

可視化情報学会ワークショップ

チュートリアル1:

Unityハンズオンセミナー お手軽ビジュアライゼーション講座

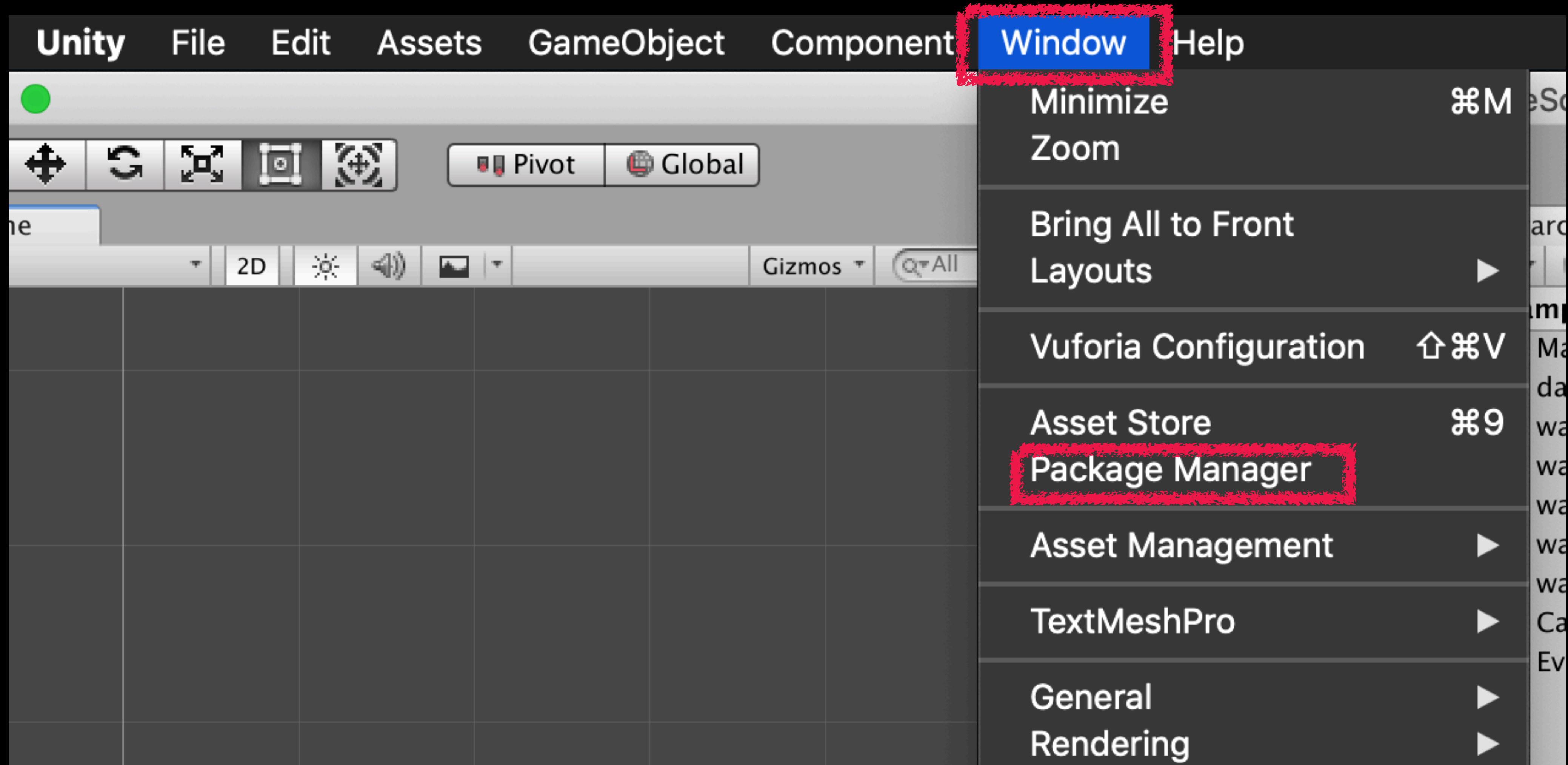
補足資料

ユニティ・テクノロジーズ・ジャパン

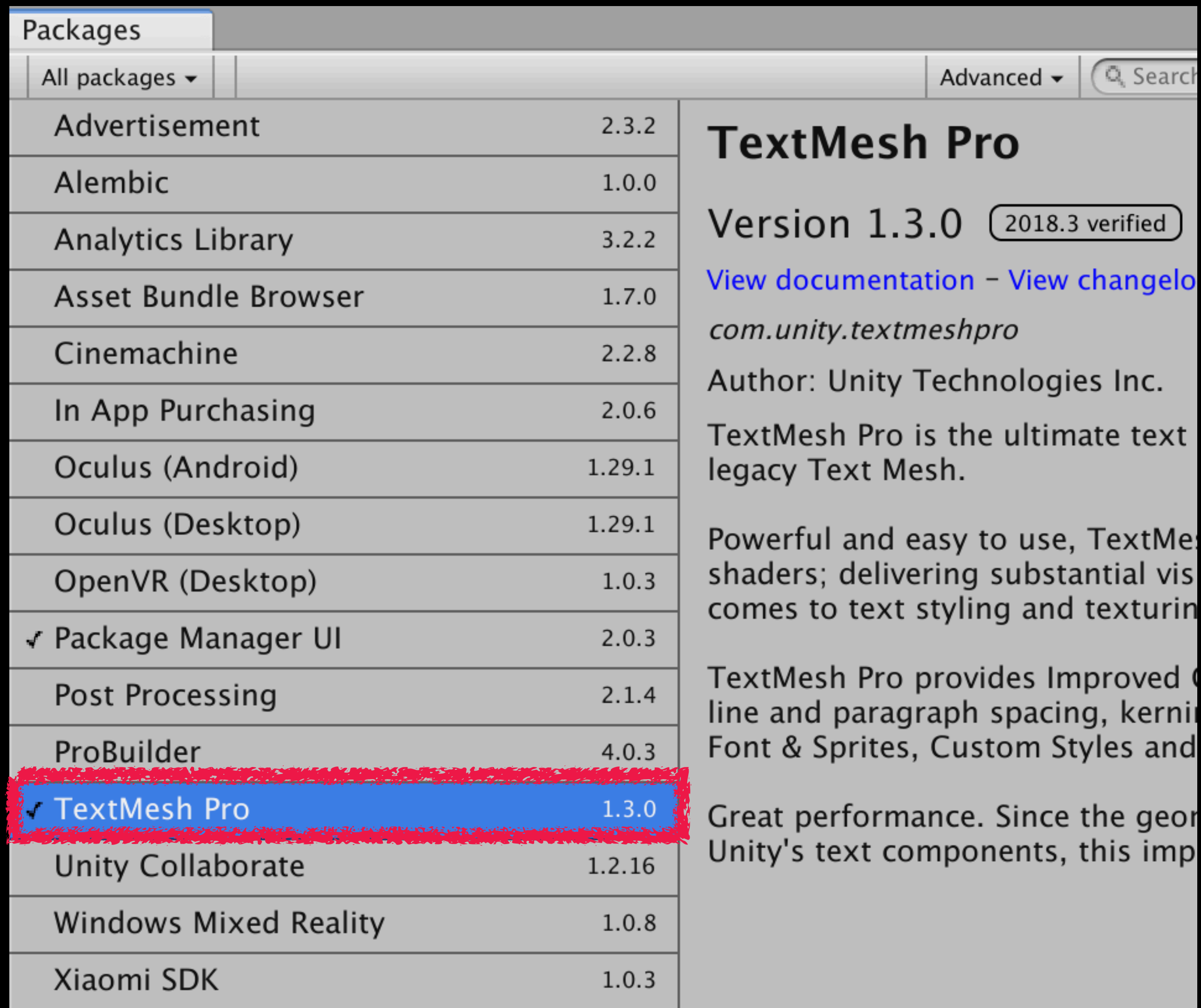
安原 祐二

Package について

Unity の機能追加はパッケージマネージャで



今回は TextMesh Pro をインストール済み



The screenshot shows the Unity Package Manager window. On the left, a list of packages is displayed with their versions. The 'TextMesh Pro' package is highlighted with a blue background and a red border, indicating it is installed. On the right, the details for TextMesh Pro are shown, including the version 1.3.0 (verified), a link to documentation, the author Unity Technologies Inc., and a description of the package's features.

Package Name	Version
Advertisement	2.3.2
Alembic	1.0.0
Analytics Library	3.2.2
Asset Bundle Browser	1.7.0
Cinemachine	2.2.8
In App Purchasing	2.0.6
Oculus (Android)	1.29.1
Oculus (Desktop)	1.29.1
OpenVR (Desktop)	1.0.3
✓ Package Manager UI	2.0.3
Post Processing	2.1.4
ProBuilder	4.0.3
✓ TextMesh Pro	1.3.0
Unity Collaborate	1.2.16
Windows Mixed Reality	1.0.8
Xiaomi SDK	1.0.3

TextMesh Pro

Version 1.3.0 2018.3 verified

[View documentation](#) - [View changelog](#)

com.unity.textmeshpro

Author: Unity Technologies Inc.

TextMesh Pro is the ultimate text legacy Text Mesh.

Powerful and easy to use, TextMesh Pro provides improved text rendering; delivering substantial visual quality improvements to text styling and texturing.

TextMesh Pro provides Improved text rendering, improved line and paragraph spacing, kerning, alignment, and more. Font & Sprites, Custom Styles and more.

Great performance. Since the geometry is generated by Unity's text components, this improves performance.

パッケージはPackages/manifest.jsonに記述される

```
/Users/yuji/Documents/unity_work/VisualizationWorkshopTutorial/Packages
total used in directory 16 available 9223372036848489877
drwxr-xr-x   4 yuji  staff   128 Mar  1 17:35 .
drwxr-xr-x  12 yuji  staff   384 Mar  4 20:47 ..
-rw-r--r--   1 yuji  staff   329 Mar  1 17:35 manifest.json
```

変化はUnityに捕捉され、適切にインストールされる。
このファイルがcommitされているので
gitでcheckoutするだけでTextMesh Proが導入される。

UnityのUpdateについて

コンソールアプリケーションの例

```
int
main(int argc, char* argv[])
{
    Param param(argc, argv);
    bool result = execute(param);
    return result ? 0 : 1;
}
```

ウィンドウアプリケーションの例

```
private void button1_Click(object sender, System.EventArgs e)
{
    bool result = execute();
    Form myForm = button1.FindForm();
    myForm.Text = result ? "success" : "failed";
}
```


Unityの例

```
public class MyScript : MonoBehaviour
{
    void Start ()
    {
        // initialization
    }

    void Update ()
    {
        // called once per frame
    }
}
```

Arduinoの例



```
sketch_apr16a §
```

```
void setup() {  
  // put your setup code here, to run once:  
  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
  
}  
  
|
```

Arduinoのsetupとloopは単純

```
int
main()
{
    setup();
    for (;;) {
        loop();
    }
    return 0;
}
```

Unityはたくさんのおブジェクトがあるので

```
public class MyScript : MonoBehaviour
{
    void Start ()
    {
        // initialization
    }

    void Update ()
    {
        // called once per frame
    }
}
```

Updateに重い処理（ReadFileなど）を書かないこと

コルーチン

ブロック関数を書ける

シーケンスの記述に便利

yield という予約語

```
public class Test : MonoBehaviour {  
    private int count_;  
  
    void Start()  
    {  
        count_ = 100;  
        StartCoroutine(loop());  
    }  
  
    void Update()  
    {  
        --count_;  
    }  
  
    IEnumerator loop()  
    {  
        while (count_ > 0) {  
            yield return null;  
        }  
        Debug.Log("done.");  
    }  
}
```