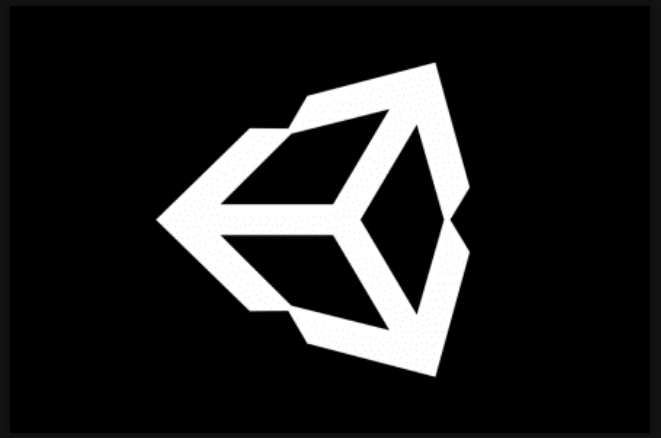
Unity x Live2D 基礎使用講座

分享者: 陳秉凡

報告主題

Live2D Export For Unity

Unity Apply Live2D SDK

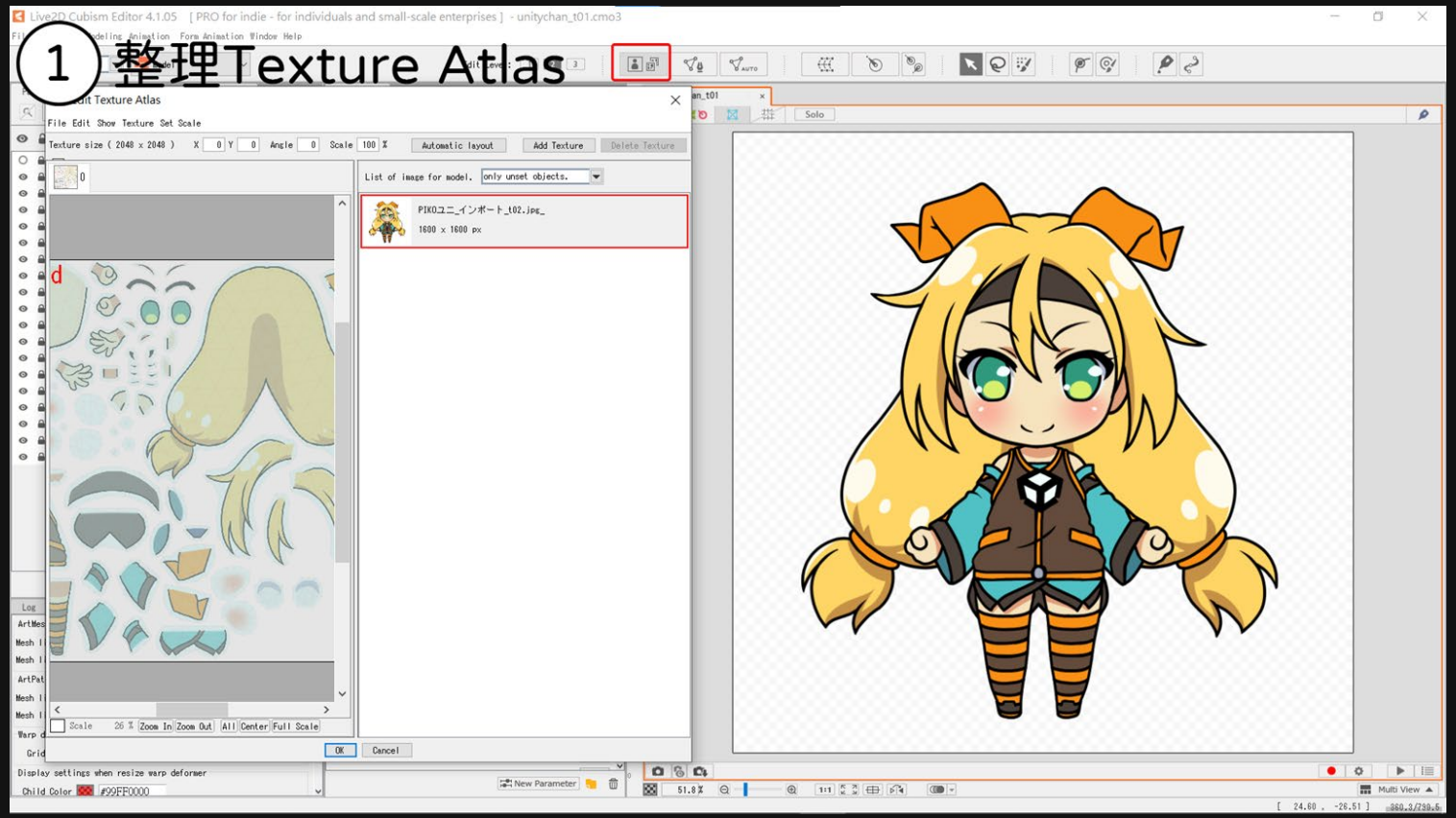


講座大綱

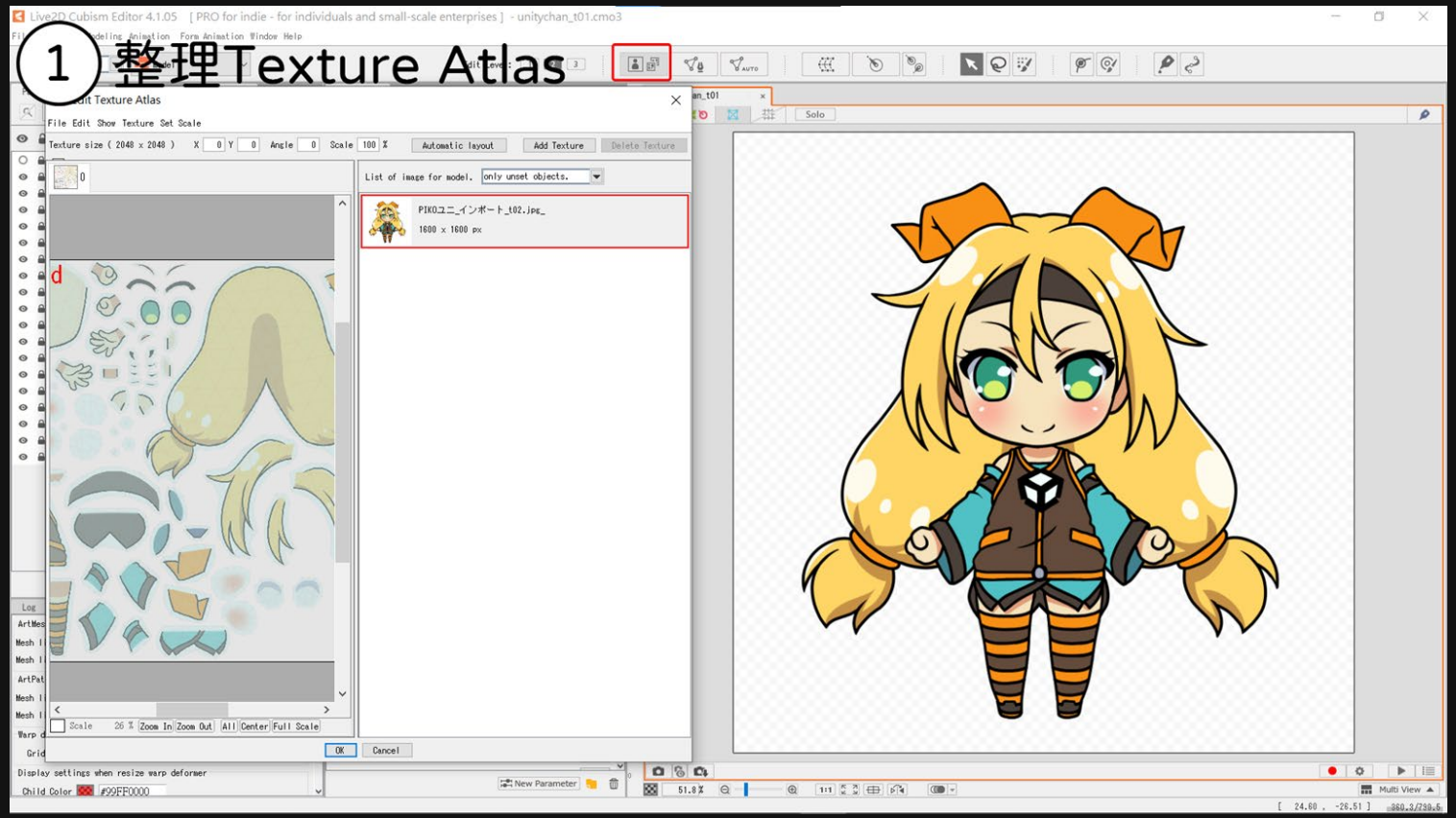
1. Live2D模型導出Runtime
2. Unity安裝Live2D SDK
3. 自動眨眼腳本導入
4. 動畫切換器— 動畫編輯、條件參數、轉場條件
5. 使用C#程式做到動畫切換
6. 補充. 模型滑鼠追蹤腳本導入

事前準備— Live2D導出

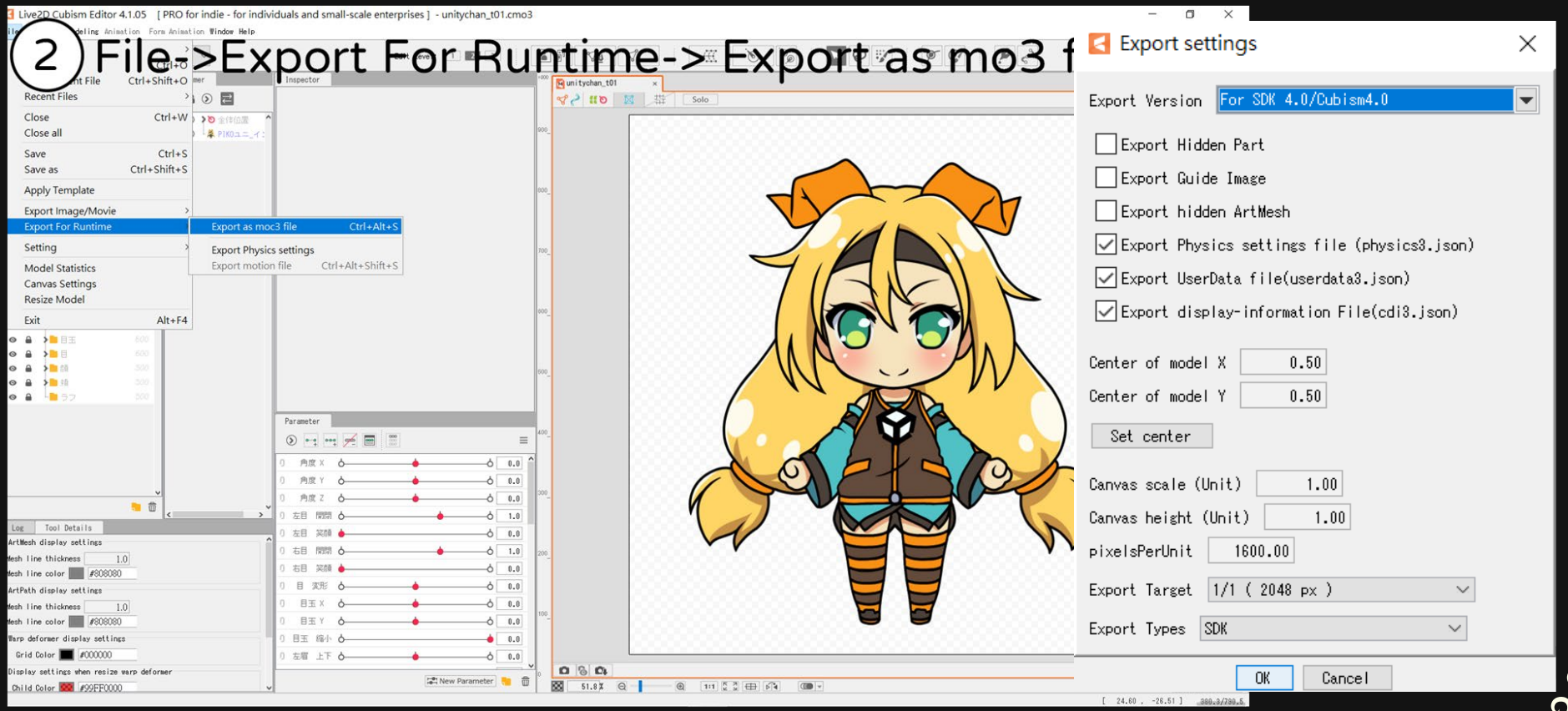
整理部件


1 整理Texture Atlas
Edit Texture Atlas
File Edit Show Texture Set Scale
Automatic layout
Add Texture
Delete Texture
List of image for model.
only unset objects.
PIKOユニ_インポート_t02.jpg_
1600 x 1600 px
OK
Cancel

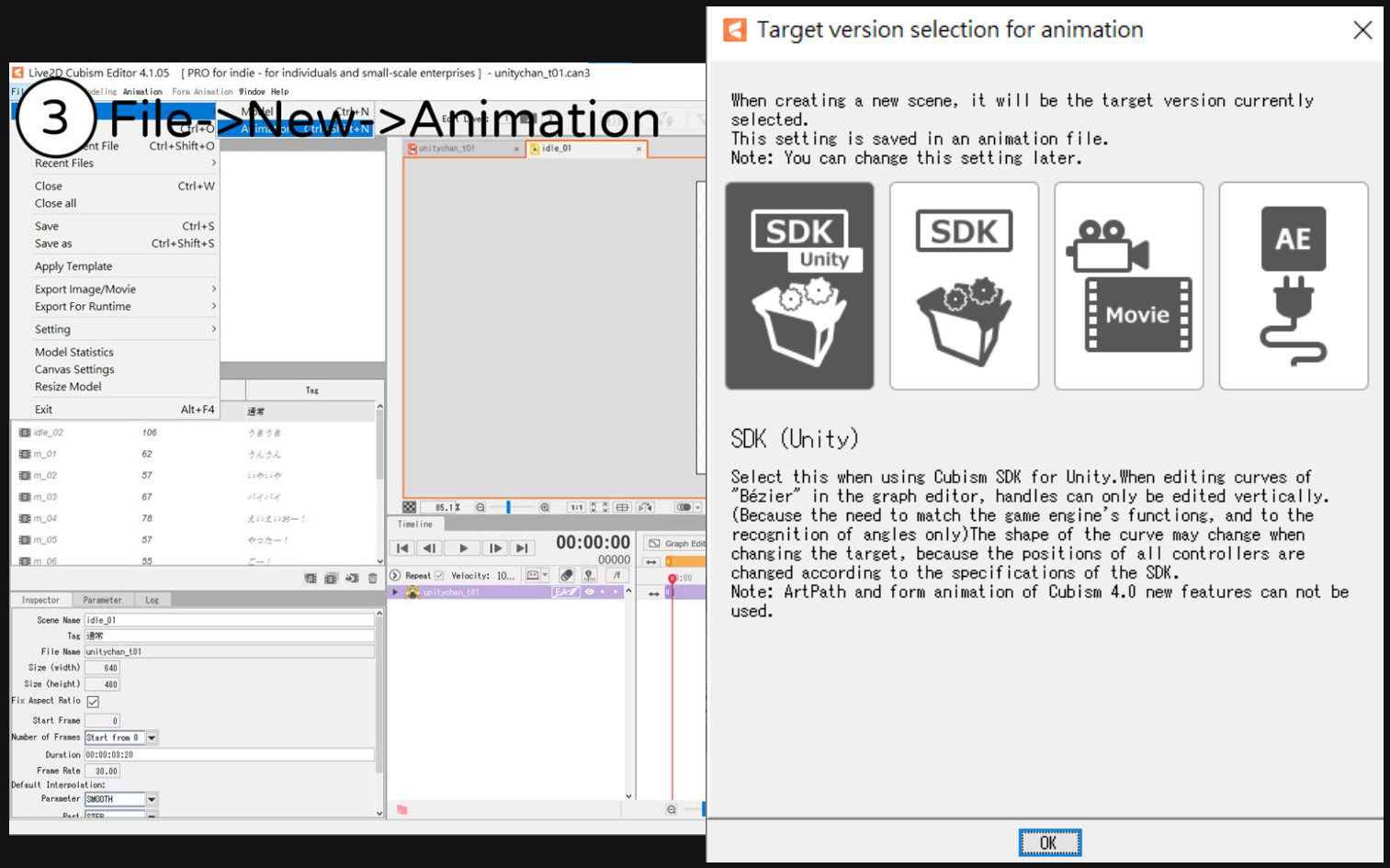
整理部件


1 整理Texture Atlas
Edit Texture Atlas
File Edit Show Texture Set Scale
Automatic layout
Add Texture
Delete Texture
List of image for model.
only unset objects.
PIKOユニ_インポート_t02.jpg_
1600 x 1600 px
OK
Cancel

導出


2 File->Export For Runtime->Export as mo3 f
Live2D Cubism Editor 4.1.05 [PRO for indie - for individuals and small-scale enterprises] - unitychan_t01.cmo3
Recent Files
Close Ctrl+W
Close all
Save Ctrl+S
Save as Ctrl+Shift+S
Apply Template
Export Image/Movie
Export For Runtime
Setting
Model Statistics
Canvas Settings
Resize Model
Exit Alt+F4
Export as moc3 file Ctrl+Alt+S
Export Physics settings
Export motion file Ctrl+Alt+Shift+S
Export settings
Export Version
For SDK 4.0/Cubism4.0
Export Hidden Part
Export Guide Image
Export hidden ArtMesh
Export Physics settings file (physics3.json)
Export UserData file(userdata3.json)
Export display-information File(cdi3.json)
Center of model X 0.50
Center of model Y 0.50
Set center
Canvas scale (Unit) 1.00
Canvas height (Unit) 1.00
pixelsPerUnit 1600.00
Export Target 1/1 (2048 px)
Export Types SDK
OK
Cancel

Animation製作注意



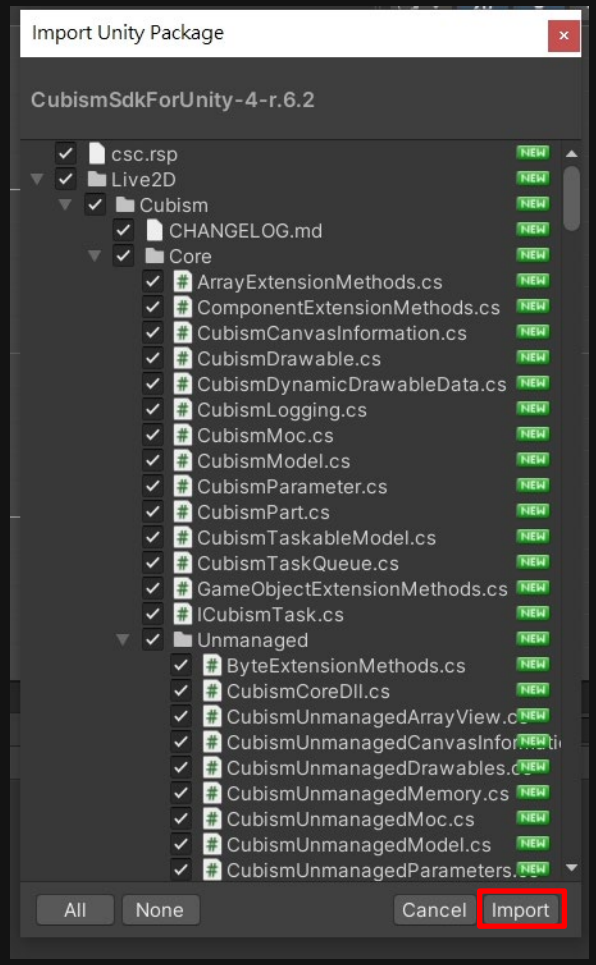
導出成功下應該要有的檔案

Runtime
- ./motion
  - ./ *.json (動畫檔)
- ./[module_name].[texture_px] (Texture Atlas)
- ./ *.json (物理、參數、模型資料)
- ./ *.moc3 (輸出檔)

Live2D導入Unity

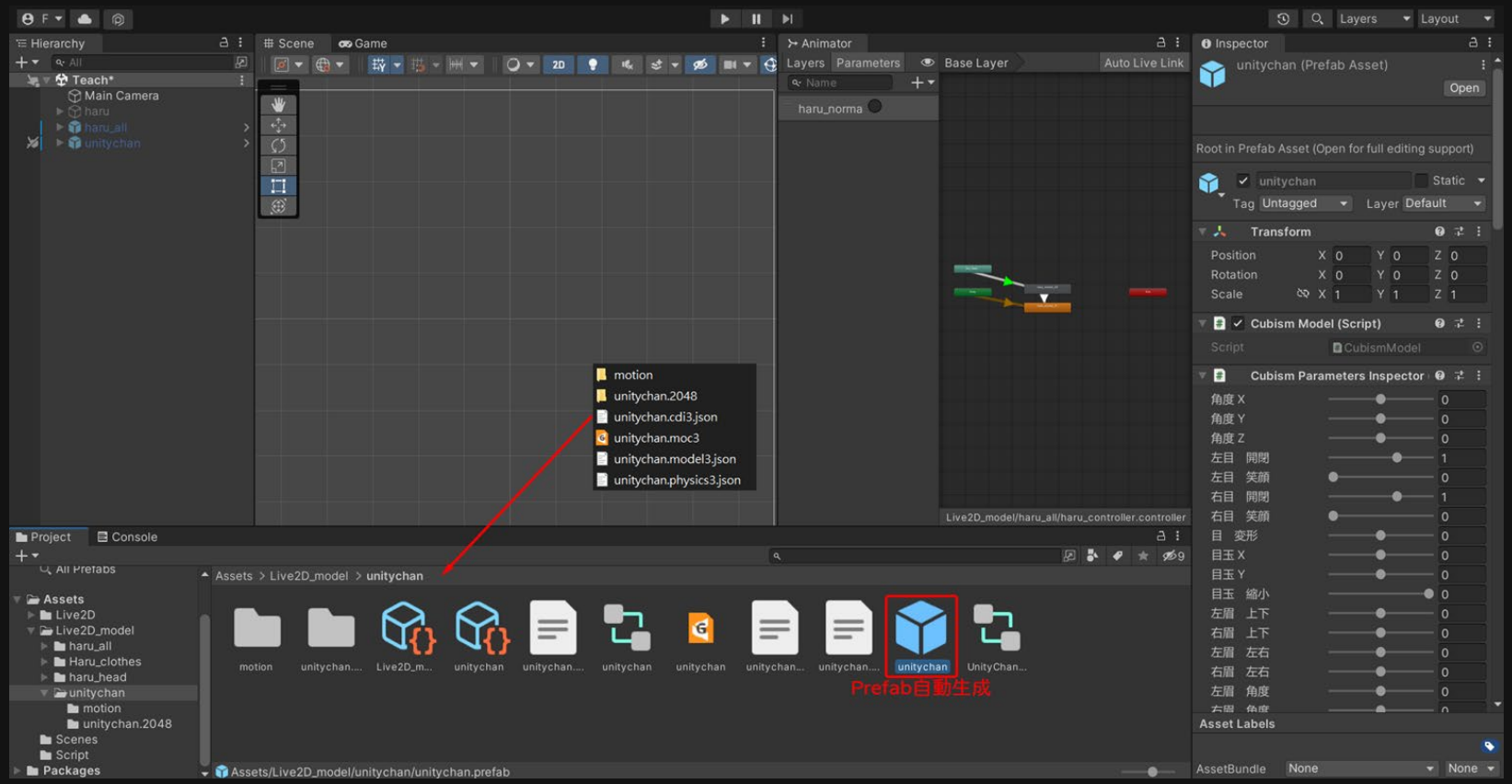
安装Package


CubismSdkForUnity-4-r.6.2.unitypac
kage


Import Unity Package
CubismSdkForUnity-4-r.6.2
csc.rsp
Live2D
Cubism
CHANGELOG.md
Core
ArrayExtensionMethods.cs
ComponentExtensionMethods.cs
CubismCanvasInformation.cs
CubismDrawable.cs
CubismDynamicDrawableData.cs
CubismLogging.cs
CubismMoc.cs
CubismModel.cs
CubismParameter.cs
CubismPart.cs
CubismTaskableModel.cs
CubismTaskQueue.cs
GameObjectExtensionMethods.cs
ICubismTask.cs
Unmanaged
ByteExtensionMethods.cs
CubismCoreDll.cs
CubismUnmanagedMemory.cs
CubismUnmanagedMoc.cs
CubismUnmanagedModel.cs
All
None
Cancel
Import

導入模型

1. 將剛剛Live2D導出的Runtime裡的檔案放入Unity Project
2. 導入後Unity會自動生成Prefab跟Animator Controller
3. 便可把Prefab拖曳至Hierarchy

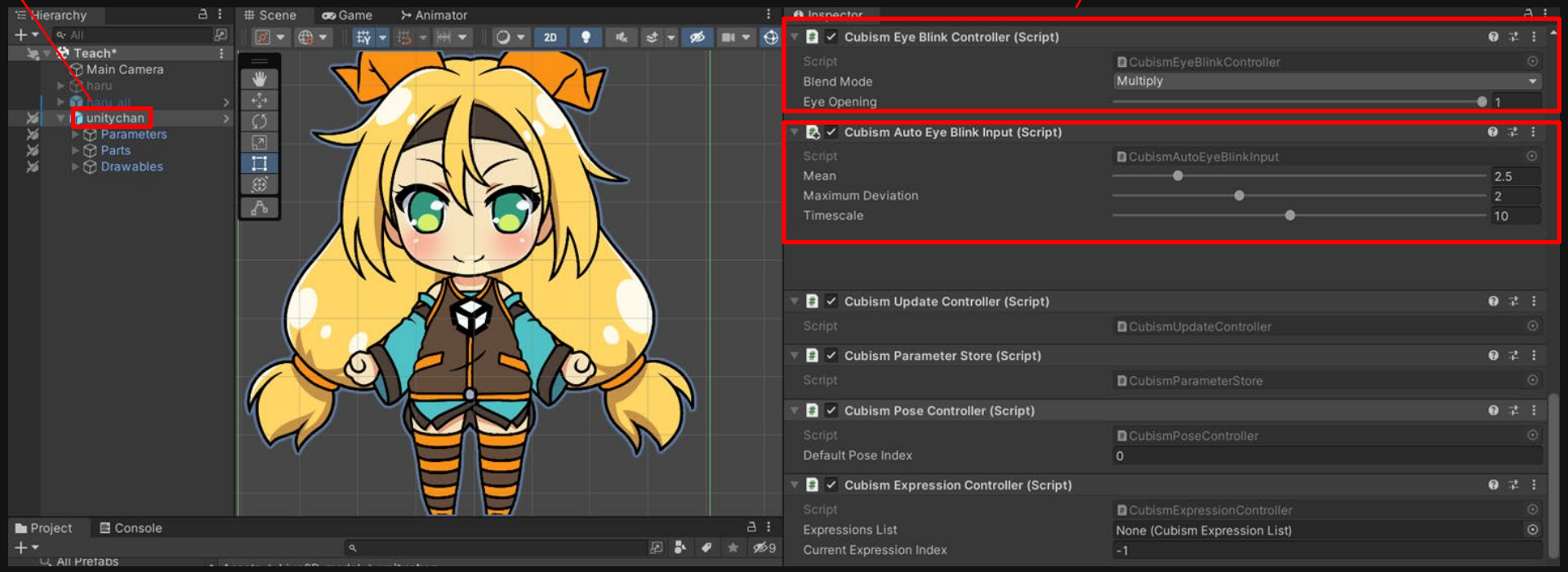


初步測試動畫—眨眼

自動眨眼1

步驟1. 點選使用Prefab生成的Gameobject

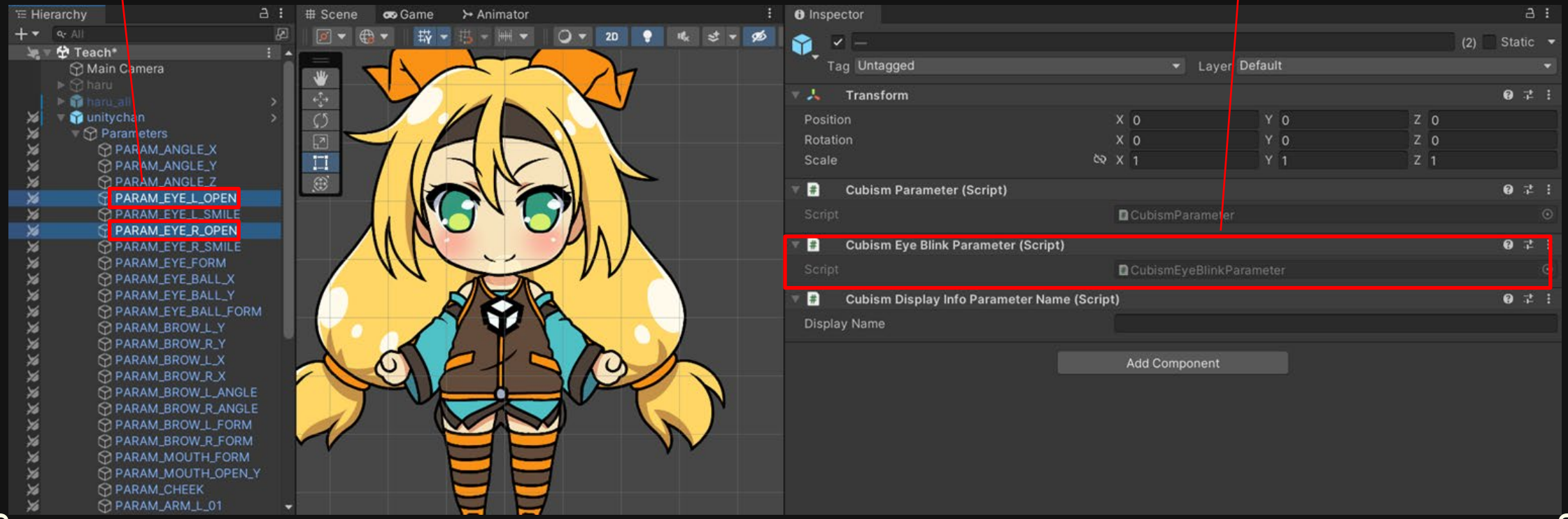
步驟2. 添加以下腳本
- Cubism Eye Blink Controller
- Cubism Auto Eye Blink Input



自動眨眼2

步驟1. 點選模型Gameobject底下的Parameters底下的控制眼睛開關的Gameobject

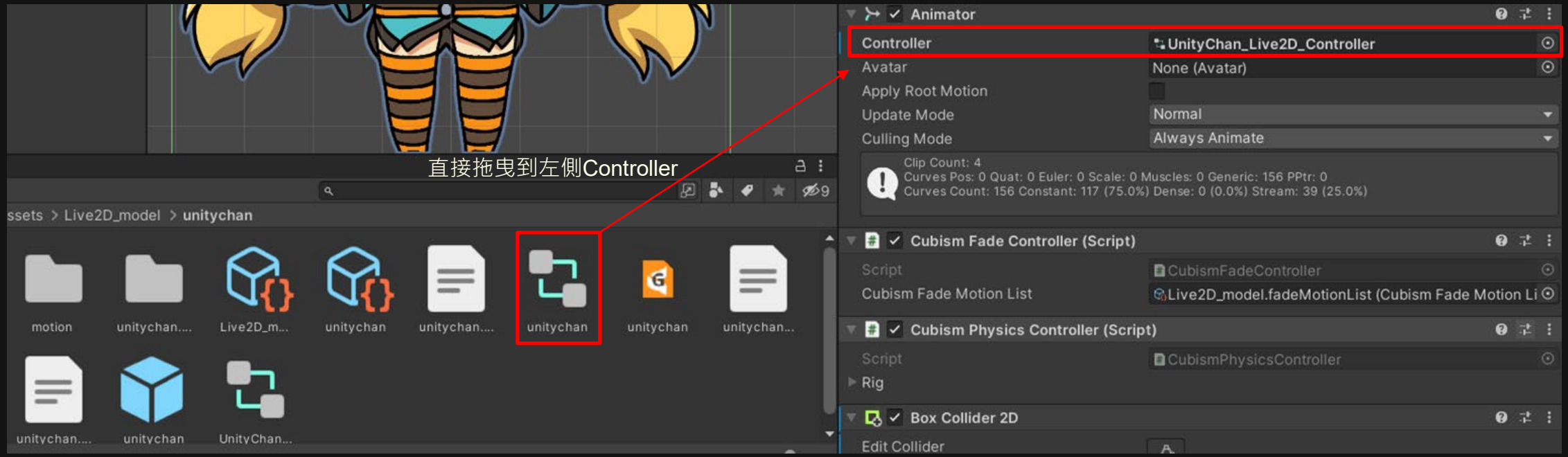
步驟2. 添加以下的腳本
- Cubism Eye Blink Parameter



動畫切換

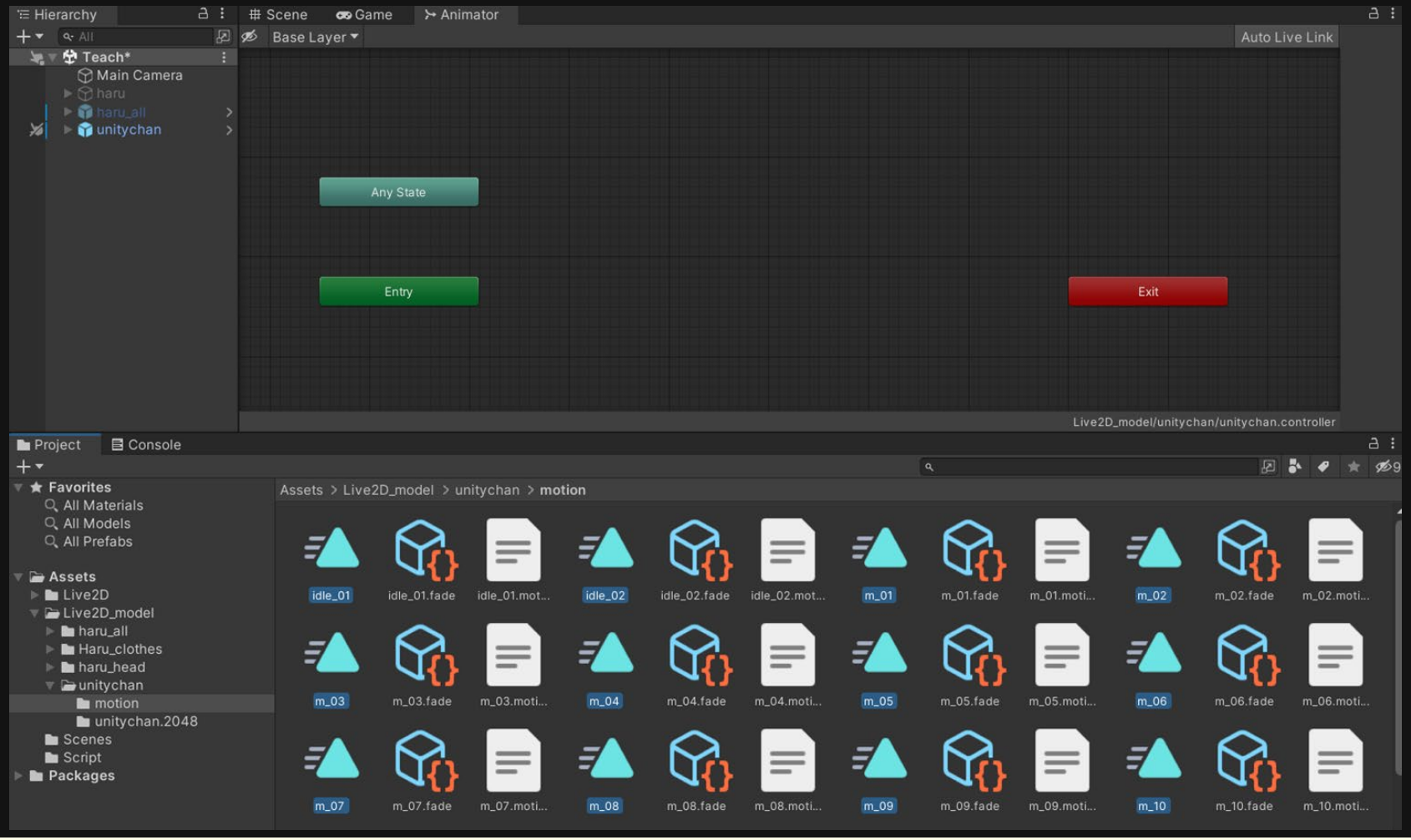
添加Animator Controller

※ 生成prefab的同時也會自動生成AnimatorController，可以使用

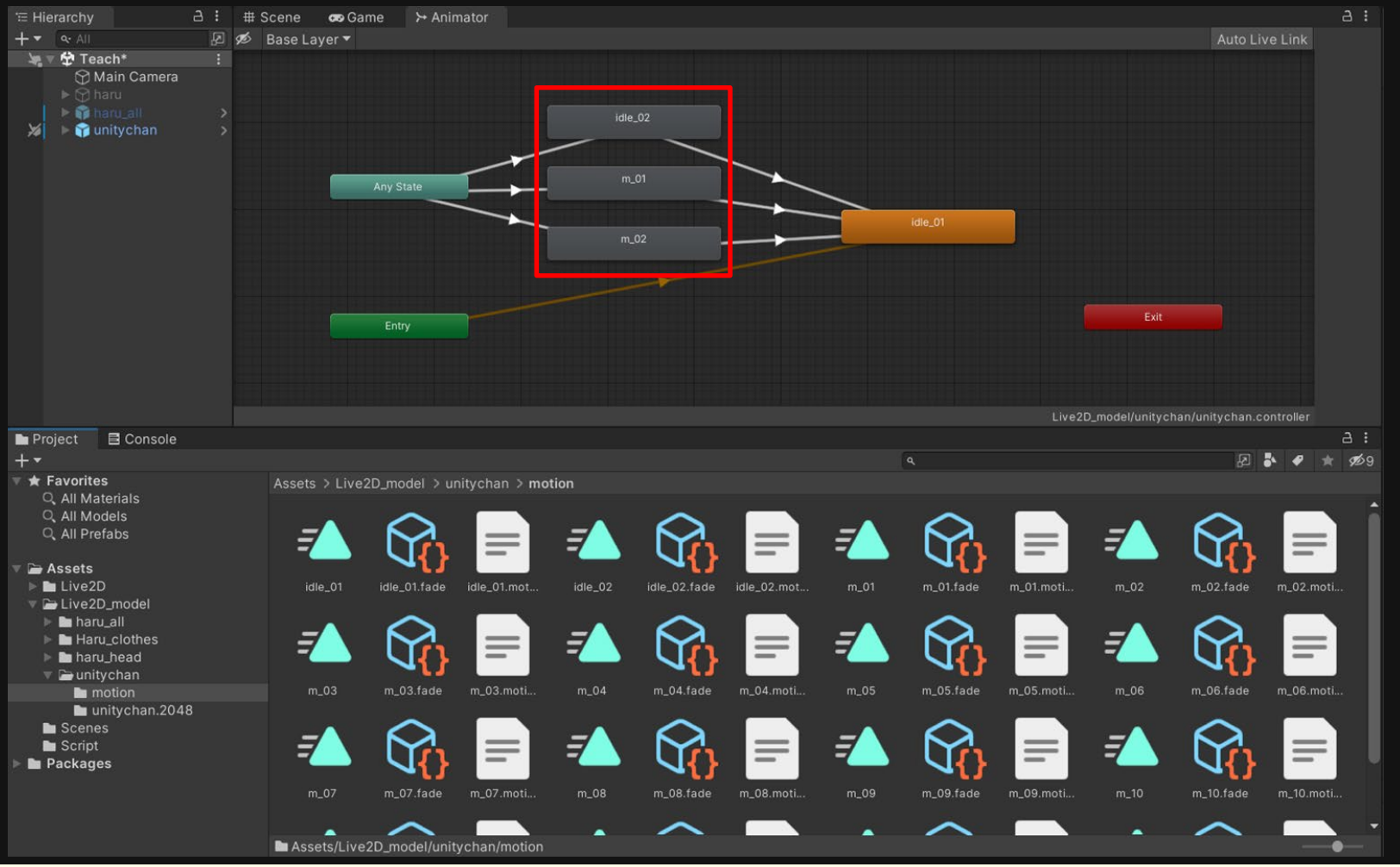


添加Animator Controller

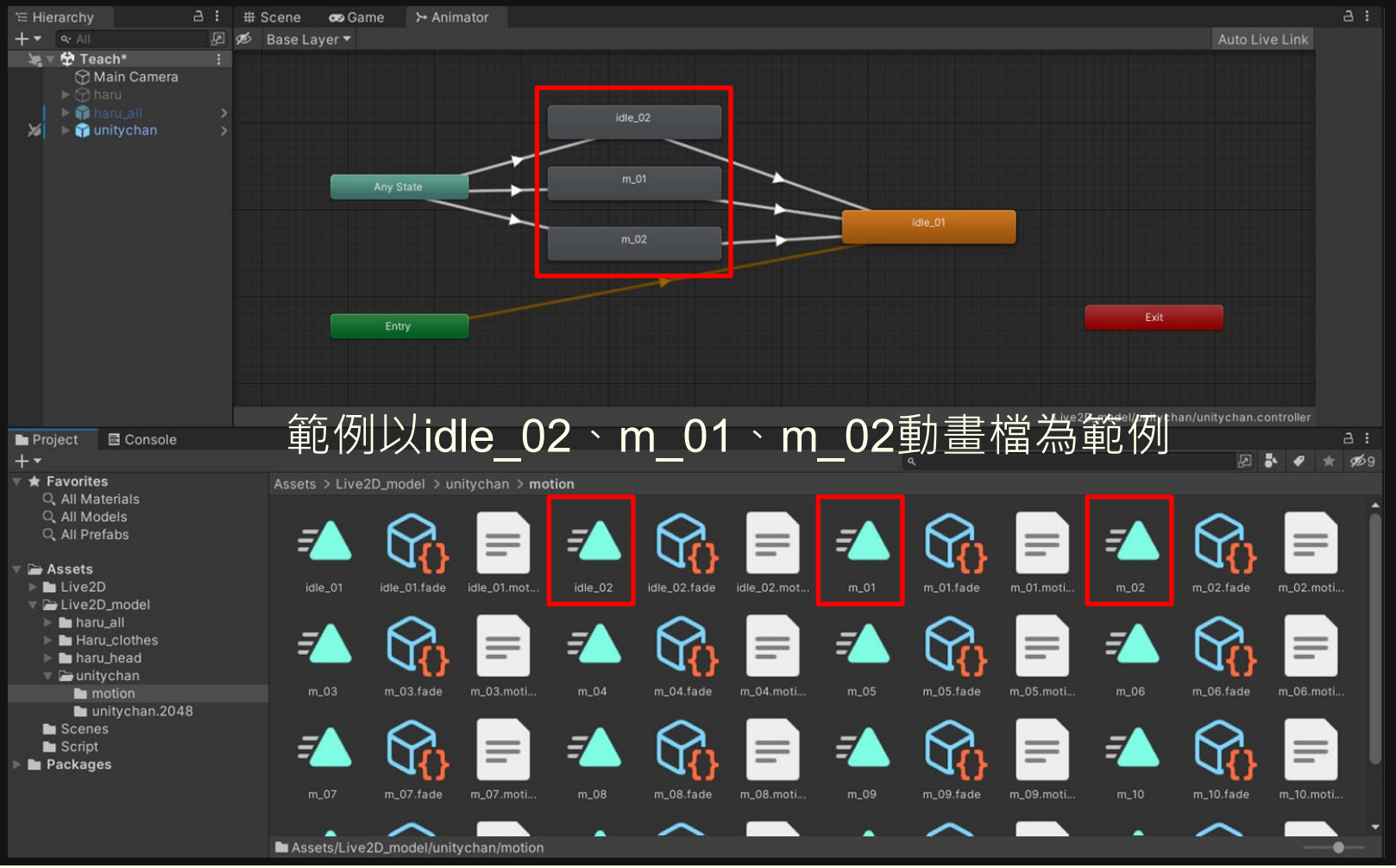
添加Animation



添加Animation

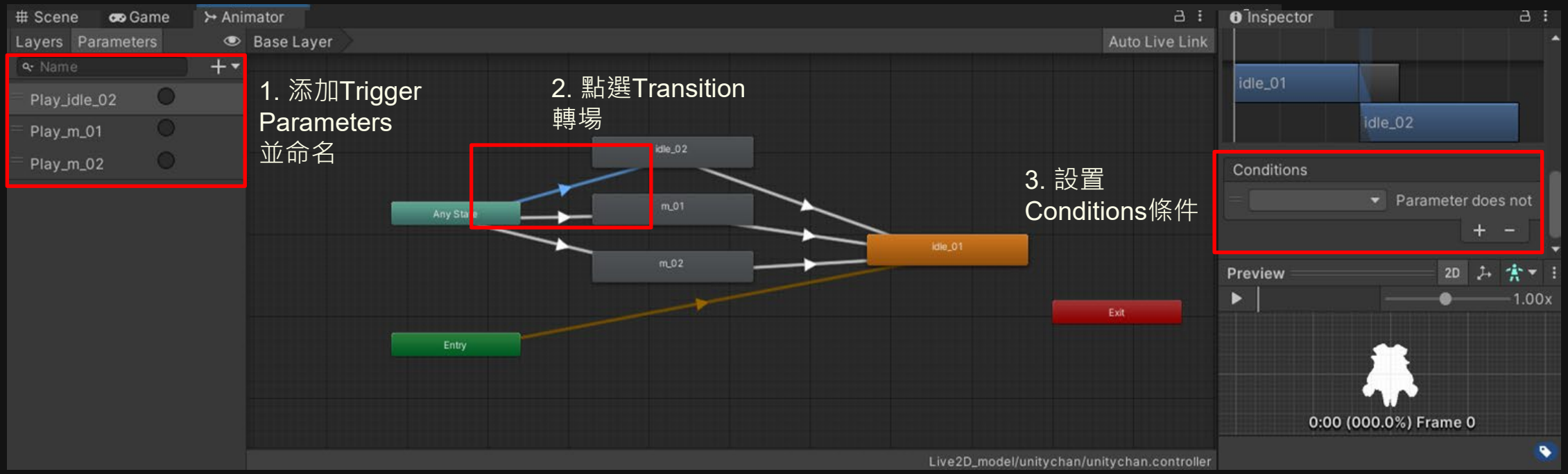


添加Animation



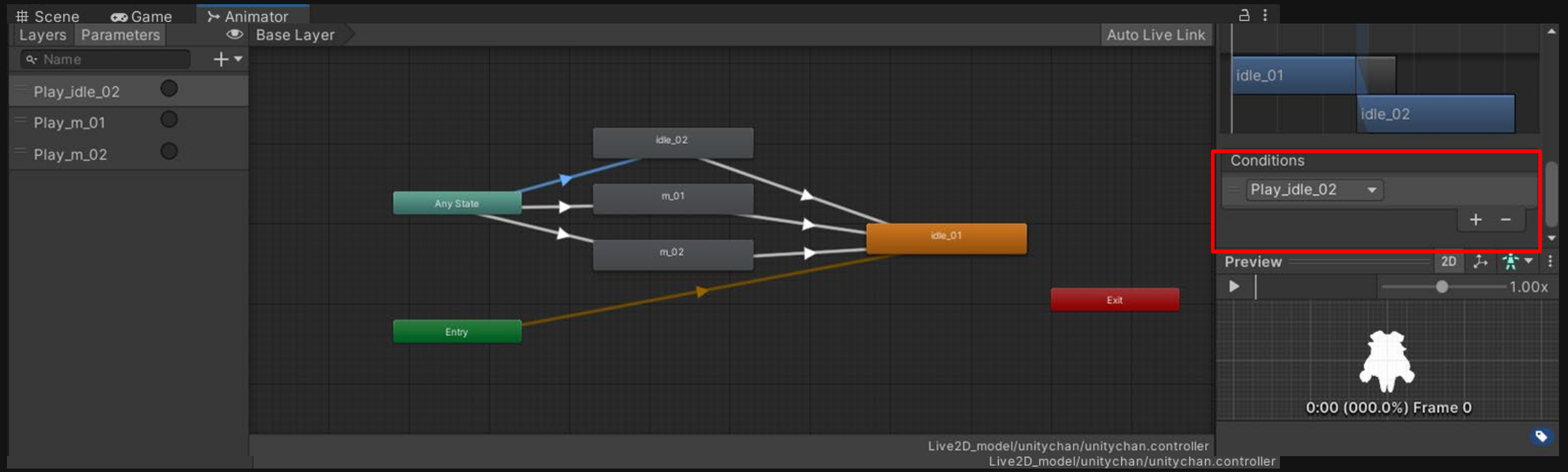
範例以idle_02、m_01、m_02動畫檔為範例

設置動畫切換條件


Scene
Game
Animator
Layers
Parameters
Base Layer
Auto Live Link
Name
Play_idle_02
Play_m_01
Play_m_02
1. 添加Trigger Parameters 並命名
2. 點選Transition 轉場
idle_02
m_01
m_02
Any State
idle_01
Entry
Exit
3. 設置 Conditions條件
Inspector
idle_01
idle_02
Conditions
Parameter does not
Preview
2D
1.00x
0:00 (000.0%) Frame 0
Live2D_model/unitychan/unitychan.controller

設置動畫切換條件-範例

※ 其他Transition同理



開始寫程式

教學檔案

步驟1. 讓腳本(Script)取得到Component (Animator)

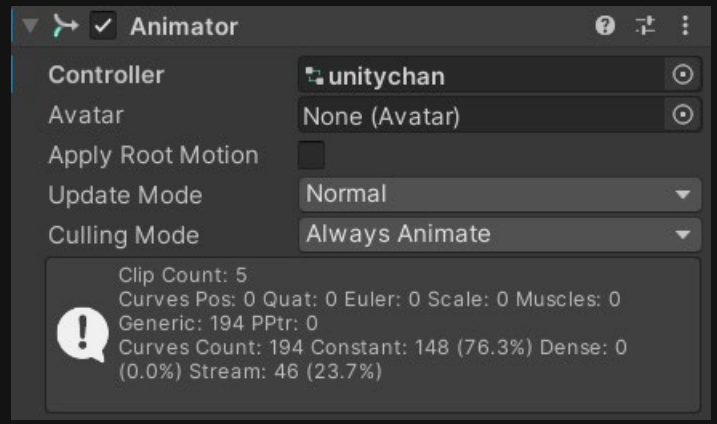
記得新建一個Script並掛載到GameObject
上

```
// define container
6 references
Animator UnityChan_Live2D_Animator;
```

放到已定義好的容器

取得
Component

```
void Start()
{
    this.UnityChan_Live2D_Animator = this.gameObject.GetComponent<Animator>();
}
```



完整程式碼

```
// define container
6 references
Animator UnityChan_Live2D_Animator;

// Start is called before the first frame update
0 references
void Start()
{
    this.UnityChan_Live2D_Animator = this.gameObject.GetComponent<Animator>();
}
```

完整程式碼

```
// define container
6 references
Animator UnityChan_Live2D_Animator;

// Start is called before the first frame update
0 references
void Start()
{
    this.UnityChan_Live2D_Animator = this.gameObject.GetComponent<Animator>();
}
```

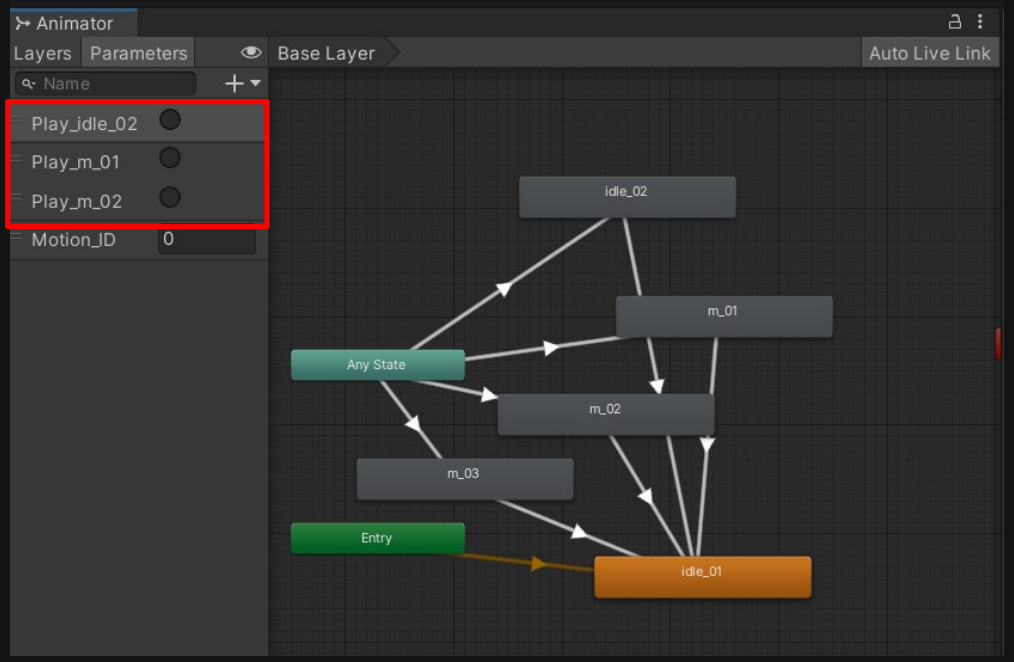
步驟2. 使用Animator中的SetTrigger切換動畫

這是函式的說明，可以看到函式需要一個型態String的參數(Argument)

```
// mouse
// 0 for left button, 1 for
if(Input.GetMouseButton((int
    this.UnityChan_Animator.SetTrigger("Play_idle_02");
}
```

```
void Animator.SetTrigger(string name) (+ 1 多載)
Sets the value of the given trigger parameter.
```

而函式輸入的參數(name)為剛定義的Parameters

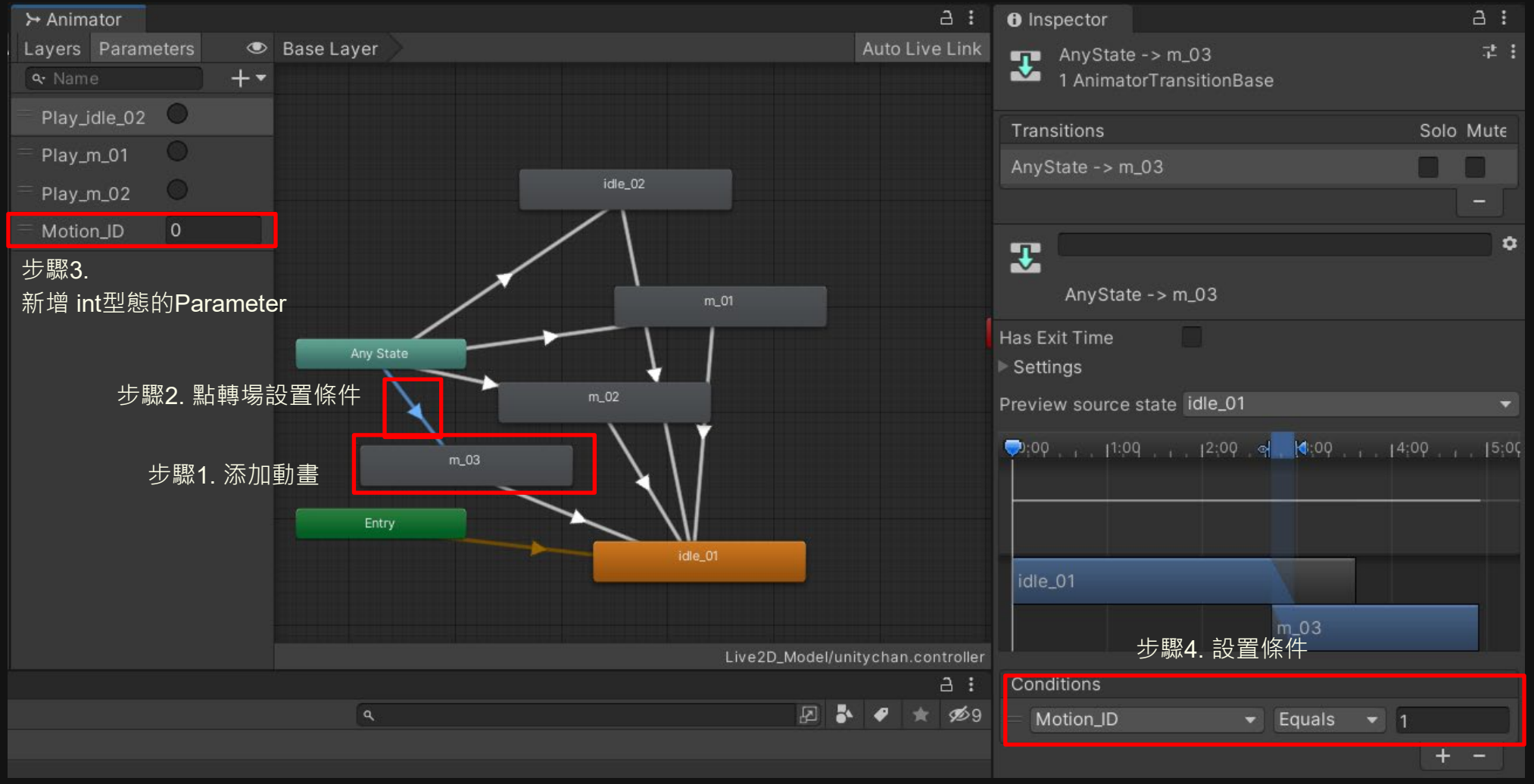


步驟2. 使用Animator中的SetTrigger切換動畫

這裡示範用滑鼠按鍵觸發切換動畫

```
// I set three situation to change animation
// mouse
// 0 for left button, 1 for right button, 2 for the middle button.
if(Input.GetMouseButton((int)MyMouseCode.Left)){
    this.UnityChan_Animator.SetTrigger("Play_idle_02");
}
if(Input.GetMouseButton((int)MyMouseCode.Right)){
    this.UnityChan_Animator.SetTrigger("Play_m_01");
}
if(Input.GetMouseButton((int)MyMouseCode.Middle)){
    this.UnityChan_Animator.SetTrigger("Play_m_02");
}
```

補充: 使用int作為Parameters作為動畫切換的條件



補充:使用Animator中的SetInteger切換動畫

這裡示範用鍵盤空白鍵觸發切換動畫

```
}

if(Input.GetKey(KeyCode.Spac
    this.UnityChan_Animator.SetInteger("Motion_ID",1);
```

```
void Animator.SetInteger(string name, int value) (+ 1 多載)
Sets the value of the given integer parameter.
```

聰明的你或許想到，在Controller中我們是在Any State去做條件偵測轉場，這會導致一個問題: 重複達成條件觸發
那麼，你也許會想說，「那我在下一行再設置Parameters為0就好了吧」

```
if(Input.GetKey(KeyCode.Space)){
    this.UnityChan_Animator.SetInteger("Motion_ID",1);

    // return 0 to avoid trigger again
    this.UnityChan_Animator.SetInteger("Motion_ID",0);
```

但是是錯的，因為Parameter設定太快，導致Unity來不及偵測就已經改回來了
※ 牽扯到Unity的執行緒，因此讀者先記解法

補充:使用SetInteger切換動畫的解法

```
if(Input.GetKey(KeyCode.Space)){
    this.UnityChan_Animator.SetInteger("Motion_ID",1);

    StartCoroutine(DelayLastFrame());
}

IEnumerator DelayLastFrame(){
    yield return new WaitForEndOfFrame();
    this.UnityChan_Animator.SetInteger("Motion_ID",0);
}
```

補充:使用SetInteger切換動畫的解法

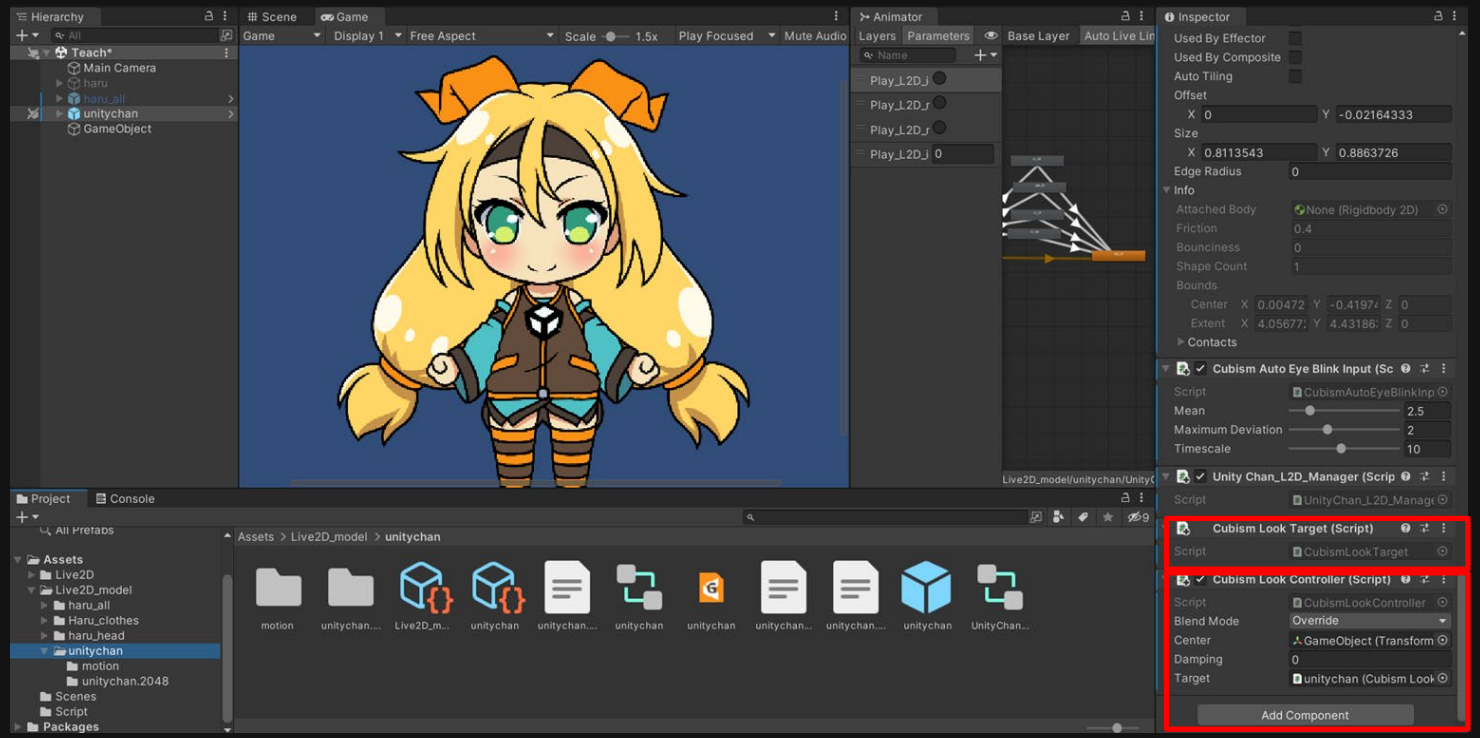
```
if(Input.GetKey(KeyCode.Space)){
    this.UnityChan_Animator.SetInteger("Motion_ID",1);

    StartCoroutine(DelayLastFrame());
}

IEnumerator DelayLastFrame(){
    yield return new WaitForEndOfFrame();
    this.UnityChan_Animator.SetInteger("Motion_ID",0);
}
```

簡單說明：
StartCoroutine開啟協程(Coroutine)，
WaitForEndOfFrame()等待到幀結尾，
會先yield return，
等到幀結尾後，
會返回繼續往下執行SetInteger(“…”,0)

這關於到
Coroutine(協程)、
Ienumerator(列舉器) ※ 集合、
Yield return(反覆運算子) ※ 迭代
有興趣的讀者自己研究

補充: 讓人物看向滑鼠方向

給Prefab gameobject腳本



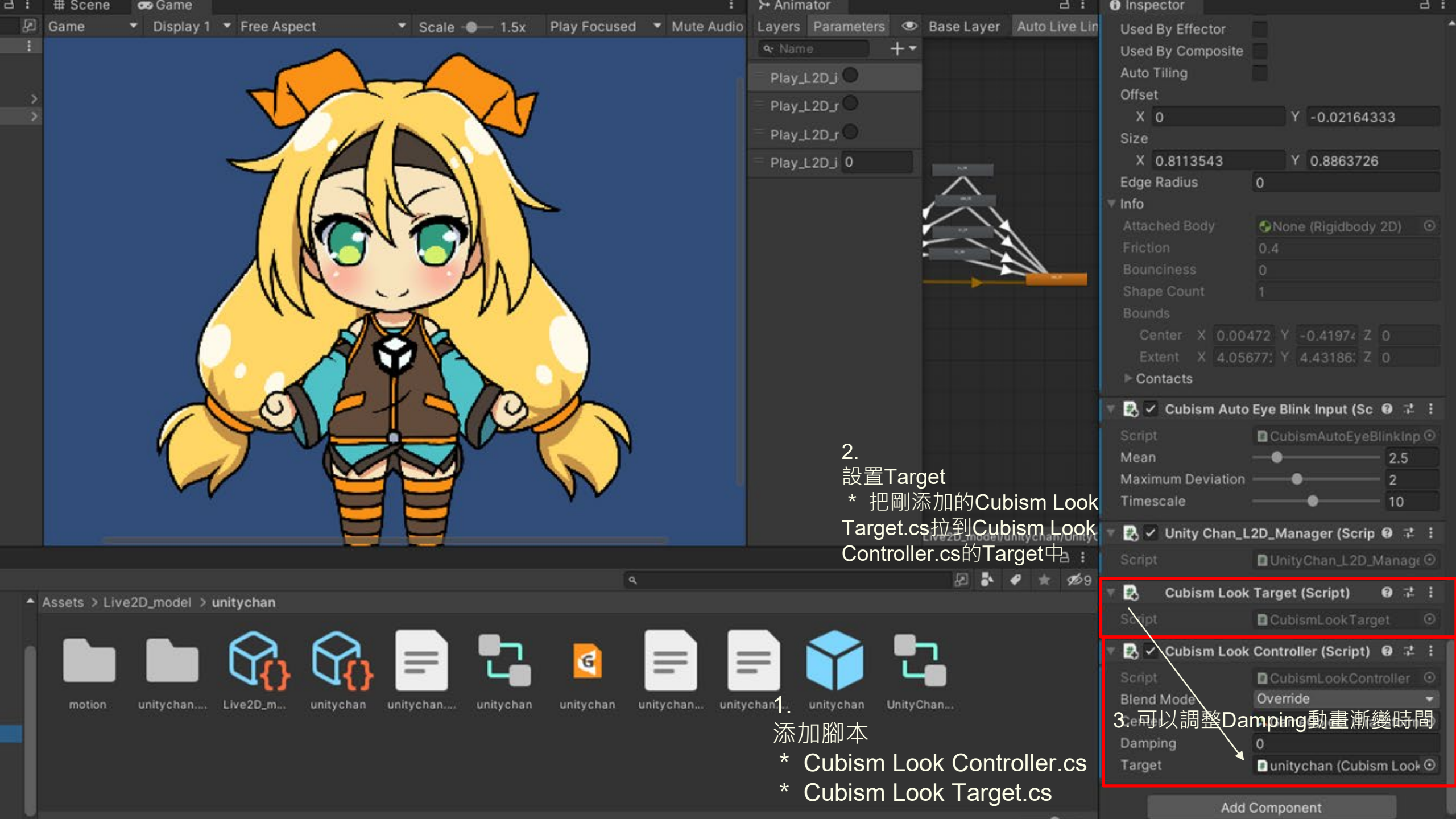
添加腳本

* Cubism Look Controller.cs
* Cubism Look Target.cs

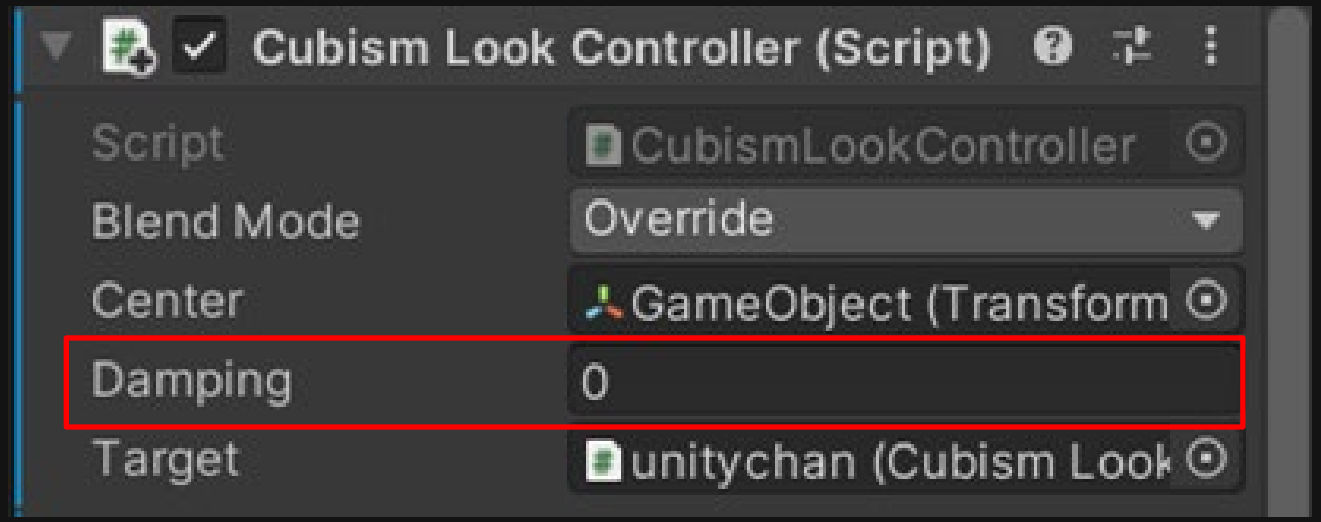
設置Target

* 把剛添加的Cubism Look Target.cs拉到Cubism Look Controller.cs的Target中

可以調整Damping動畫漸變時間

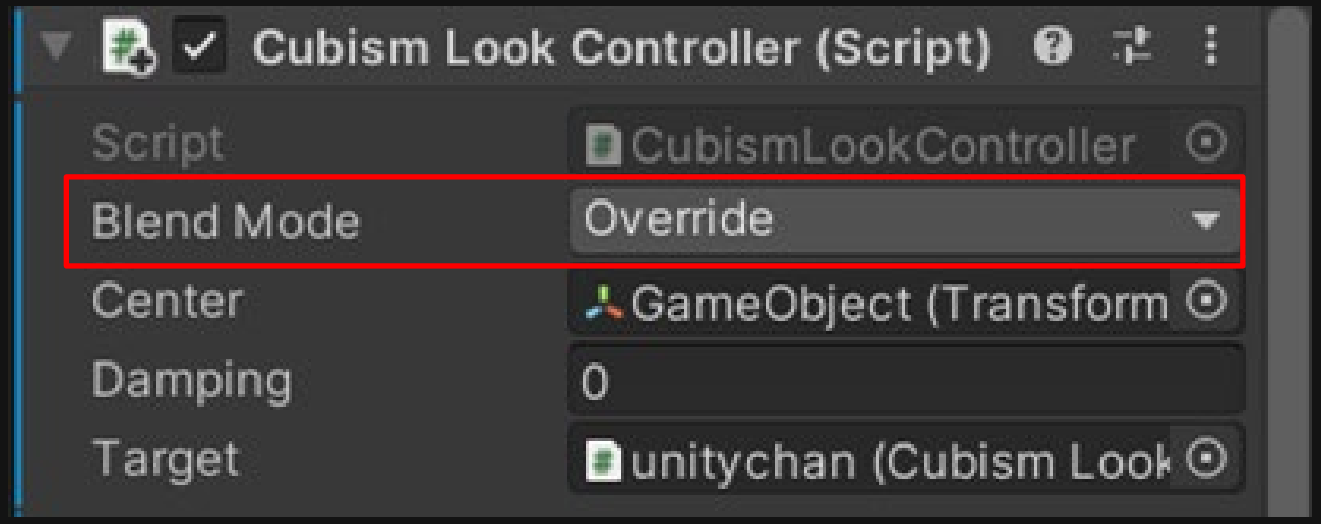

Scene
Game
Game
Display 1
Free Aspect
Scale
1.5x
Play Focused
Mute Audio
Animator
Layers
Parameters
Base Layer
Auto Live Lin
Name
Play_L2D_j
Play_L2D_r
Play_L2D_r
Play_L2D_j 0
Inspector
Used By Effector
Used By Composite
Auto Tiling
Offset
X 0
Y -0.02164333
Size
X 0.8113543
Y 0.8863726
Edge Radius 0
Info
Attached Body None (Rigidbody 2D)
Friction 0.4
Bounciness 0
Shape Count 1
Bounds
Center X 0.00472 Y -0.4197 Z 0
Extent X 4.05677 Y 4.43186 Z 0
Contacts
Cubism Auto Eye Blink Input (Sc
Script CubismAutoEyeBlinkInp
Mean 2.5
Maximum Deviation 2
Timescale 10
Unity Chan_L2D_Manager (Scrip
Script UnityChan_L2D_Manag
Cubism Look Target (Script)
Script CubismLookTarget
Cubism Look Controller (Script)
Script CubismLookController
Blend Mode Override
Damping 0
Target unitychan (Cubism Look
Add Component
Assets > Live2D_model > unitychan
motion
unitychan...
Live2D_m...
unitychan
unitychan...
unitychan
unitychan
unitychan...
unitychan...
unitychan
UnityChan...
2.
設置Target
* 把剛添加的Cubism Look Target.cs拉到Cubism Look Controller.cs的Target中
1.
添加腳本
* Cubism Look Controller.cs
* Cubism Look Target.cs
3. 可以調整Damping動畫漸變時間

漸變時間



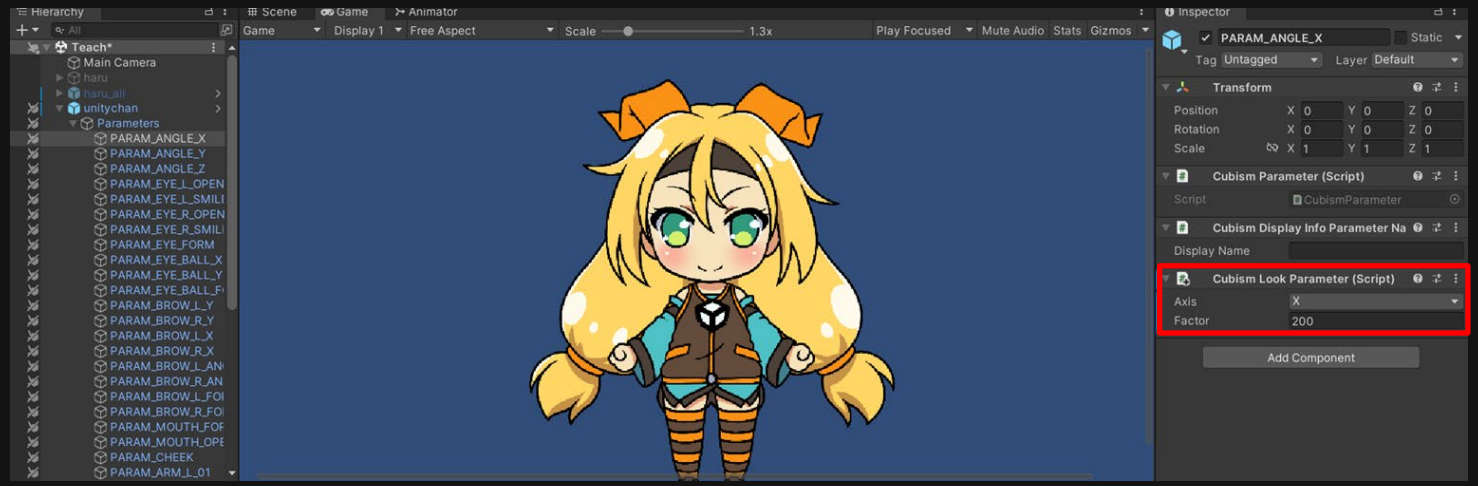
為看向滑鼠（動畫）的速度，預設是0，
數字越大，看向滑鼠的速度就會變慢，
1就近乎當機狀態
越小看向滑鼠的速度就會變快
讀者可以嘗試看看-1

混合模式(Blend Mode)

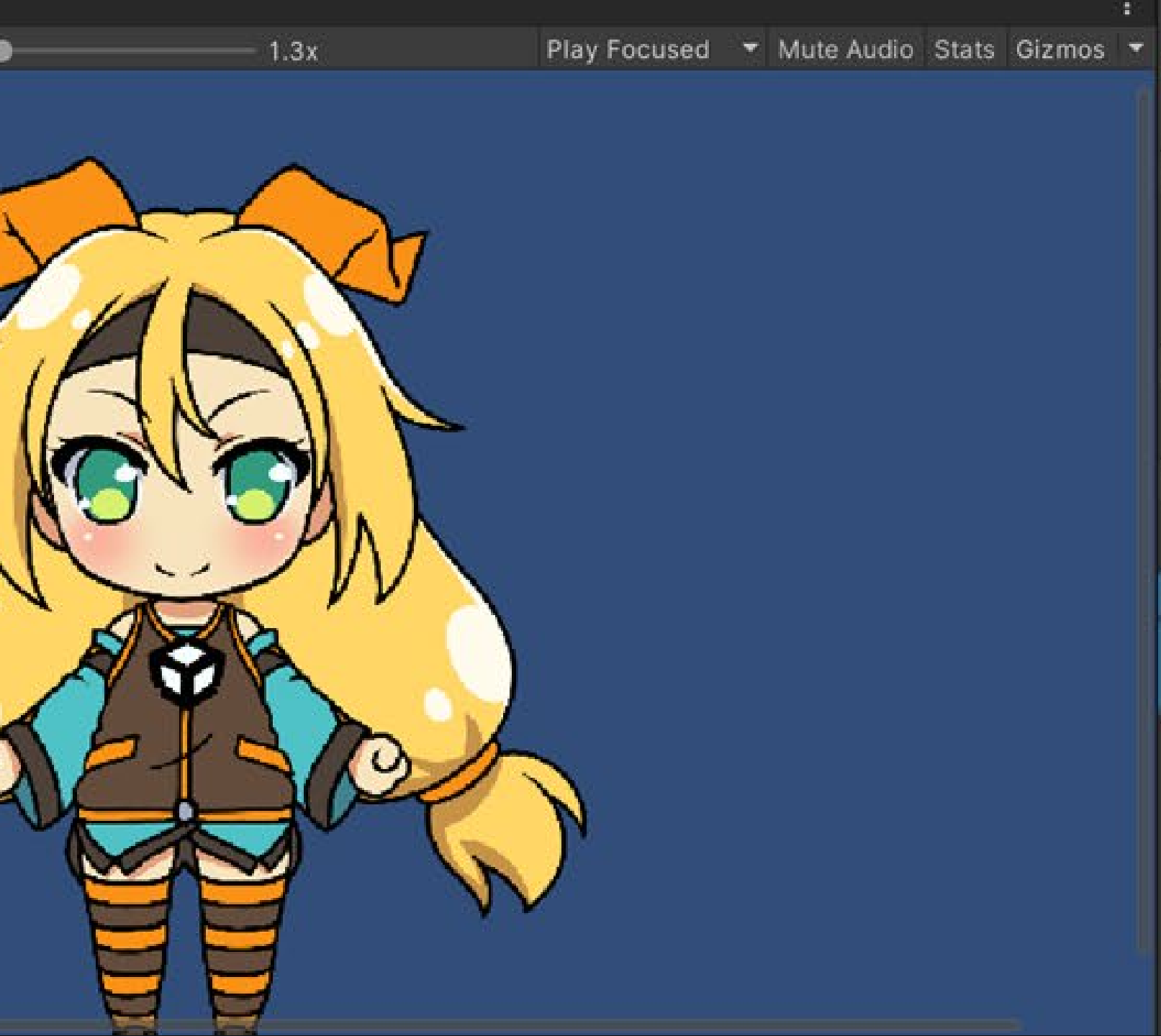


預設是Additive(相加)
建議使用Override(覆蓋)
效果會比較明顯

給部件腳本



添加腳本
＊ Cubism Look Parameter.cs
可以調整Factor倍率



Inspector

PARAM_ANGLE_X

Static

Tag Untagged

Layer Default

Transform

Position	X	0	Y	0	Z	0
Rotation	X	0	Y	0	Z	0
Scale	X	1	Y	1	Z	1

Cubism Parameter (Script)

Script CubismParameter

Cubism Display Info Parameter Na

Display Name

Cubism Look Parameter (Script)

Axis X

Factor 200

Add Component

Factor倍率影響參數變化，
越大動畫效果越明顯
建議50~200區間