山東大學的王小云破解了；SHA1的安全性最高。
5. 效率不同，CRC的計算效率很高；MD5和SHA1比較慢。
6. 用途不同。CRC一般用作通信數據的校驗；MD5和SHA1用於安全（Security）領域，比如文件校驗、數字簽名等。

Unity Asset Bundle Browser tool

這是一個AssetBundle的查看工具，是Unity官方發佈的一個擴展工具，可以查看幫助打包AssetBundle和查看AssetBundle內容。可以去GitHub上下載

https://github.com/Unity-Technologies/AssetBundles-Browser

下載後直接將裏面的Editor擴展工具拖入我們的Unity Project工程中

Paste_Image.png

然後再窗口Window下找到並選擇AssetBundle Browser選項，就可以打開看到我們AssetBundle 窗口了

Paste_Image.png

Paste_Image.png

這是一個輕量級的AssetsBundle使用工具，裏面可以打包可以查看打包的內容可以刪除打包的內容，非常好用

發表評論

登錄以後才評論...

登 录

所有評論

還沒有人評論，想成為第一個評論的人麼? 請在上方評論欄輸入並且點擊發布.

相關文章

Python實現經典小遊戲貪食蛇-趣玩Python系列三

前言： 上一篇已採用pygame做了一個Python實現黑客帝國代碼雨-趣玩Python系列二的效果，今天升級一下，來實現一個經典小遊戲-貪食蛇吧。 首先我們需要導入待使用的模塊： import pygame, sys, random

明哥看世界 2020-07-08 05:11:00

U3D菜單中文註解

File　文件： ----New Scene　新場景 ----Open Scene　打開場景 ----Save Scene　保存場景 ----Save Scene as　另存爲 ----New Project

禾子续 2020-07-08 07:48:55

【U3D/簡單框架】6.UI模塊

自我介紹 廣東雙非一本的大三小白，計科專業，想在製作畢設前夯實基礎，畢設做出一款屬於自己的遊戲！ UI模塊 UI基類 BasePanel.cs，以後子類只需要繼承這個基類，使用如圖 一定一定一定要注意泛型的約束 where T:

神经大爆炸 2020-07-08 07:23:31

功能簡單的吞噢蛇（shell編程)

天老師叫我們做了個吞噢蛇，只實現了很簡單的功能，不廢話，上馬： #!/bin/bash trap input_key=0 20 trap input_key=1 21 trap input_key=2 22 trap input

z0203153008 2020-07-08 06:36:40

程序員中的奇葩，使用php構建魔獸世界

這是用PHP編寫的魔獸世界服務器。 現在它已經調試了登錄服務器的過程。 目前的魔獸世界客戶端是2.4.3_8606。服務器列表和帳戶密碼數據需要查詢AUTH庫。 世界服務器身份驗證過程已完成，數據包加密已完成 後續進程正在開發

明哥看世界 2020-07-08 05:11:00

《甩了，甩了，甩了他》----轉

美國情感作家葛瑞哥在他的新書《甩了，甩了，甩了他》裏寫道，停止悲傷吧，一個假裝在人間蒸發的男人沒那麼值得懷念。你就當他死了。 你沉迷的不是愛情，

cubijing 2020-07-08 03:32:42

Android遊戲開發之繪畫旋轉的物體（Matrix類的邏輯異常）

Android遊戲開發中可能有一些旋轉的物體：例如旋轉的地球，花朵，齒輪等零部件，或者背景，甚至承載重要邏輯的對象。 如果由像素點構成，邏輯變化，當然是一種思路，但是這樣很難有很好的視覺效果，還牽扯到重要的底層算法的實現。 但是如果想讓一

zy19980116 2020-07-08 00:56:41

真的好辛苦

資料那麼少 錯誤那麼多 基礎那麼差 網速那麼慢 登陸那麼慢 遊戲那麼脆弱 你丫的就不能不崩潰麼 你知道登陸一次要花多長時間麼 我日你奶奶的　22:15　　invoke GetDlgItem,hWnd1,1

huangjunfengok 2020-07-08 00:23:43

以前用java寫的貪喫蛇遊戲

　/*　*遊戲的主畫布類　*/import javax.swing.*;import java.awt.*;public class GameCanvas extends JPanel{ private int rows = 30,

lsrj 2020-07-07 20:12:14

AgoBot 殭屍網絡研究筆記（十五）

十五、2008年04月09日　作者：青青子衿 email：anzijin@sina.com 1、CCDKeyGrab的作用是處理cdkey.get消息，功能是獲取被控端，下述遊戲的cdkey的值 遊戲列表：（1）、Hal

anzijin 2020-07-07 17:54:06

09.2.5

因爲無法忍受冠捷液晶電視一個星期左右才能到貨的等待，我早上殺至中關村選電視。先看帶TV功能的液晶顯示器，這類產品比較少且價格不低。看中三星的一款和LG的一款，價格差不多都1700多，感覺三星的2232MW外觀非常漂亮功能也很強大，可是價格

crespo5454 2020-07-07 17:44:00

Cocos2d-x 與 ISO 內存管理

一，IOS與圖片內存 在IOS上，圖片會被自動縮放到2的N次方大小。比如一張1024*1025的圖片，佔用的內存與一張1024*2048的圖片是一致的。圖片佔用內存大小的計算的公式是：長*寬*4。這樣一張512*512 佔用的內存就是

游戏码头 2020-07-07 14:14:08

這個社會最大的現實是“大魚喫小魚，小魚喫蝦米”

　認真看到最後的人，纔是勇者和朋友。呵呵……　　今天看英語有點想睡，給了自己兩巴掌還是想睡。於是我到處亂看，無意中在360軟

欢乐雅子 2020-07-07 12:53:12

大學畢業生們的25種經典總結

1、大學仍在，只是我在飛逝，畢業後站在校門口一看，才知道熟悉也可以變成一種模糊地映像 2、 以前吧也曾以爲大學吧就是用青春來追夢的過程，畢業了才知道我的青春沒了，夢早就沒了 3、 人生最可怕的就是理想迷失了方向，可是我卻連理想都沒有 4、

gzdqdb 2020-07-07 10:47:14