Unity中的各种path

2021年9月15日 14:37

目录

dataPath\streamingAssetsPath\persistantDataPath\temporaryCachePath
学习一下这些路径的具体含义、区别，在不同平台下的情况，以及相关的一些用法。

streamingAssetPath (页首)

Refs:
https://docs.unity3d.com/Manual/StreamingAssets.html
https://docs.unity3d.com/Manual/SpecialFolders.html
https://docs.unity3d.com/ScriptReference/Application-streamingAssetsPath.html
https://blog.csdn.net/Wenbooboo/article/details/98976493

为啥需要streamingAssetPath (页首)

Unity combines most Assets into a Project when it builds the Project. However, it is sometimes useful to place files into the normal filesystem on the target machine to make them accessible via a pathname. An example of this is the deployment of a movie file on iOS devices; the original movie file must be available from a location in the filesystem to be played by the `PlayMovie` function.

streamingAssetsPath在不同平台下的实际路径有所不同 (页首)

The location returned by `Application.streamingAssetsPath` varies per platform:

- Most platforms (Unity Editor, Windows, Linux players, PS4, Xbox One, Switch) use `Application.dataPath + "/StreamingAssets"`,
- macOS player uses `Application.dataPath + "/Resources/Data/StreamingAssets"`,
- iOS uses `Application.dataPath + "/Raw"`,
- Android uses files inside a compressed APK/JAR file, `"jar:file://" + Application.dataPath + "!/assets"`.

streamingAssetsPath下的文件如何读取 (页首)

To read streaming Assets on platforms like Android and WebGL, where you cannot access streaming Asset files directly, use UnityWebRequest. For an example, see Application.streamingAssetsPath.

放置在streamingAssets文件夹下的视频不能用clip的方式放到VideoPlayer组件上，需要走URL方式。

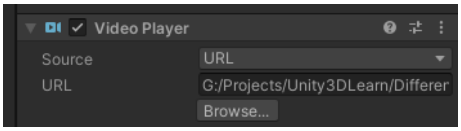
```
0 个引用
void Start()
{
    GameObject cam = GameObject.Find("Main Camera");
    videoPlayer = cam.AddComponent<UnityEngine.Video.VideoPlayer>();

    Debug.Log("streamingAssetsPath = " + Application.streamingAssetsPath);
    // Obtain the location of the video clip.
    videoPlayer.url = Path.Combine(Application.streamingAssetsPath, "test1.mp4");

    // Restart from beginning when done.
    videoPlayer.isLooping = true;

    // Do not show the video until the user needs it.
    videoPlayer.Pause();

    status = "Press to play";
}
```



局限和建议 (页首)

On many platforms, the streaming assets folder location is read-only; you can not modify or write new files there at runtime. Use Application.persistentDataPath for a folder location that is writable.

Note: .dll and script files located in the **StreamingAssets** folder don't participate in the script compilation.

Asset Bundles or Addressables are alternative ways of accessing content that is not part of regular game build data, and in many cases are preferred over Streaming Assets folder.

streamingAssets文件夹只读，且在安卓平台的路径为
jar:file:///data/app/com.DefaultCompany.DifferentPaths-2/base.apk!/assets
不能用System.IO.File来操作。

实际应用 (页首)

链接分享这边有个需求是需要**把一张icon图片的文件路径传给sdk，这个文件就需要在系统文件夹下**，所以没法使用StreamingAssets文件加下的内容，这里因为打包流程中会清空StreamingAssets文件夹，不想改这个流程代码，就采用了如下方式：
运行时保存Resources下的贴图到手机，Resources下贴图被压缩过，需要解压后才能保存到persitantDataPath中。

```
/// <summary>
/// 解压一个贴图，使其能够被保存文件
/// refs: https://stackoverflow.com/questions/51315918/how-to-encodetopng-compressed-textures-in-unity
/// </summary>
/// <param name="source"></param>
/// <returns></returns>
1 个引用
public static Texture2D DeCompress(Texture2D source)
{
    RenderTexture renderTex = RenderTexture.GetTemporary(
                source.width,
                source.height,
                0,
                RenderTextureFormat.Default,
                RenderTextureReadWrite.Linear);

    Graphics.Blit(source, renderTex);
    RenderTexture previous = RenderTexture.active;
    RenderTexture.active = renderTex;
    Texture2D readableText = new Texture2D(source.width, source.height);
    readableText.ReadPixels(new Rect(0, 0, renderTex.width, renderTex.height), 0, 0);
    readableText.Apply();
    RenderTexture.active = previous;
    RenderTexture.ReleaseTemporary(renderTex);
    return readableText;
}
```

```
def share_icon_prepare(self):
    """
    安卓平台分享链接使用
    拷贝Rescource/Images/AppIcon/AppIcon76x76@2x~ipad.png到
    persistentDataPath(/storage/emulated/0/Android/data/com.netease.hana/files)下
    """
    # 非安卓平台无需拷贝
    if not config.is_android():
        return
    target_path = Application.get_persistentDataPath() + "/AppIcon76x76@2x~ipad.png"
    # 判断是否已经有这张图片了
    if os.path.exists(target_path):
        return
    icon_path = "Images/AppIcon/AppIcon76x76@2x~ipad"
    sprite = Resources.Load(icon_path, Sprite)
    texture_2d = sprite.texture
    readable_texture_2d = GameUtil.DeCompress(texture_2d)    # 获取解压后的贴图，使其能够被保存文件
    bytes = ImageConversion.EncodeToPNG(readable_texture_2d)
    try:
        with open(target_path, "wb") as f:
            f.write(bytes)
    except Exception as e:
        from hexm.common.util import util
        util.on_traceback()
        return False
```

如果StreamingAssetPath文件夹可用的话，**也可以将贴图放到StreamingAssetsPath下，然后用UnityWebRequest的方式读取**，并File.WriteAllBytes到**persistantDataPath**中，这里就不需要解压了，因为StreamingAssetsPath下的文件还保持着原来的格式。

You may want the Asset to be available as a separate file in its original format (although it's more common to directly incorporate Assets into a build). For example, you need to access a video file from the filesystem to play the video on **IOS** using Handheld.PlayFullScreenMovie.

To include streaming Assets, do the following:

1. Place the file in the StreamingAssets folder.
2. The file remains unchanged when copied to the target machine, where it's available from a specific folder.

persistantDataPath (页首)

Refs: https://docs.unity3d.com/ScriptReference/Application-persistentDataPath.html

为啥需要persistantDataPath (页首)

可以将一些数据保存在这里，这里的文件可以System.IO.File接口进行读写；
且这些数据在关闭并重进游戏的时候不会被改变。

This value is a directory path where you can store data that you want to be kept between runs. When you publish on iOS and Android, persistentDataPath points to a public directory on the device. Files in this location are not erased by app updates. The files can still be erased by users directly.

When you build the Unity application, a GUID is generated that is based on the Bundle Identifier. This GUID is part of persistentDataPath. If you keep the same Bundle Identifier in future versions, the application keeps accessing the same location on every update.

persistantDataPath在不同平台下的实际路径有所不同 (页首)

Windows Store Apps: Application.persistentDataPath points to `%userprofile%\AppData\Local\Packages\<productname>\LocalState`.

Windows Editor and Standalone Player: Application.persistentDataPath usually points to `%userprofile%\AppData\LocalLow\<companyname>\<productname>`. It is resolved by SHGetKnownFolderPath with FOLDERID_LocalAppDataLow, or SHGetFolderPathW with CSIDL_LOCAL_APPDATA if the former is not available.

WebGL: Application.persistentDataPath points to `/idbfs/<md5 hash of data path>` where the data path is the URL stripped of everything including and after the last '/' before any '?' components.

Linux: Application.persistentDataPath points to `$XDG_CONFIG_HOME/unity3d` or `$HOME/.config/unity3d`.

iOS: Application.persistentDataPath points to `/var/mobile/Containers/Data/Application/<guid>/Documents`.

tvOS: Application.persistentDataPath is not supported and returns an empty string.

Android: Application.persistentDataPath points to `/storage/emulated/0/Android/data/<packagename>/files` on most devices (some older phones might point to location on SD card if present), the path is resolved using android.content.Context.getExternalFilesDir.

Mac: Application.persistentDataPath points to the user Library folder. (This folder is often hidden.) In recent Unity releases user data is written into `~/Library/Application Support/company name/product name`. Older versions of Unity wrote into the `~/Library/Caches` folder, or `~/Library/Application Support/unity.company name.product name`. These folders are all searched for by Unity. The application finds and uses the oldest folder with the required data on your system.

dataPath (页首)

Refs: https://docs.unity3d.com/ScriptReference/Application-dataPath.html

Contains the path to the game data folder on the target device (Read Only).

The value depends on which platform you are running on:

Unity Editor: <*path to project folder*>/Assets

Mac player: <*path to player app bundle*>/Contents

iOS player: <*path to player app bundle*>/<*AppName.app*>/Data (this folder is read only, use Application.persistentDataPath to save data).

Win/Linux player: <*path to executablename_Data folder*> (note that most Linux installations will be case-sensitive!)

WebGL: The absolute url to the player data file folder (without the actual data file name)

Android: Normally it points directly to the APK. If you are running a split binary build, it points to the OBB instead.

Windows Store Apps: The absolute path to the player data folder (this folder is read only, use Application.persistentDataPath to save data)

Note that the string returned on a PC will use a forward slash as a folder separator.

temporaryCachePath (页首)

安卓端：
/storage/emulated/0/Android/data/com.DefaultCompany.DifferentPaths/cache
Windows端：
C:/Users/XUHONG~1/AppData/Local/Temp/DefaultCompany/DifferentPaths

注意 (页首)

this folder may or may not be cleared between runs

来自 < https://forum.unity.com/threads/saving-application-data.239298/?_ga=2.196003816.596601032.1631699880-83447172.1631699880#post-1587347>

PlayerPrefs (页首)

用途 (页首)

方便的存储一些基础数据，如int\string等，存到对应平台的注册表中。

`PlayerPrefs` is a class that stores Player preferences between game sessions. It can store string, float and integer values into the user's platform registry.

Unity stores `PlayerPrefs` data differently based on which operating system the application runs on. In the file paths given on this page, the company name and product name are the names you set in Unity's Player Settings.

路径因平台而异。

需要注意 (页首)

不要存放敏感数据，因为没有经过加密，可以被轻松获取。

Unity stores PlayerPrefs in a local registry, without encryption. Do not use PlayerPrefs data to store sensitive data.