
Scriptable Object
入門と活用例
unity
Made with Unity - Generative Art

unity

Made with Unity

Unityと
オブジェクト


```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class CallSO : MonoBehaviour
{
    public BaseSO so{ get; set; }

    public void Call (Text label)
    {
        so.Push (label);
    }
}
```

●シーンにGameObjectを配置する

●GameObjectにComponentを追加する

音を出す

移動する

ゲーム進行管理

音を出す

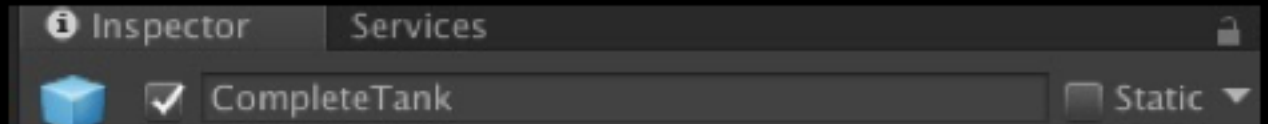
移動する

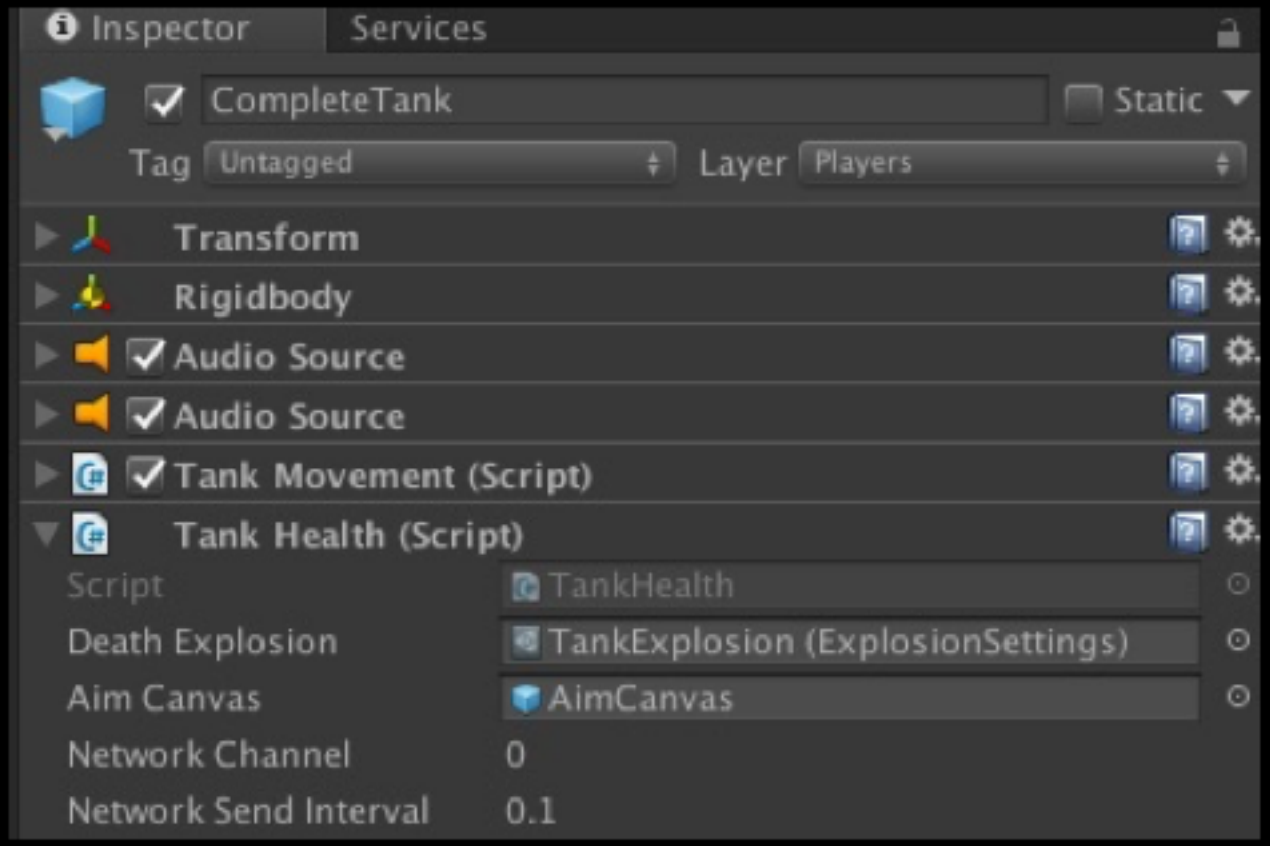
他のオブジェクト
と接触

エンジンに
モデルの
描画を依頼

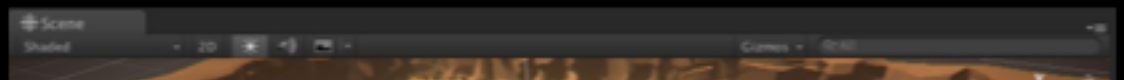
物理演算

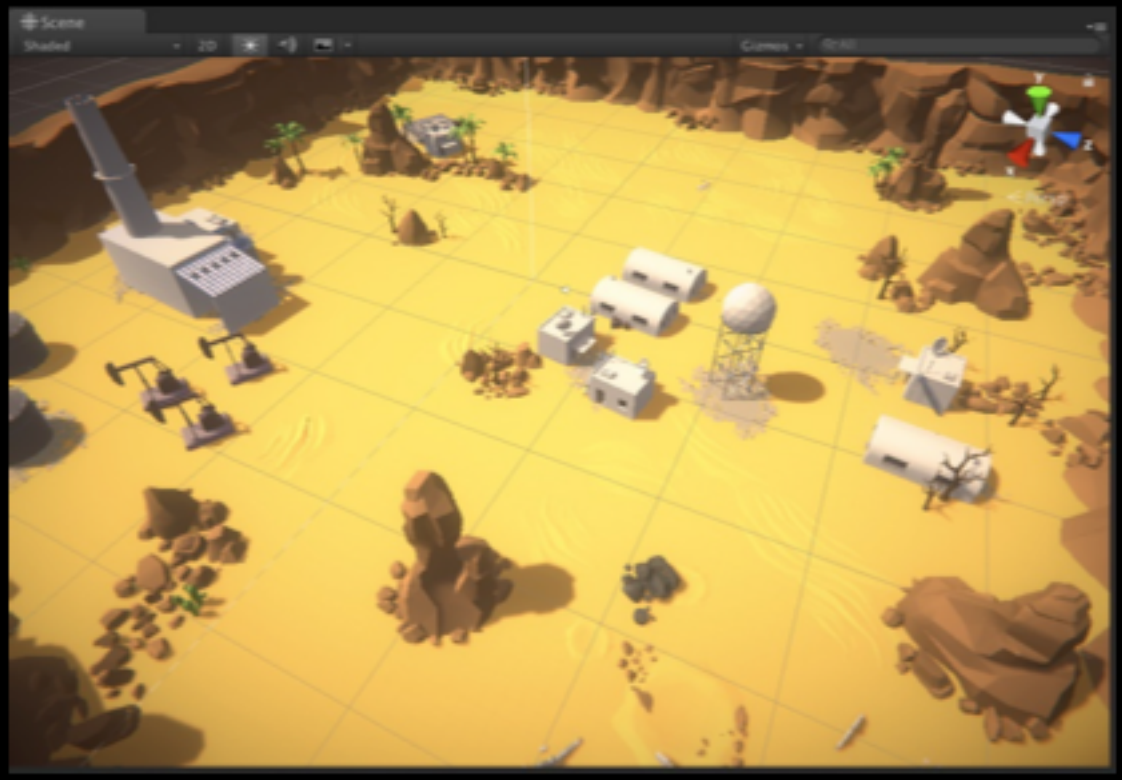
- コンポーネントが、オブジェクトの振る舞いを決める
- オブジェクト同士が干渉してゲームを作る





●GameObjectに追加したコンポーネントの
フィールドや参照関係を埋めて、振る舞いを調整

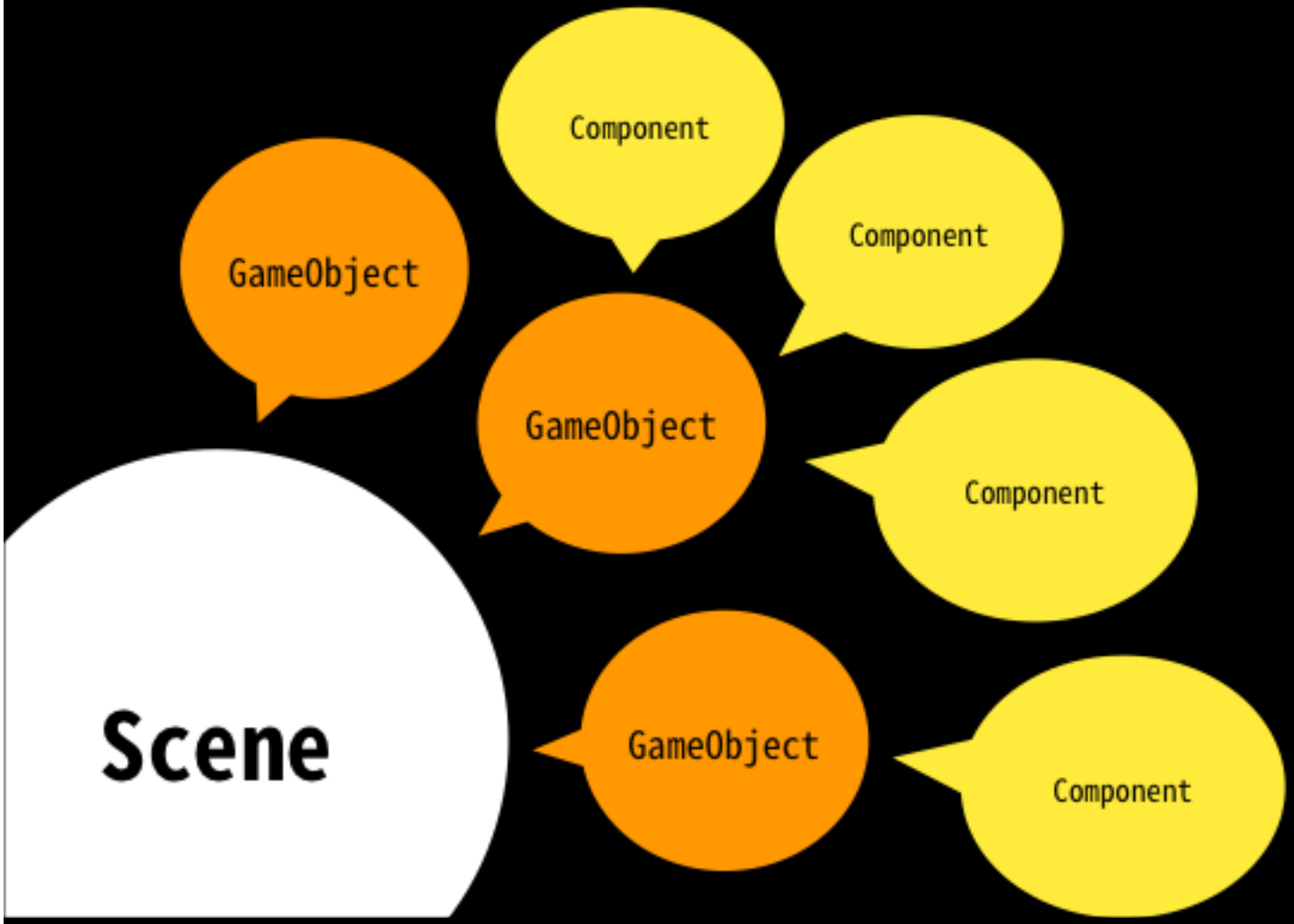


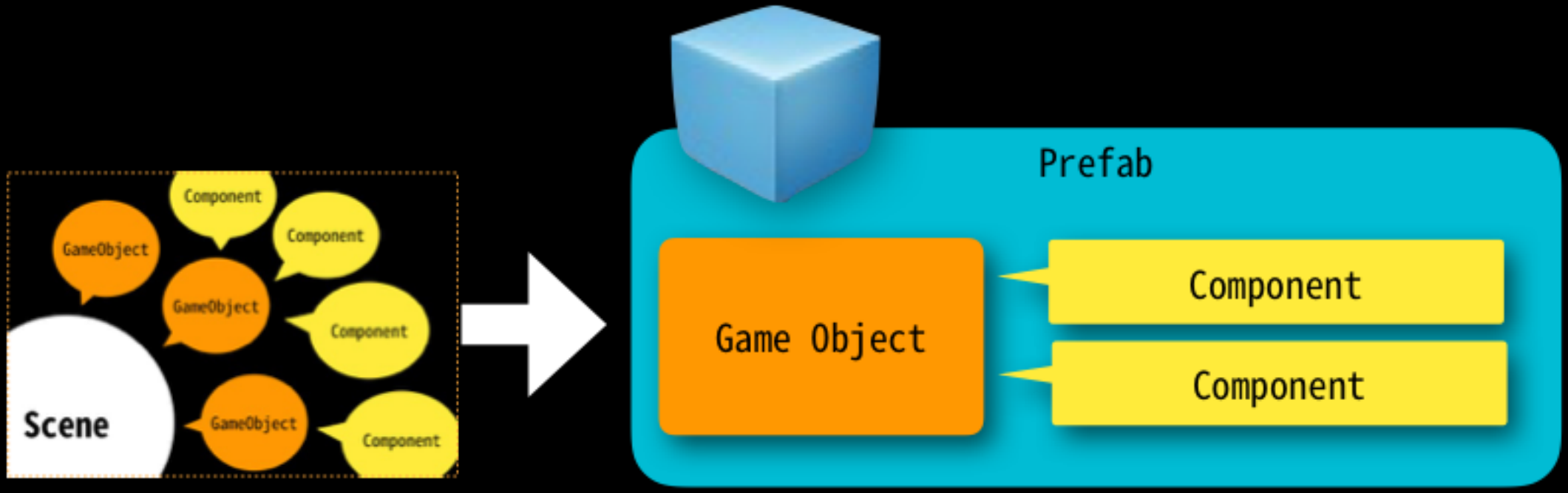


- （比較的静的な）世界が出来る

Scene / Game Object / Component の関係

Scene / Game Object / Component の関係





●PrefabでGame Objectをファイル（アセット）として書き出す

●Prefabをシーンに配置したり、動的にPrefabをInstantiate（複製）してゲームを表現するのが、Unityの最も一般的な使い方

Prefab (GameObject & Monobehaviour) では面倒なケース

Prefab (GameObject & Monobehaviour) では面倒なケース

- 使用する際にInstantiateする
- 再生停止時に変更した情報が失われる
- GameObjectがいつも一緒に生成される
- インスタンス化したPrefab間でデータを共有しない
- 沢山のフィールドを持つとInstantiateのコストが上がる

Scriptable Object

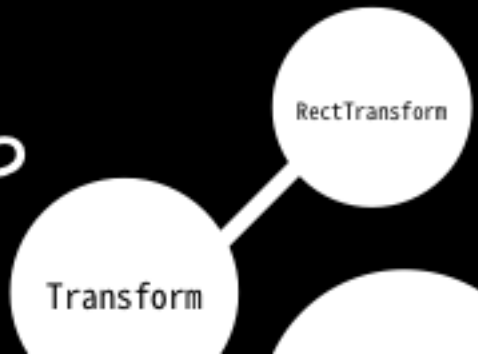
Scriptable Objectとは？

●UnityのObjectから派生したクラス

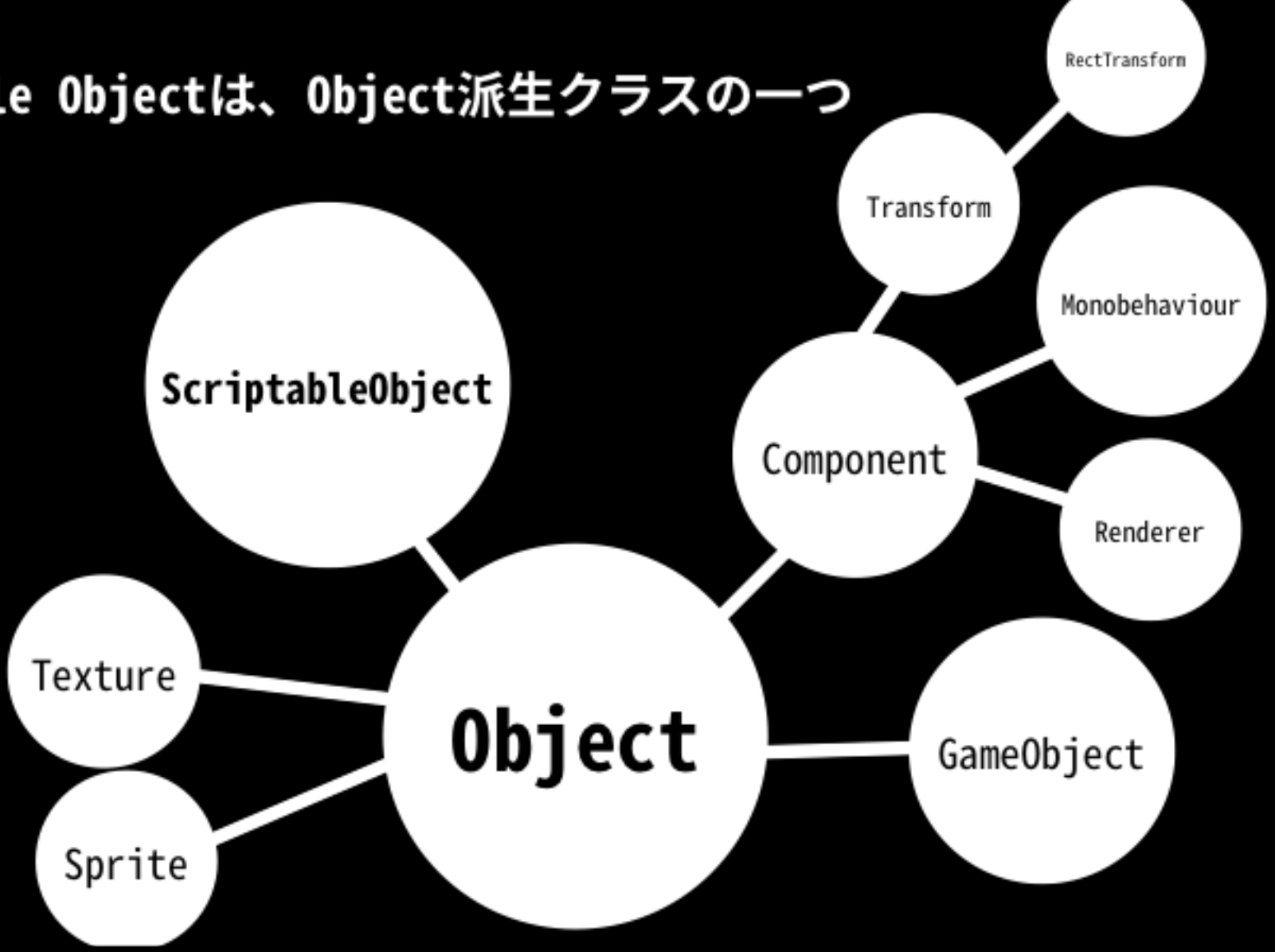
Scriptable Objectとは？

- UnityのObjectから派生したクラス
- アセットとして保存（シリアライズ）できる
- Objectへの参照やフィールドの情報を保持したまま、シリアライズできる
- 保存したデータはInspectorから簡単に確認できる

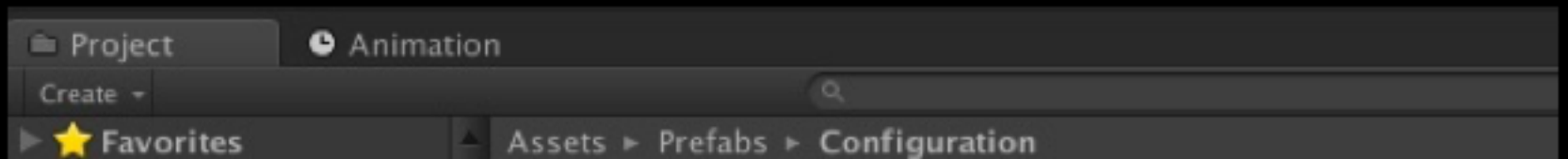
Scriptable Objectは、Object派生クラスの一つ



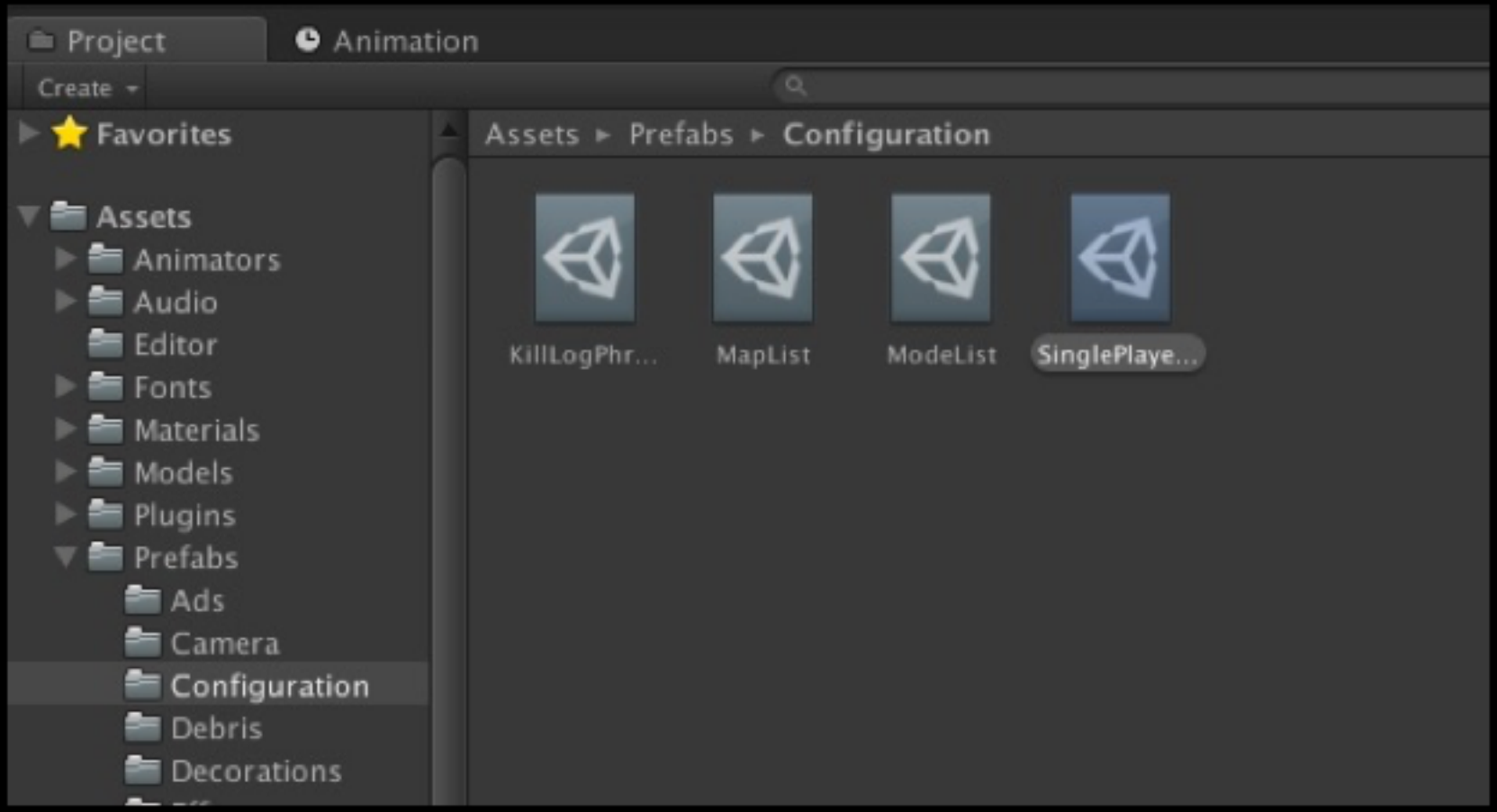
Scriptable Objectは、Object派生クラスの一つ



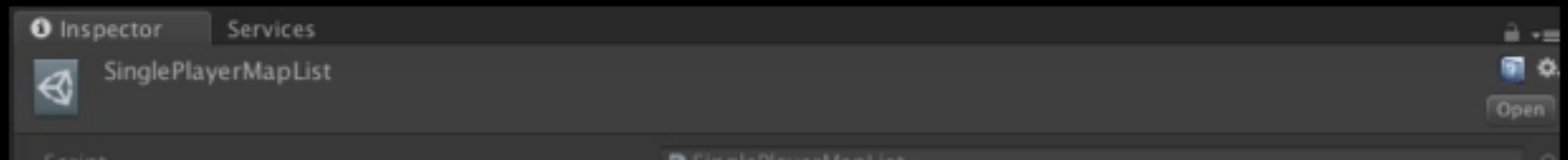
アセットとしてプロジェクトに格納・使用できる



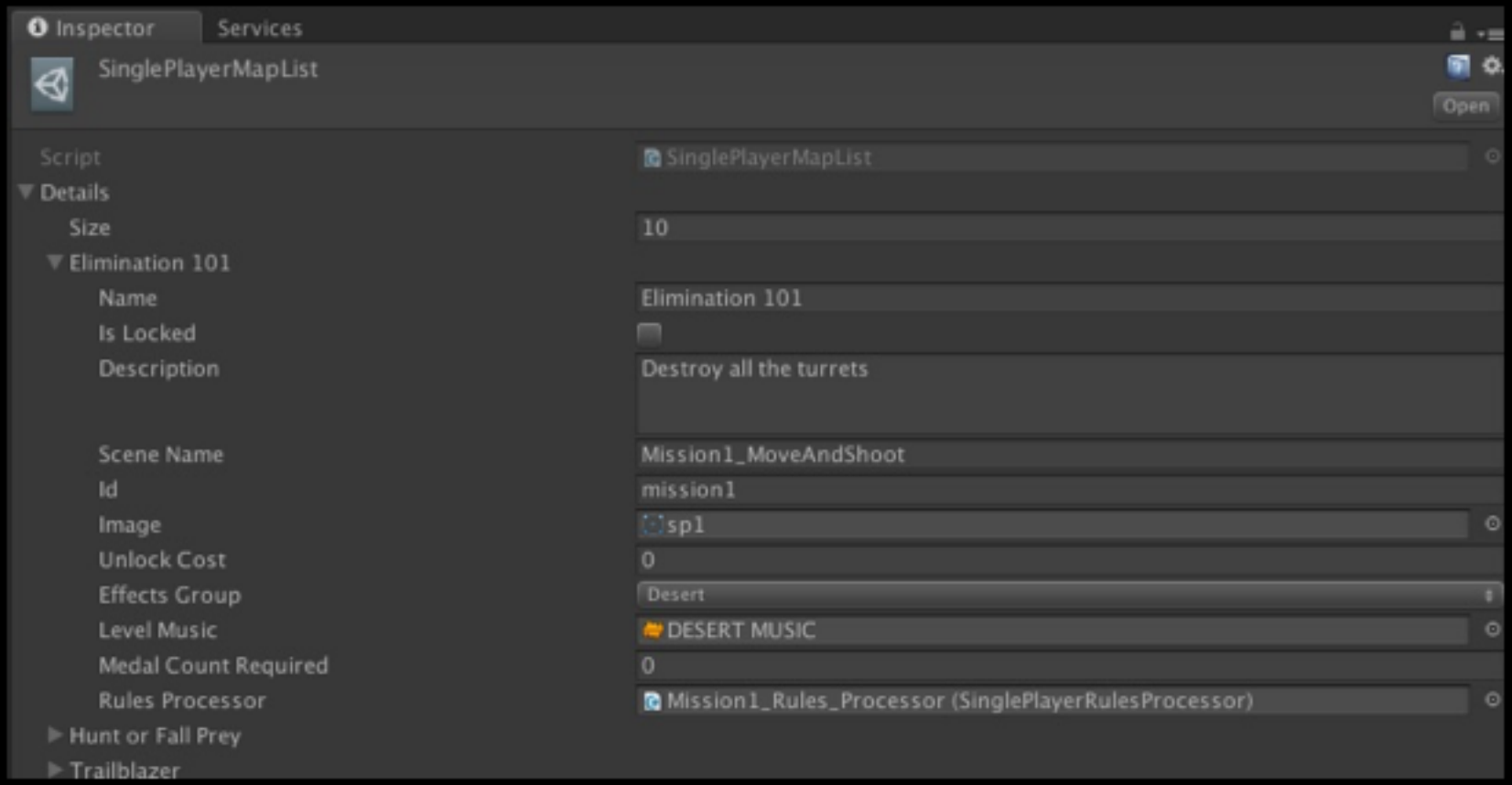
アセットとしてプロジェクトに格納・使用できる



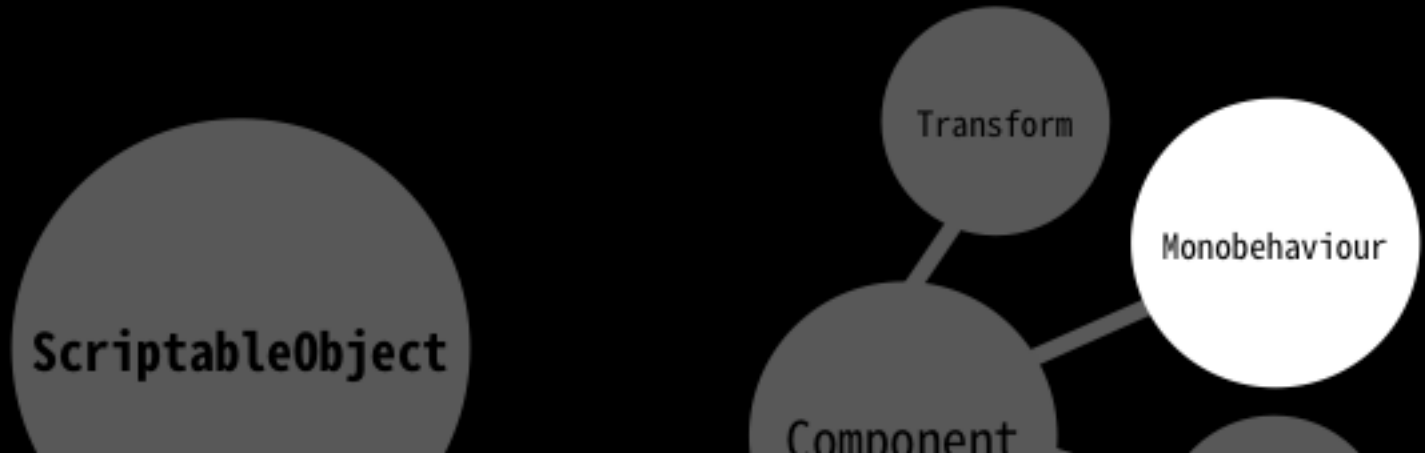
Objectへの参照を保持したままシリアライズ

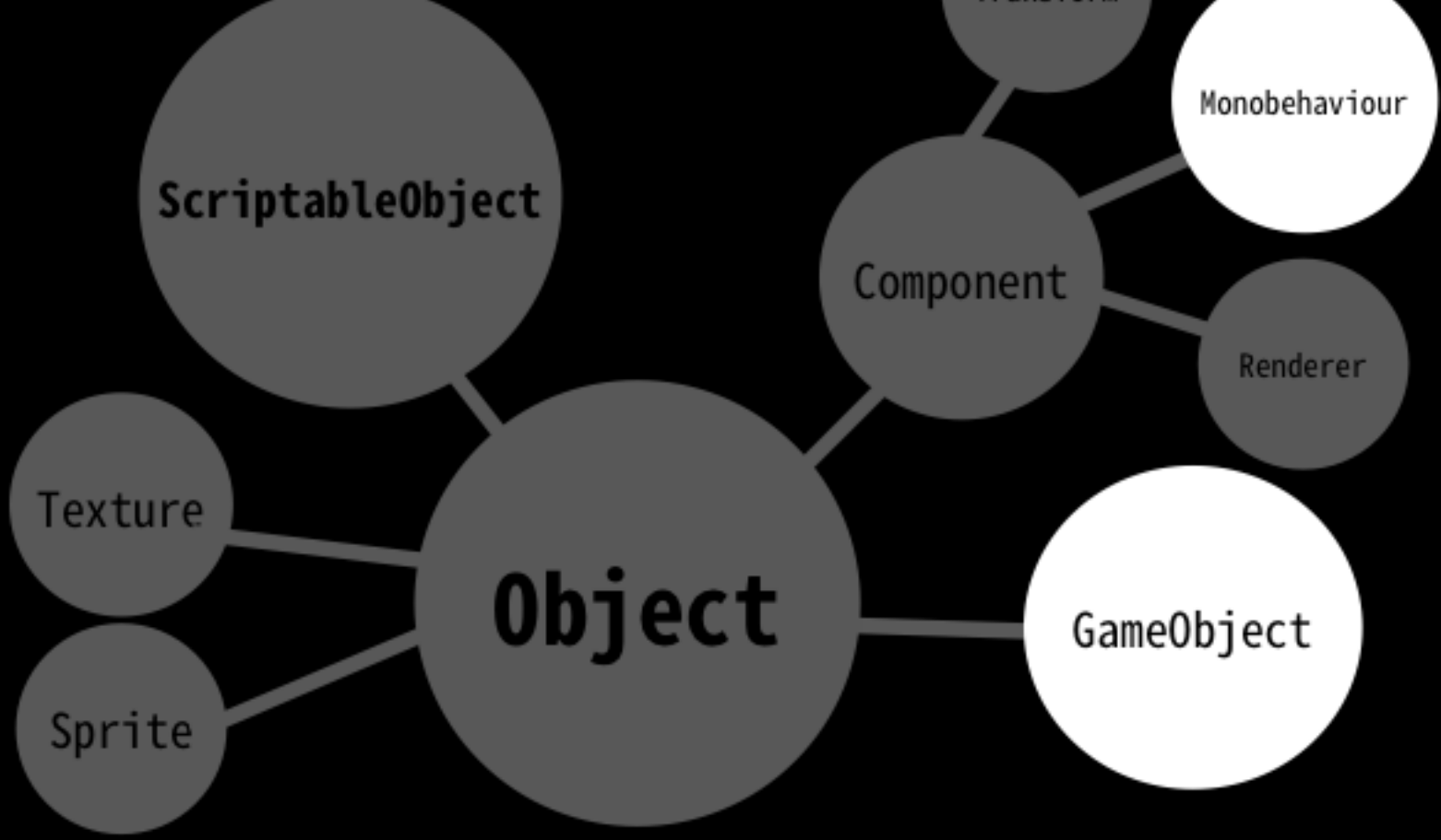


Objectへの参照を保持したままシリアライズ



なるほど…
Prefabの下位互換か…





●ScriptableObjectは
Prefab（GameObject & Monobehaviour）とは違うもの

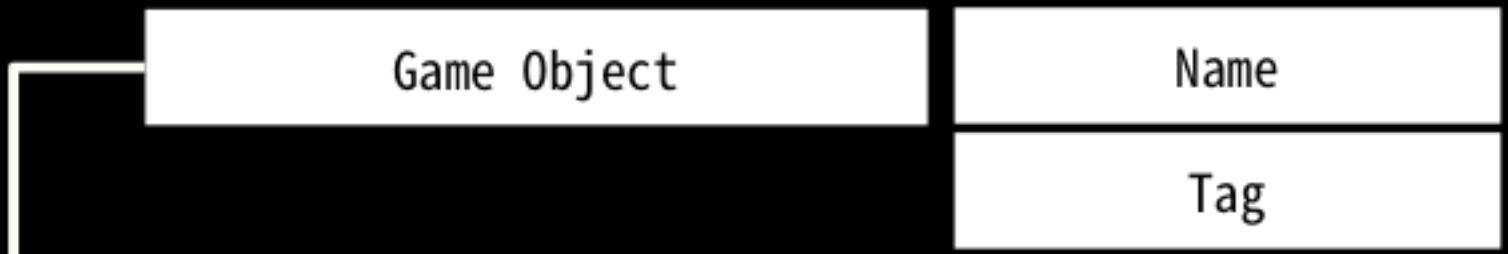
Monobehaviourを見直す

●Monobehaviourはスクリプト

Monobehaviourを見直す

- Monobehaviourはスクリプト
- 様々なコールバックをエンジン（Unity）から受け取れる
- GameObjectに追加してはじめて動作する
- SceneもしくはPrefabに保存（シリアライズ）できる
- 保存したデータはInspectorから簡単に確認できる
- ゲーム再生終了時にリセットされる

PrefabにはGameObjectとTransformが追加される



PrefabにはGameObjectとTransformが追加される

Game Object

Name

Tag

Layer

Transform

Vector3

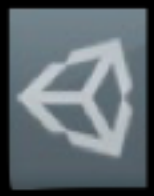
Quaternion

Component

HP

TYPE

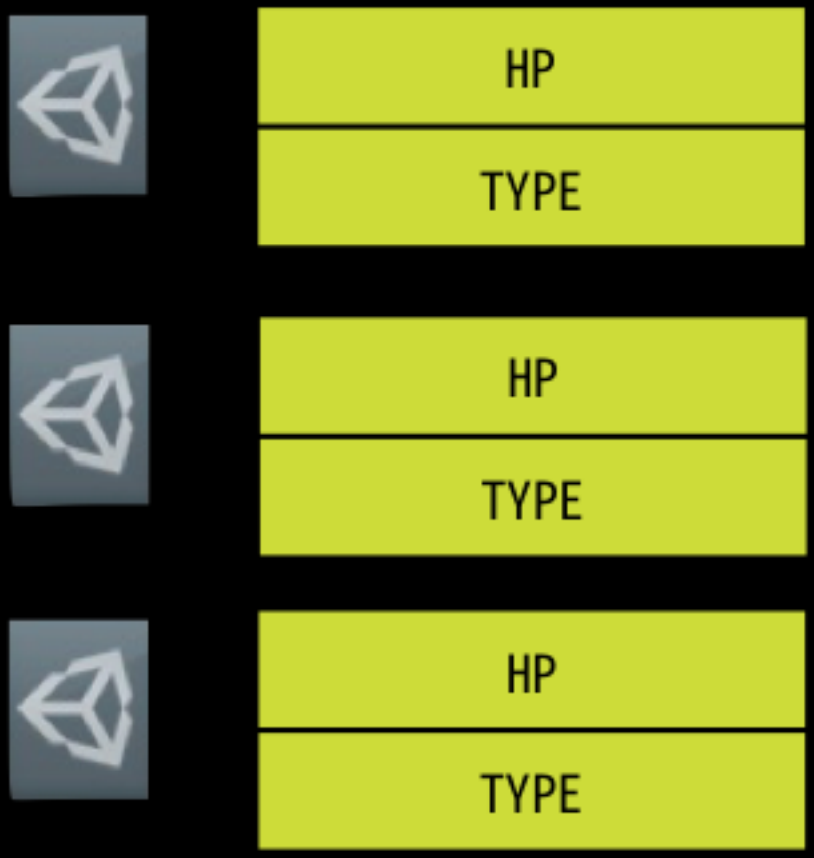
Scriptable Objectには余計なオブジェクトはつかない



HP

TYPE

ScriptableObjectには派生なオブジェクトはつかない



コールバックの量が違う

Monobehaviour　　Scriptable Object

Monobehaviour

64

コールバック

Scriptable Object

4

コールバック

※スクリプトに定義したコールバックのみ呼ばれる

使用方法が違う

Monobehaviour

Instantiateで

Scriptable Object

参照する

Monobehaviour

Instantiateで Sceneに生成

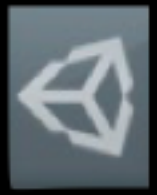
再生終了時に リセット

Scriptable Object

参照する

再生終了時に リセットされない

Scriptable Object



- Scriptable Objectはスクリプト
- 殆どのコールバックを受け取らない

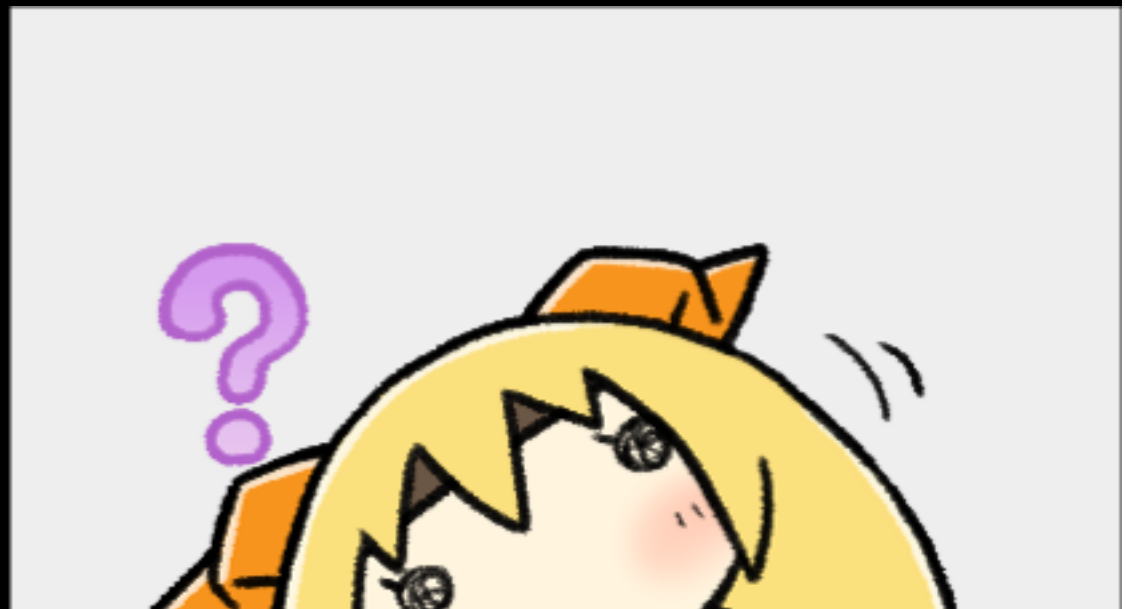
- Scriptable Objectはスクリプト
- 殆どのコールバックを受け取らない
- Game Objectにアタッチする必要はない
- インスタンス毎に異なるアセットとして保存できる
- 保存したデータはInspectorから簡単に確認できる
- ゲーム再生時にリセットされない

Monobehaviour VS ScriptableObject

	Monobehaviour	Scriptable Object
スクリプト？	YES	YES
コールバック	多い	少ない

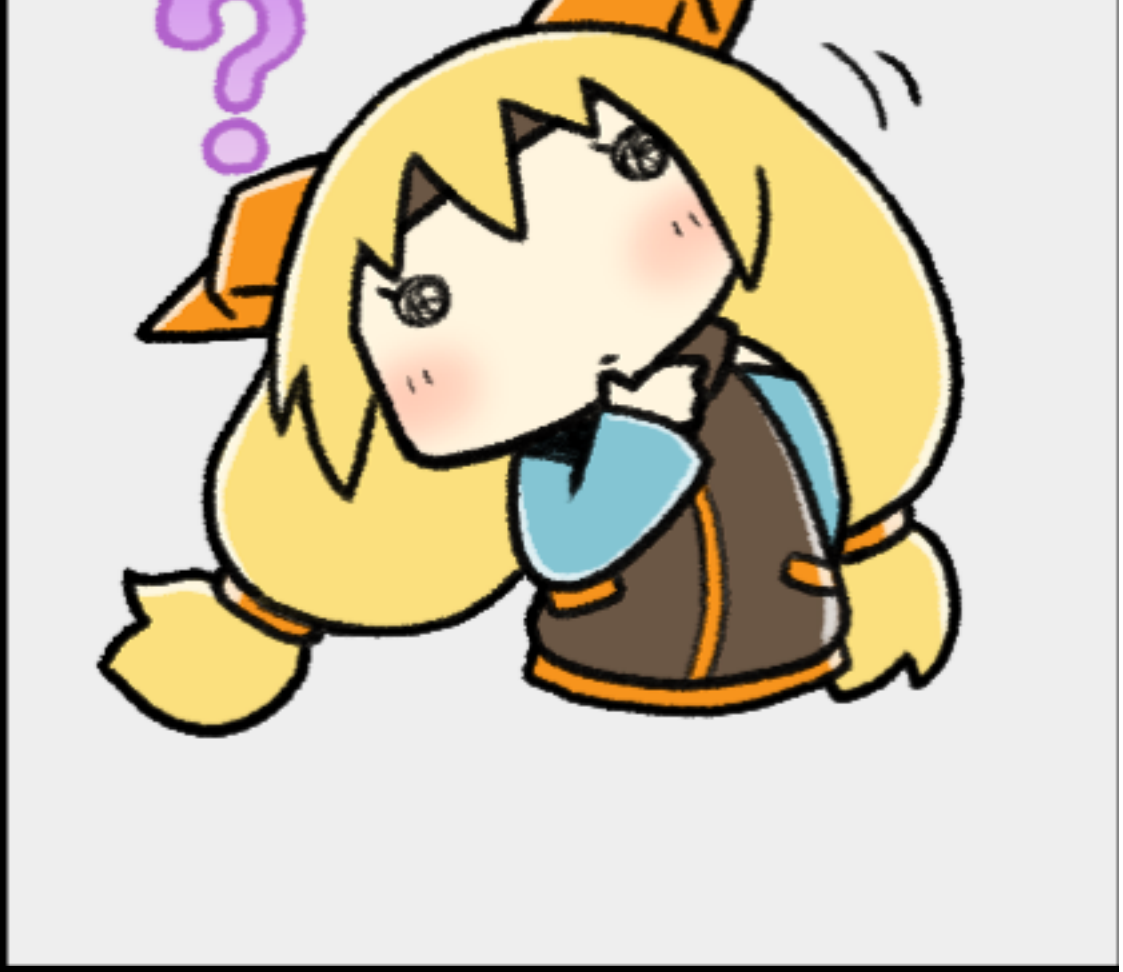
	Monobehaviour	Scriptable Object
スクリプト？	●YES	●YES
コールバック	●多い	●少ない
GameObjectに 追加する必要	●ある	●ない
余計なオブジェクト	●GameObjectとTransformが付く	●インスタンスのみで保存
保持するフィールドの値	●簡単に確認できる	●簡単に確認できる
ゲーム終了時の動作	●リセットされる	●リセットされない

どんな感じで使うの？



どんな感じで使うの？

Scriptable Objectを使ってみる



Scriptable Objectのキーワード

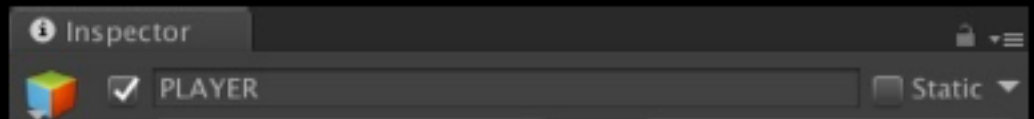
- ScriptableObject
- CreateAssetMenu

●ScriptableObject

●CreateAssetMenu

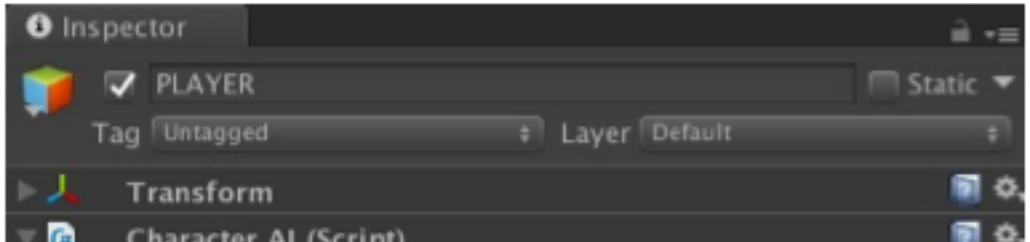
PrefabをScriptableObjectに変更

- キャラクターのPrefabをScriptableObjectに変更してみる
  - 最大HPとATKとDEFを持つ
  - HPの状況によって行う挙動

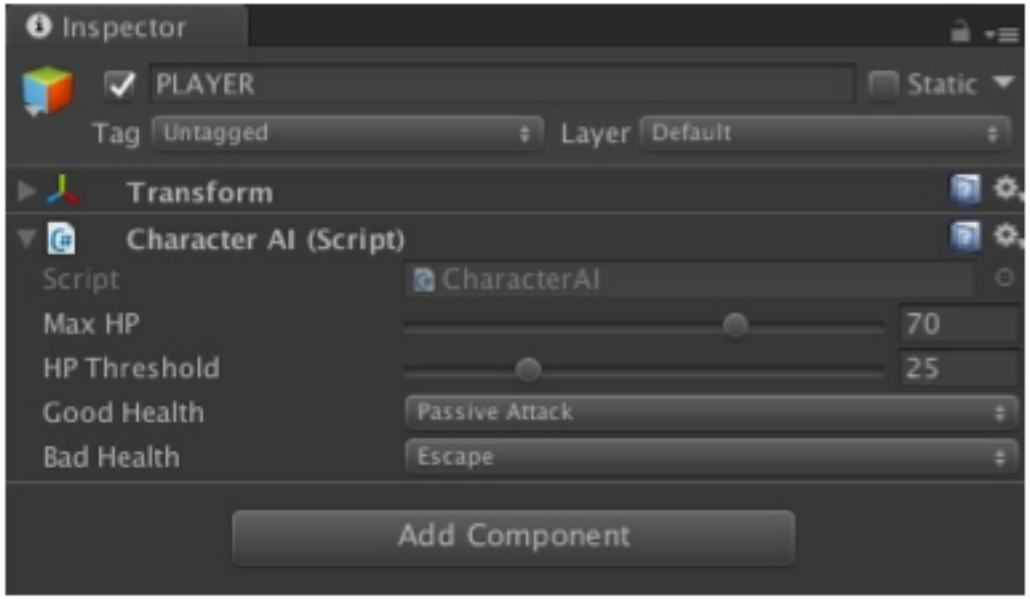


- キャラクターのPrefabをScriptableObjectに変更してみる
  - 最大HPとATKとDEFを持つ
  - HPの状況によって行う挙動


```
public class CharacterAI : MonoBehaviour
{
  [SerializeField, Range (0, 100)]
  int MaxHP = 100;

  [SerializeField, Range (0, 100)]
```



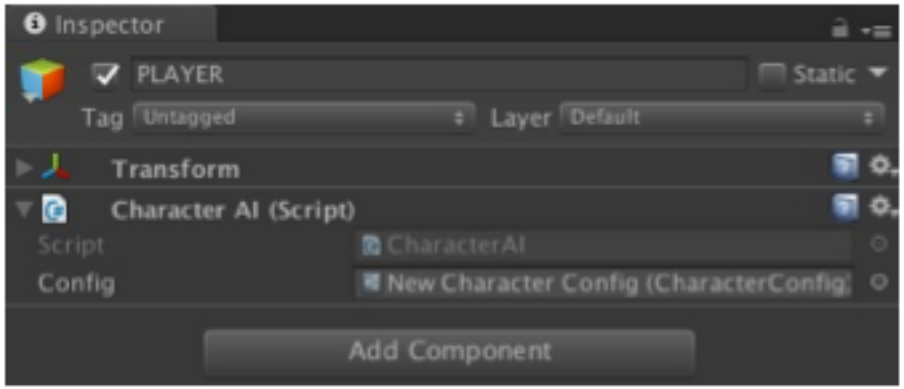
```
{
  [SerializeField, Range (0, 100)]
  int MaxHP = 100;

  [SerializeField, Range (0, 100)]
  int HPThreshold = 25;

  public int hp{ get; set; }

  public AIState goodHealth;
  public AIState badHealth;
}
```



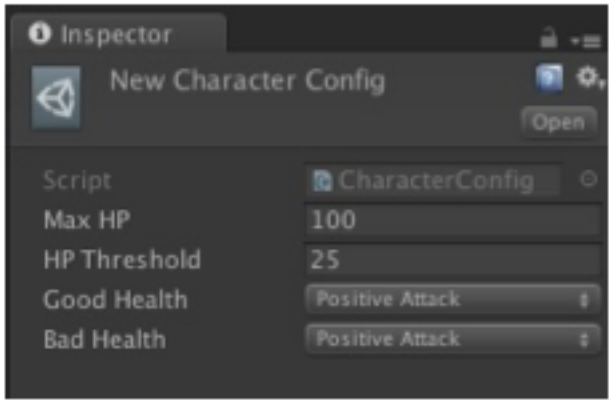

```
public class CharacterAI : MonoBehaviour
{
  [SerializeField]
  CharacterConfig config;

  public int hp{ get; set; }
}
```

```
[CreateAssetMenu]
public class CharacterConfig :
ScriptableObject
{
  public int MaxHP = 100;
  public int HPThreshold = 25;

  public AIState goodHealth;
```

```
  CharacterConfig config;

  public int hp{ get; set; }
}
```




```
{
  public int MaxHP = 100;
  public int HPThreshold = 25;

  public AIState goodHealth;
  public AIState badHealth;
}
```



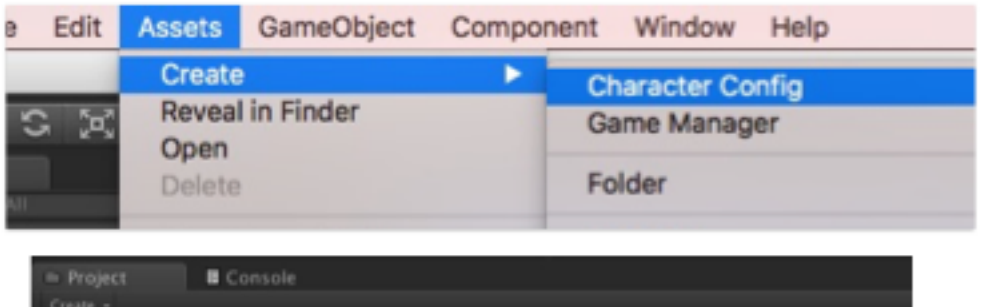

```
[CreateAssetMenu]
public class CharacterConfig :
ScriptableObject
{
  public AIState goodHealth;
```

Scriptable Objectを継承する

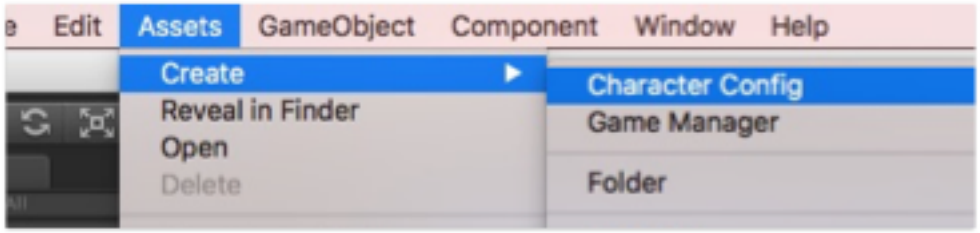
Scriptable Objectを継承する

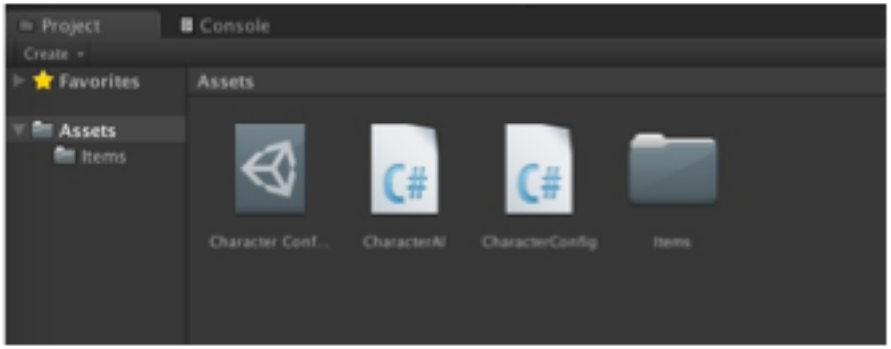
```
{
  public int MaxHP = 100;
  public AIState goodHealth;
  public AIState badHealth;
}
```

●ScriptableObjectからアセットを作成するメニューを追加




```
[CreateAssetMenu]
public class CharacterConfig :
ScriptableObject

    lic int MaxHP = 100;
    lic int HPThreshold = 25;

    lic AIState goodHealth;
```



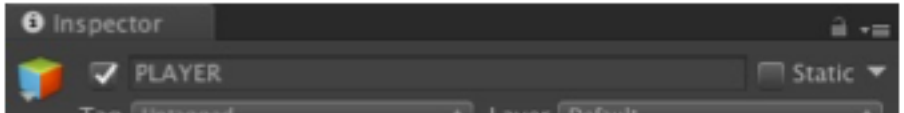


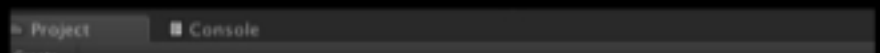

```
lic int MaxHP = 100;
lic int HPThreshold = 25;

lic AIState goodHealth;
lic AIState badHealth;
```

●ScriptableObjectからアセットを作成するメニューを追加

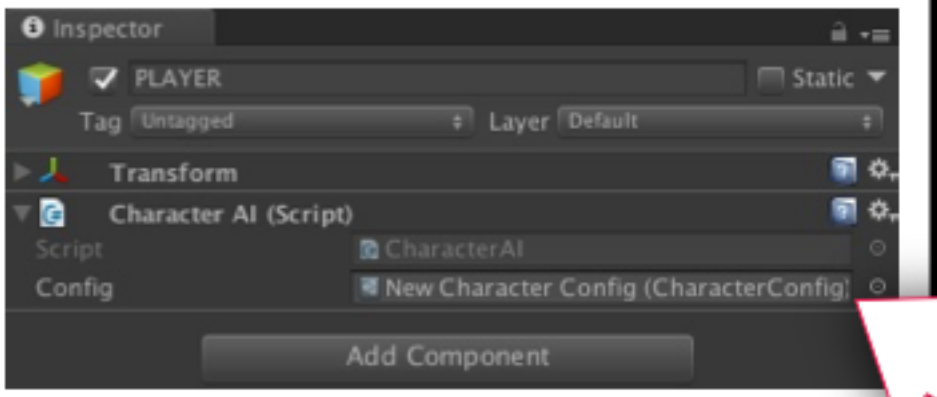
```
public class CharacterAI : MonoBehaviour
{
  [SerializeField]
  CharacterConfig config;

  public int hp{ get; set; }
}
```

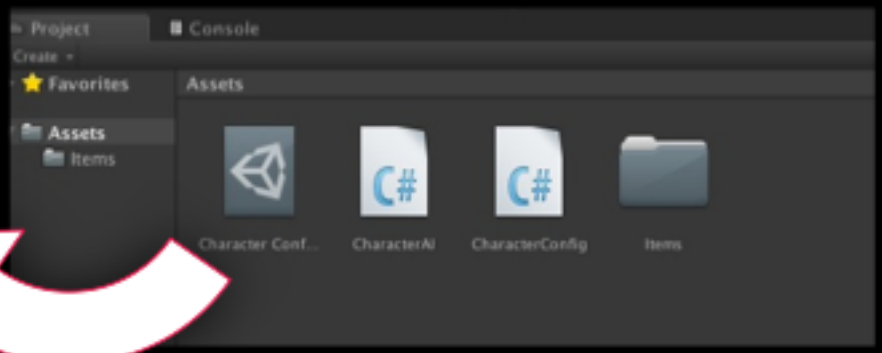
フィールドを公開





```
  public int hp{ get; set; }
}
```





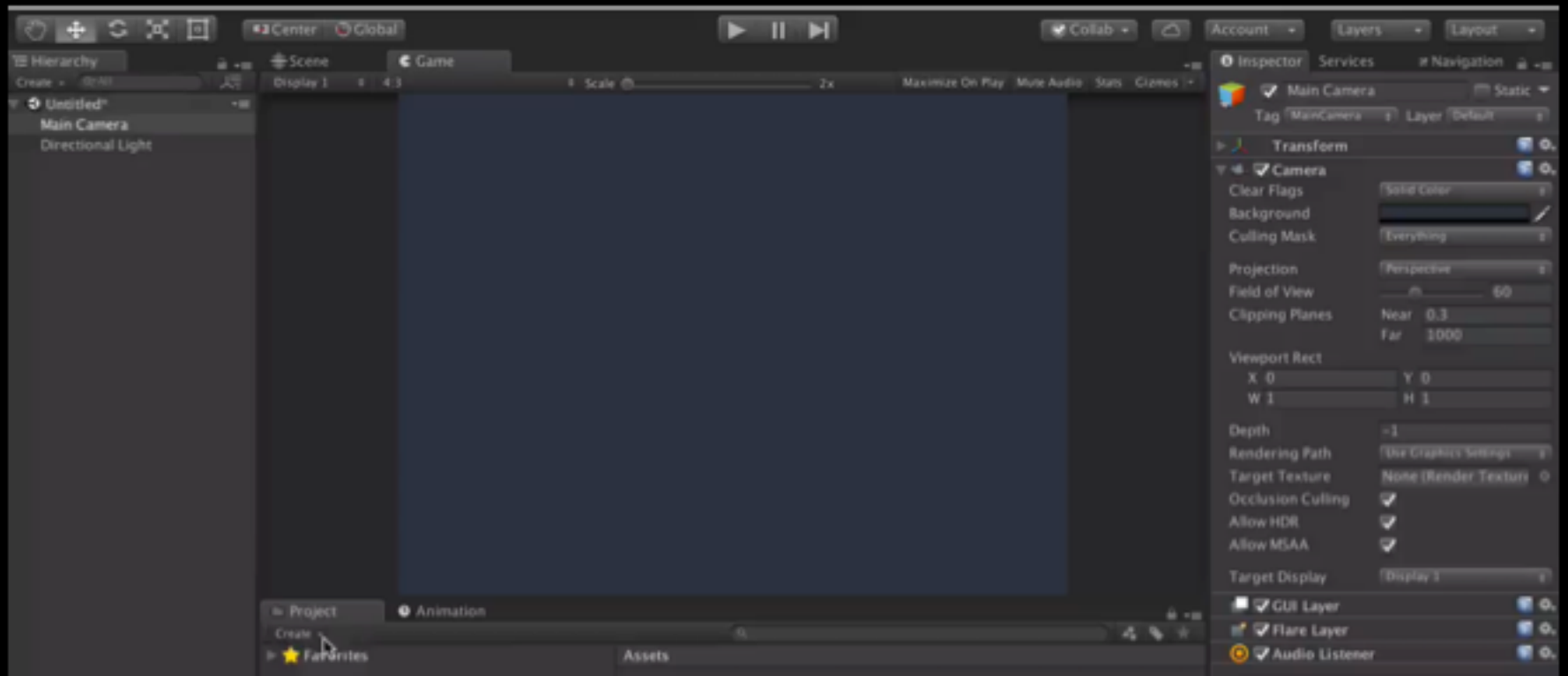
Scriptable Objectの持つコールバック

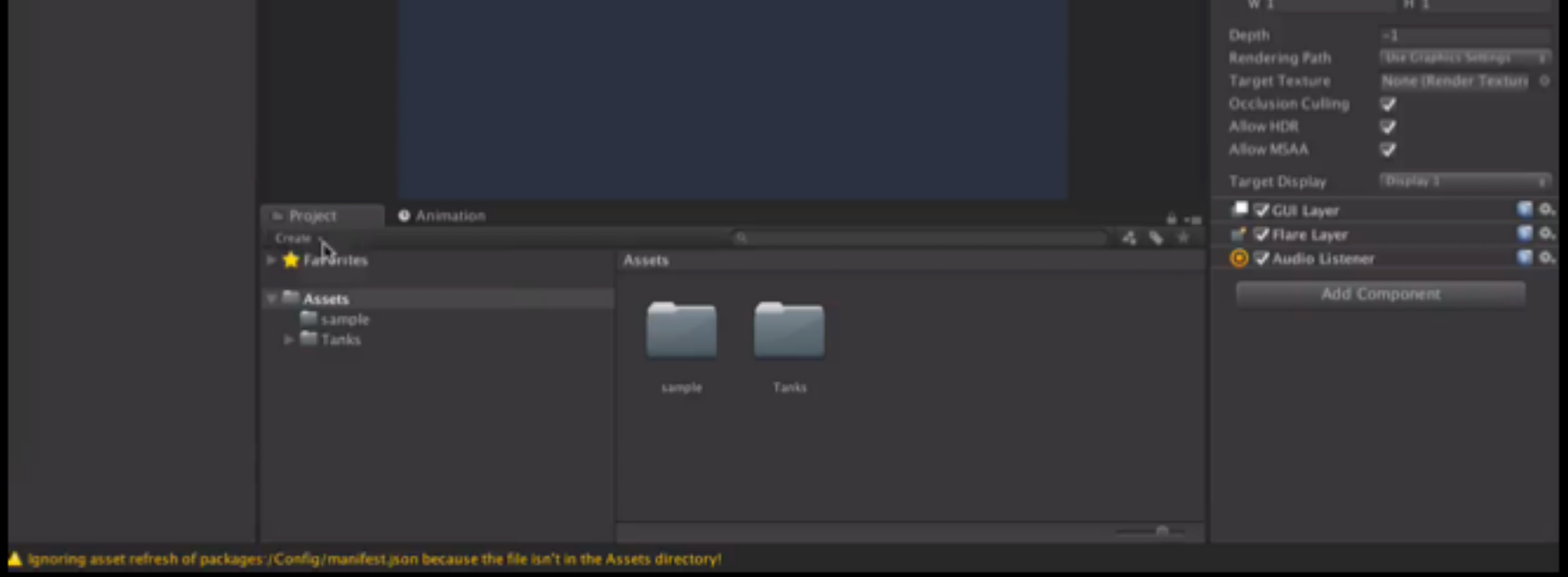
- Awake　スクリプト開始時に呼ばれる
- OnEnable　オブジェクトロード時に呼ばれる
- OnDisable　アンロード時に呼ばれる

- OnEnable　　オブジェクトロード時に呼ばれる
- OnDisable　　アンロード時に呼ばれる
- OnDestroy　　インスタンス破棄時に呼ばれる

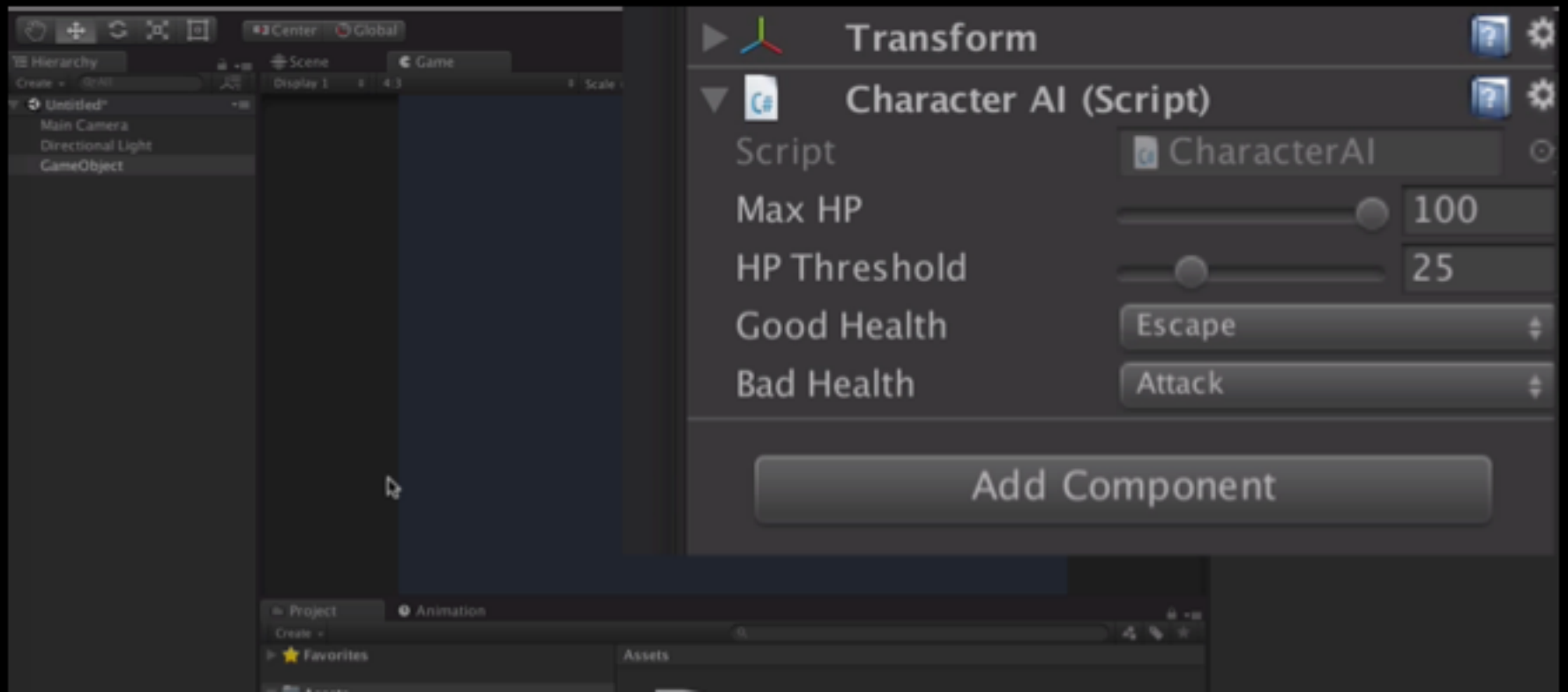
Demo

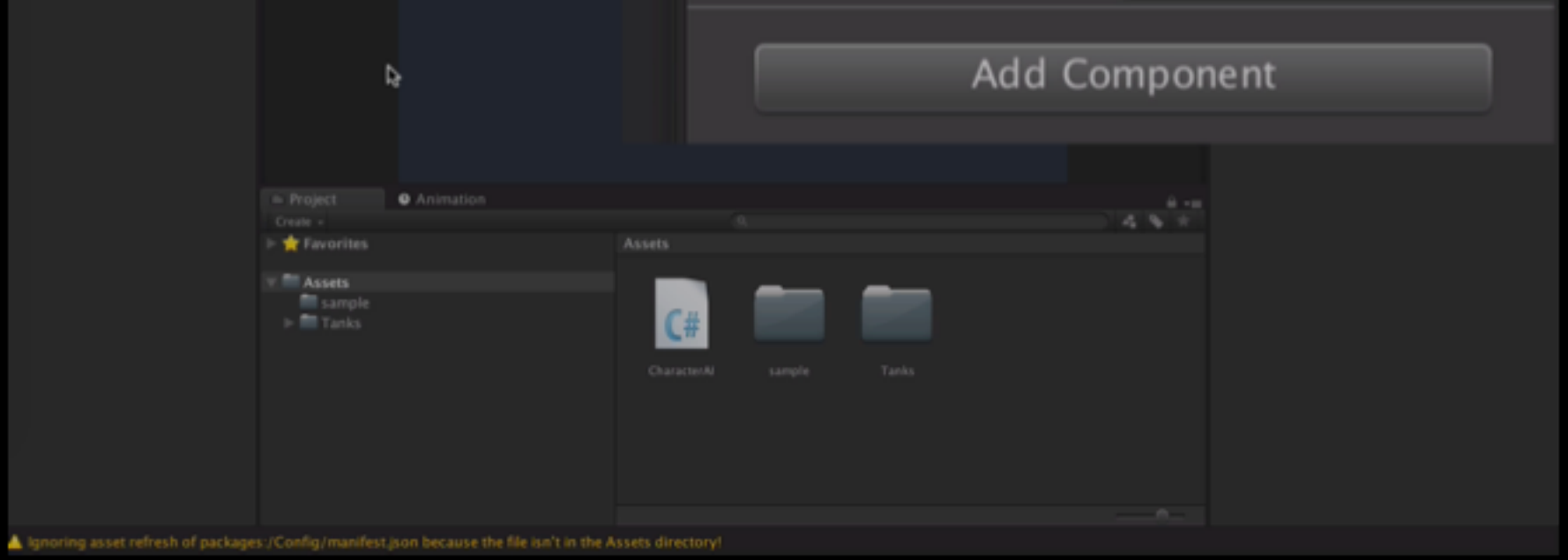
Demo


Hierarchy
Create
Untitled*
Main Camera
Directional Light
Scene
Game
Display 1
4:3
Scale
2x
Maximize On Play
Mute Audio
Stats
Gizmos
Center
Global
Collab
Account
Layers
Layout
Inspector
Services
Navigation
Main Camera
Static
Tag
MainCamera
Layer
Default
Transform
Camera
Clear Flags
Solid Color
Background
Culling Mask
Everything
Projection
Perspective
Field of View
60
Clipping Planes
Near 0.3
Far 1000
Viewport Rect
X 0
Y 0
W 1
H 1
Depth
-1
Rendering Path
Use Graphics Settings
Target Texture
None (Render Texture)
Occlusion Culling
Allow HDR
Allow MSAA
Target Display
Display 1
GUI Layer
Flare Layer
Audio Listener
Project
Animation
Create
Favorites
Assets


Depth
Rendering Path
Target Texture
Occlusion Culling
Allow HDR
Allow MSAA
Target Display
GUI Layer
Flare Layer
Audio Listener
Add Component
Project
Animation
Create
Favorites
Assets
sample
Tanks
Ignoring asset refresh of packages :/Config/manifest.json because the file isn't in the Assets directory!

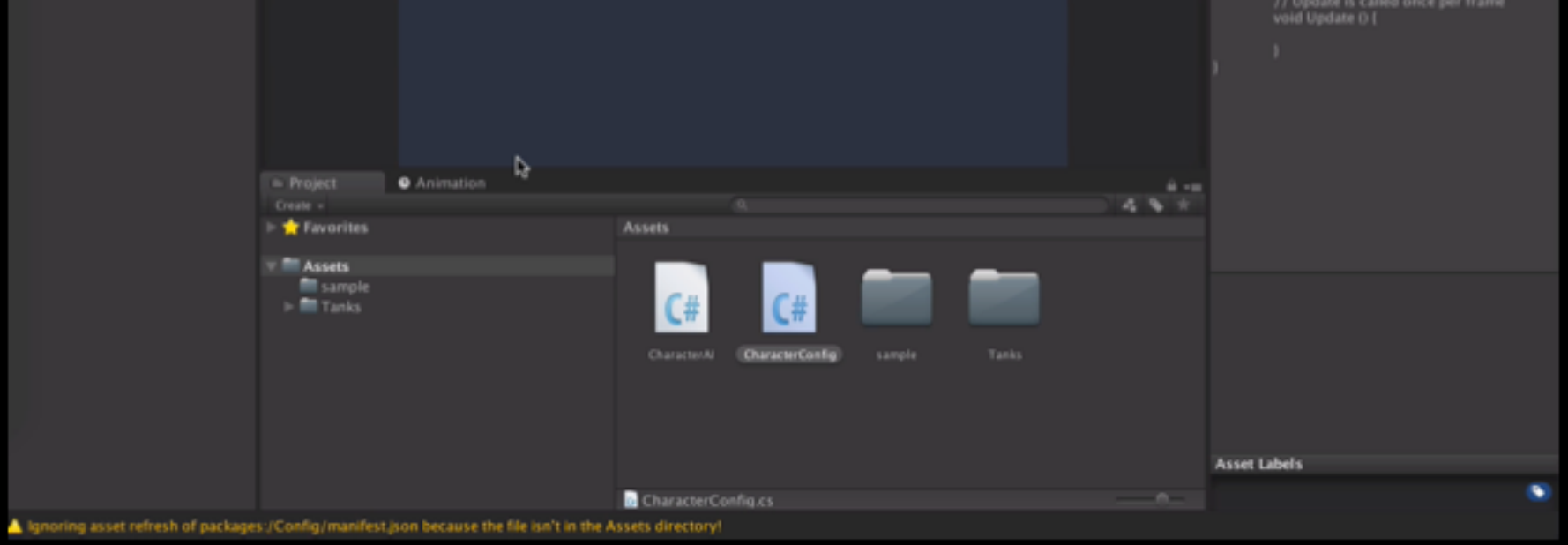
unity


Hierarchy
Scene
Game
Untitled*
Main Camera
Directional Light
GameObject
Transform
Character AI (Script)
Script
CharacterAI
Max HP
100
HP Threshold
25
Good Health
Escape
Bad Health
Attack
Add Component
Project
Animation
Favorites
Assets


Add Component
Project
Animation
Create
Favorites
Assets
sample
Tanks
CharacterAI
sample
Tanks
Ignoring asset refresh of packages :/Config/manifest.json because the file isn't in the Assets directory!

unity


Center
Global
Collab
Account
Layers
Layout
Hierarchy
Create
Untitled*
Main Camera
Directional Light
GameObject
Scene
Game
Display 1
4:3
Scale
2x
Maximize On Play
Mute Audio
Stats
Gizmos
Inspector
Services
Navigation
CharacterConfig Import Settings
Open...
Execution Order...
Imported Object
CharacterConfig
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
public class CharacterConfig : MonoBehaviour {
// Use this for initialization
void Start () {
}
// Update is called once per frame
void Update () {
}
}
Project
Animation
Create
Favorites
Assets
Assets



unity

で？

Scriptable Objectは

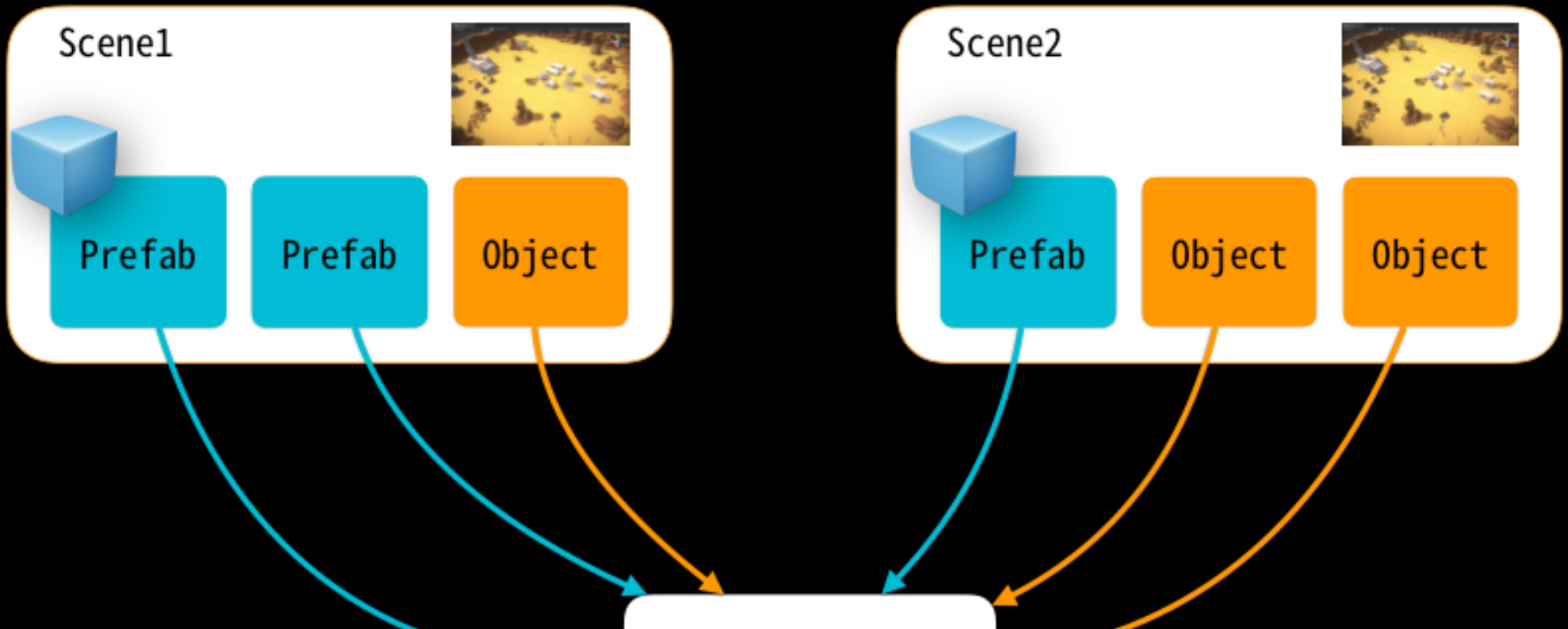
で？

Scriptable Objectは
何の役に立つの？

例えばScriptable Object活用のアイディア

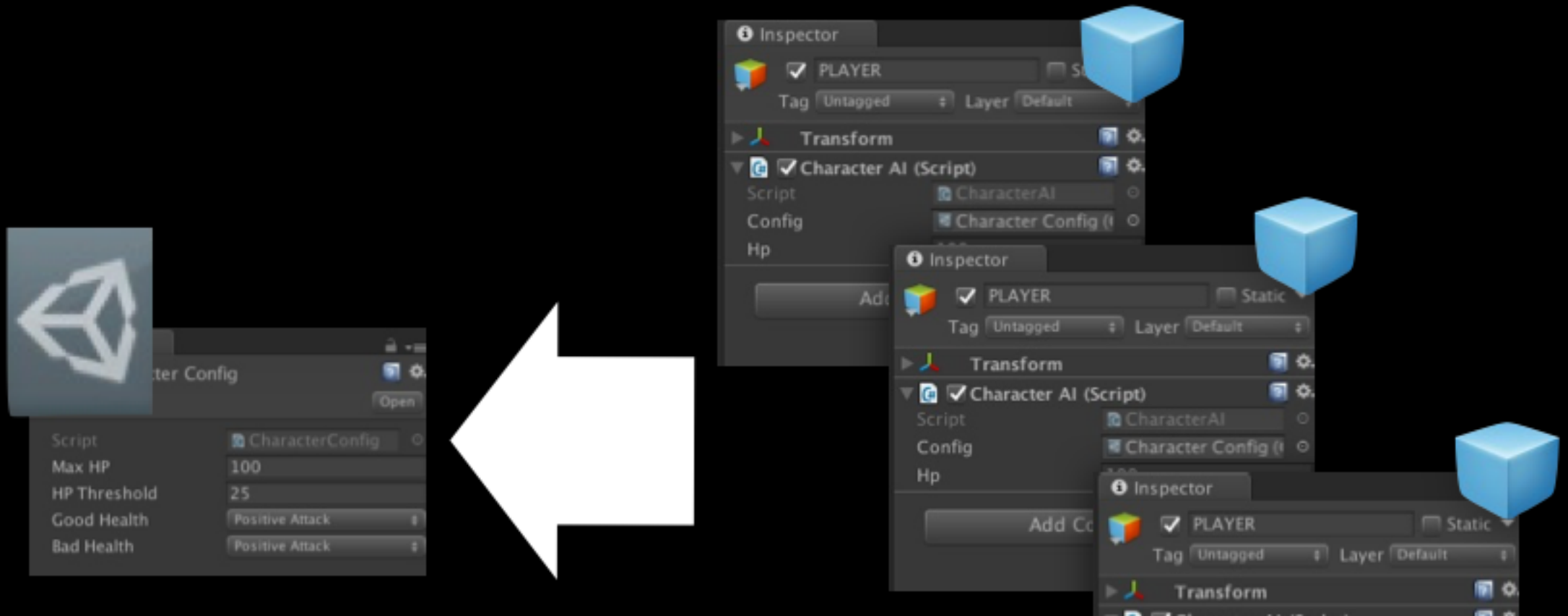
- シーン間・GameObject間で使いまわせるデータテーブル
- ロード・インスタンス化の高速化
- 動作の切り替え
- ステータスの共有

- 動作の切り替え
- ステータスの共有
- オブジェクトのバインド

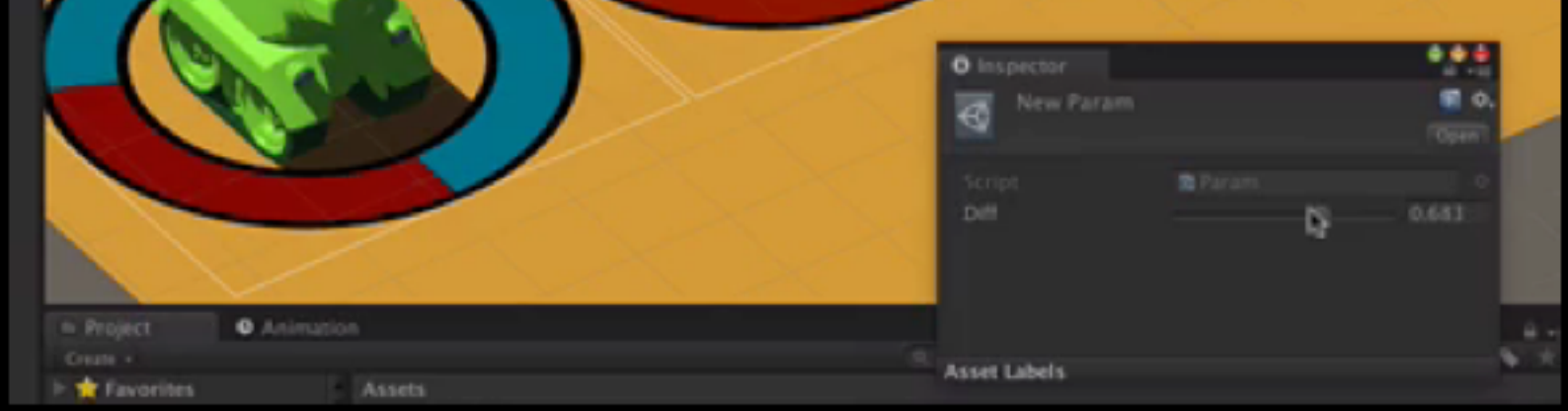


ScriptableObject

●Scene間で使用するインスタンスを共有



●同じSOを参照するならば、同じインスタンスを参照

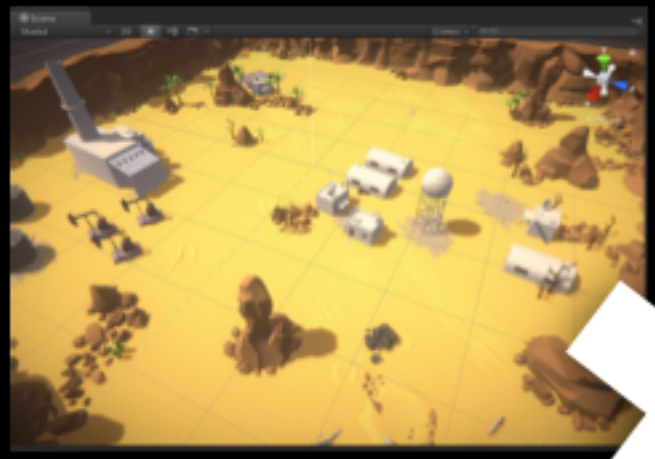


●Scriptable Objectのパラメータを変更は、参照元Prefabにも反映

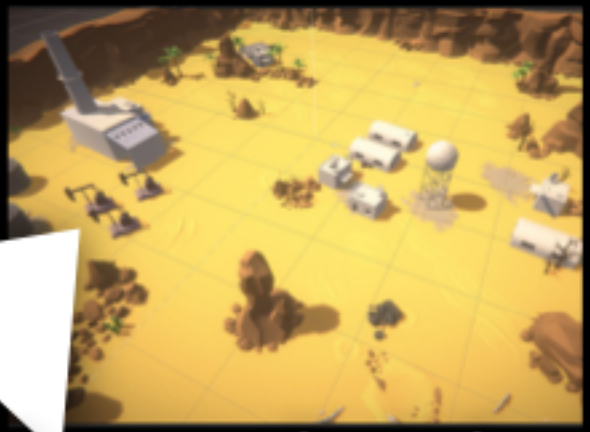
SPEED UP!!

SPEED UP!!

●ScriptableObjectのデータ共有は高速化にもメリット



Serialize

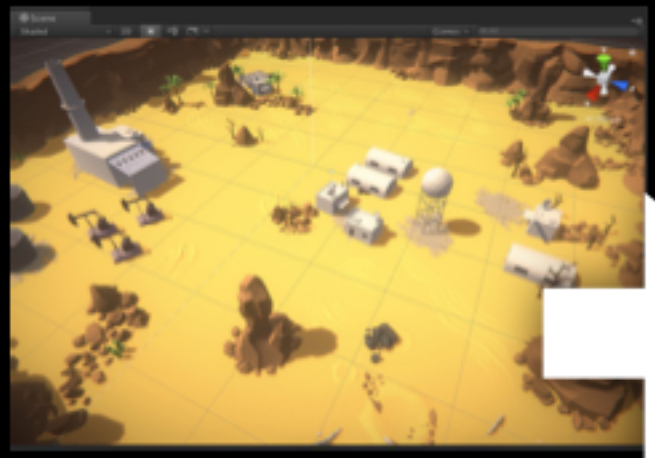


Deserialize

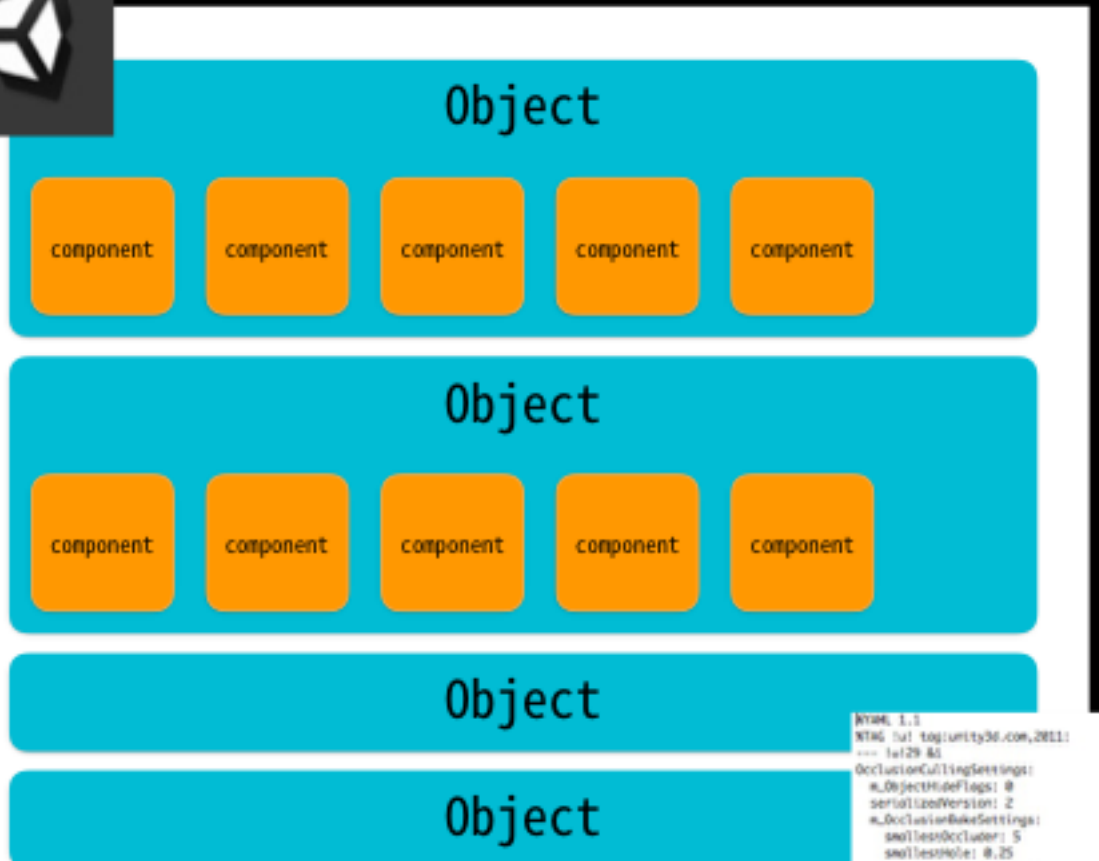
Serialize

Deserialize

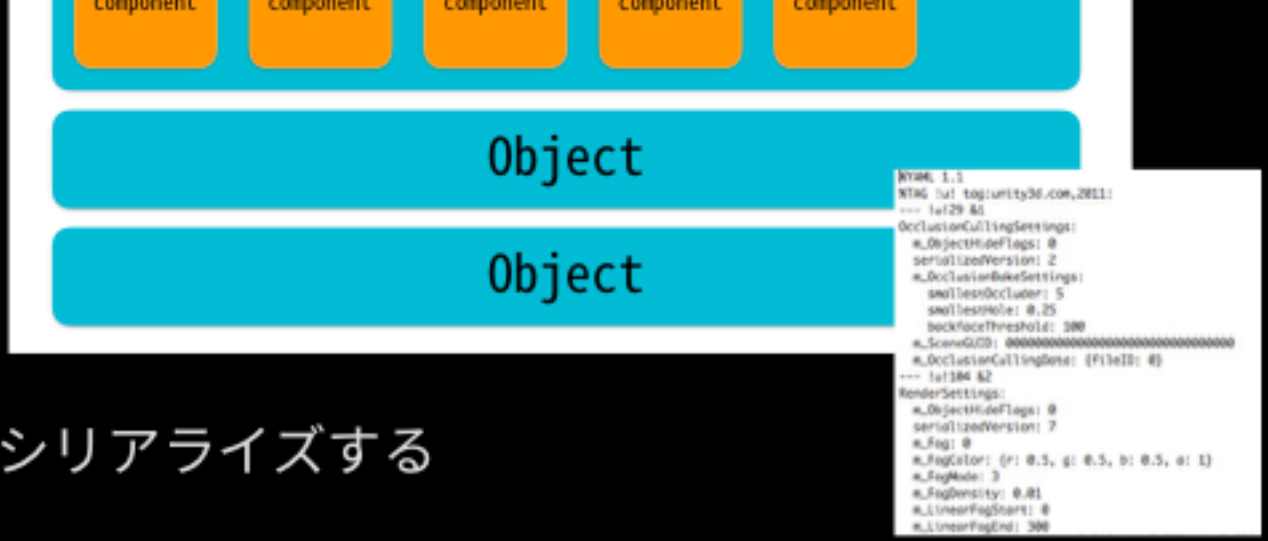
●Unityはシーンやprefabを丸ごとシリアライズし、丸ごとデシリアライズする



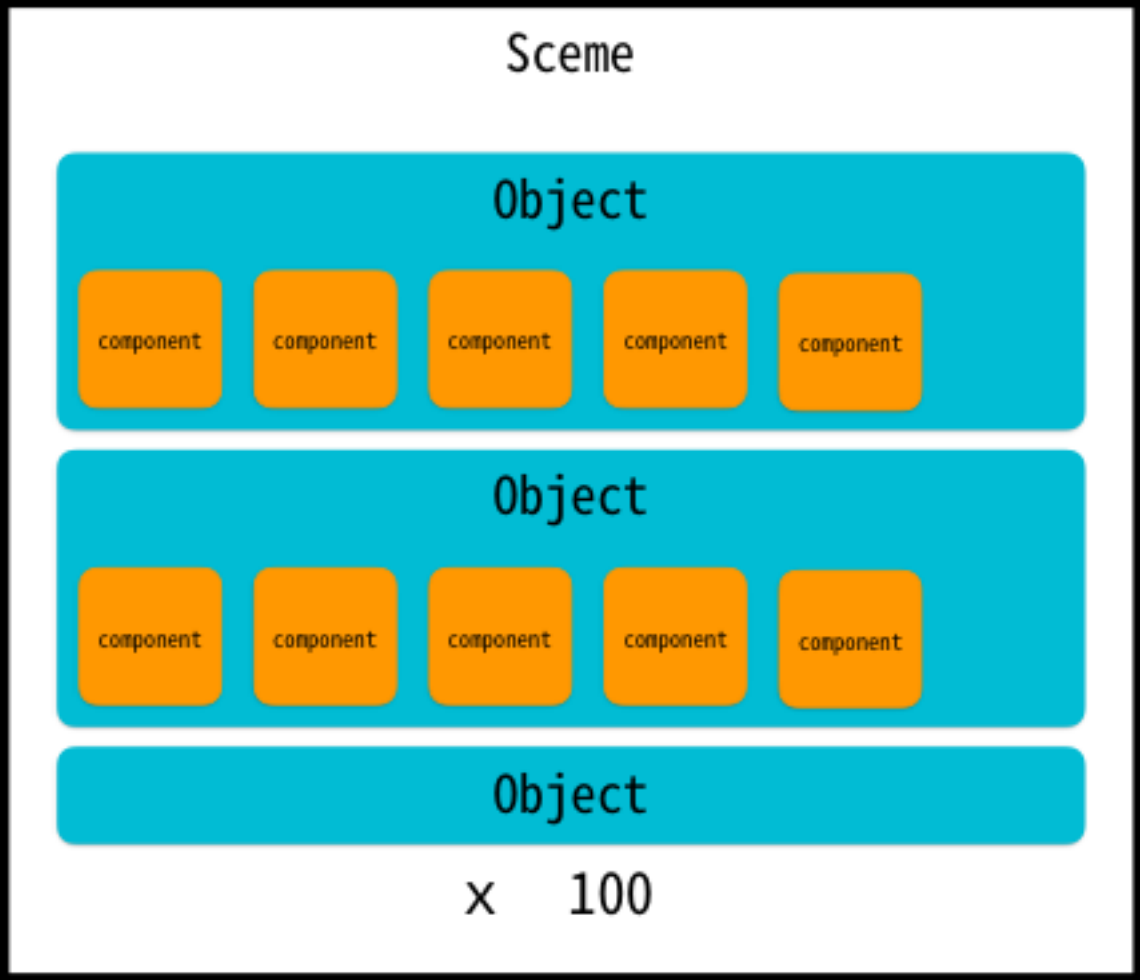
Serialize



Serialize



●シリアライズは本当にそのままシリアライズする

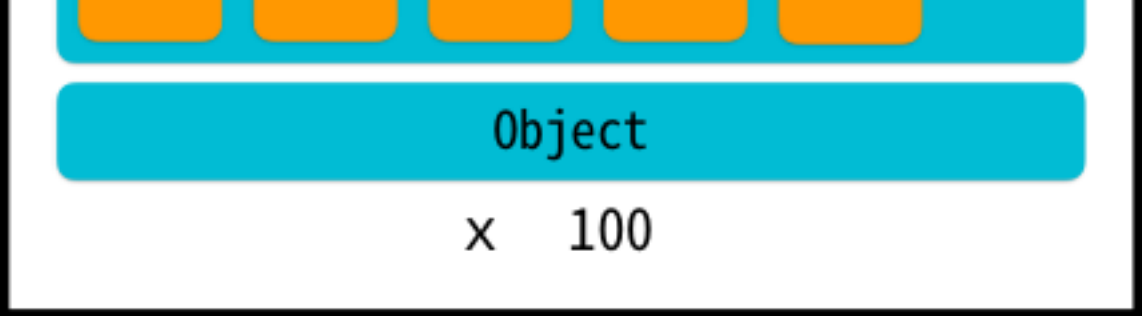


オブジェクト ： 102個

コンポーネント： 510個

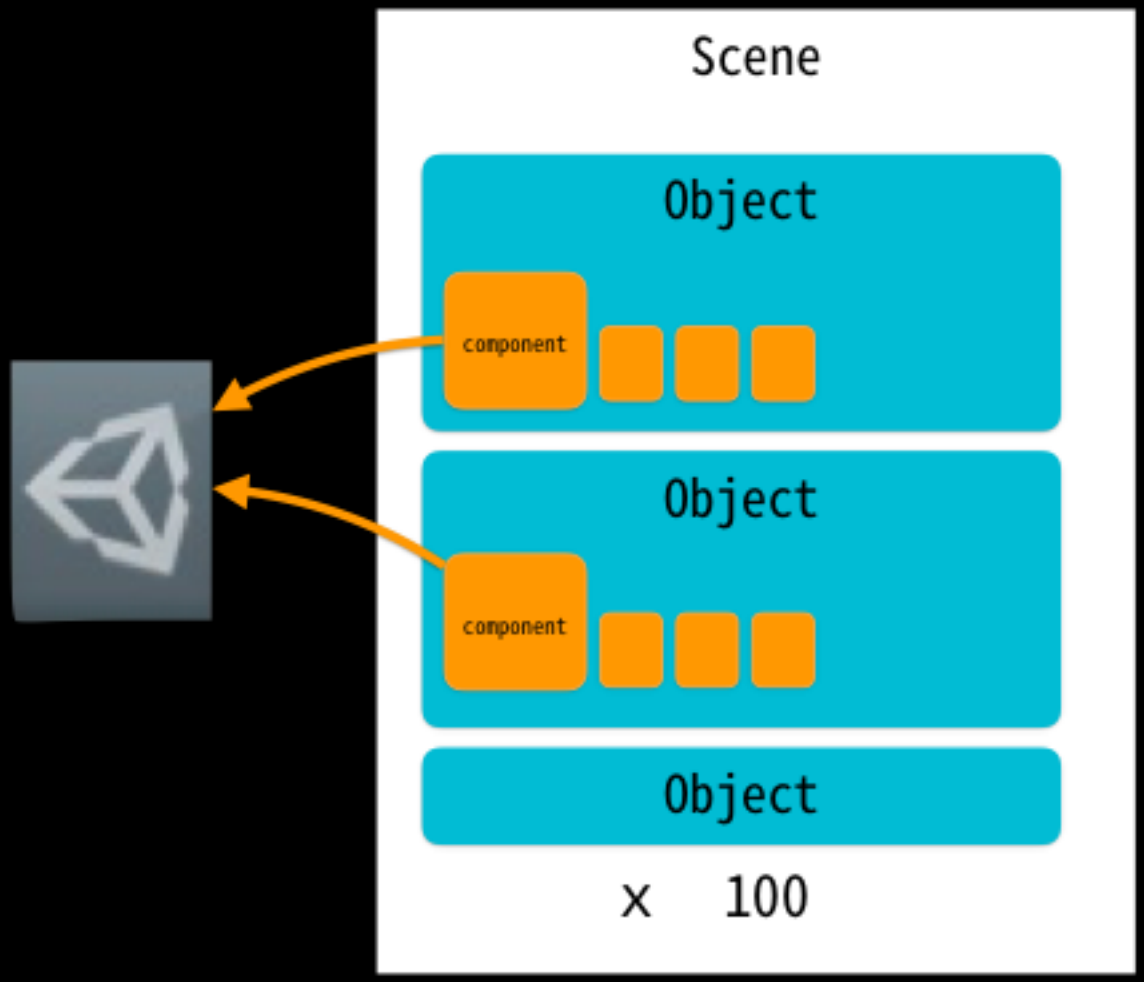
フィールド数： すごい

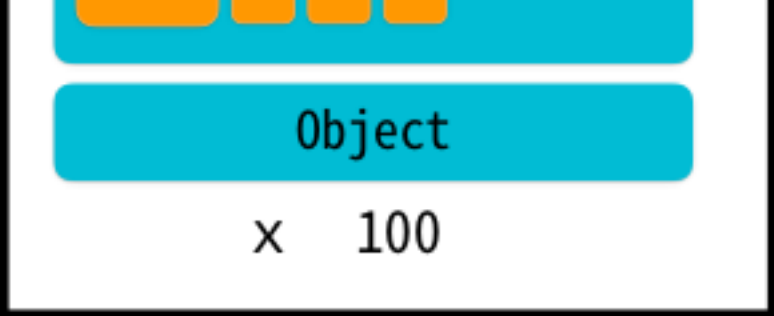
デシリアライズのコスト
プライスレス



デシリアライズのコスト
プライスレス

●完全に同一のオブジェクトがあっても、丸ごとシリアライズ





●共通項目はScriptableObjectを使い、デシリアライズの項目を減らす

staticでは駄目なのか？

●シリアライズ出来ない為ホットリロード時にリセットされる

●Inspector等から調整するには特別なコードが必要

●アクセスするにはスクリプトが必要

●同一スクリプト異Prefabで少し面倒くさい

●スクリプトで全て管理するなら、デメリットは

●同一スクリプト異Prefabで少し面倒くさい

●スクリプトで全て管理するなら、デメリットは
無視出来るかもしれない

staticでは駄目なのか？

●シリアライズ出来ない為ホットリロード時にリセットされる

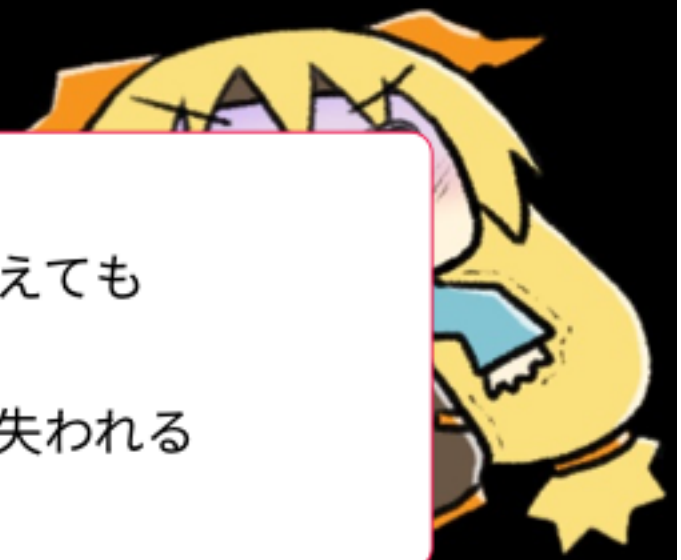
●Inspector等から調整するには特別なコードが必要

●アクセス

●同一スク

●スクリプトで全て管理するなら、デメリットは

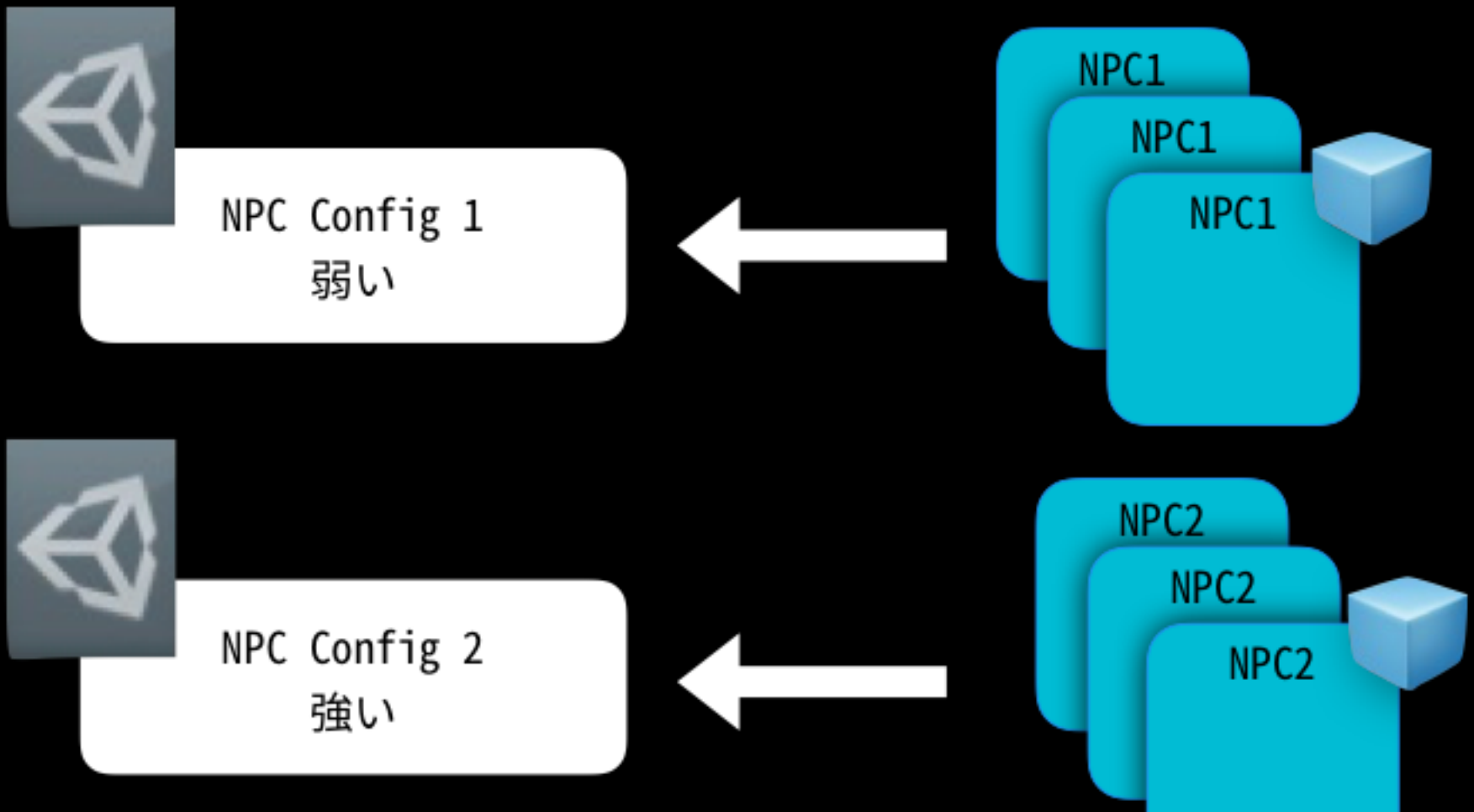
プログラム実行中にプログラムを書き換えても
即座に動作に反映される挙動。
static等のシリアライズ出来ない項目は失われる



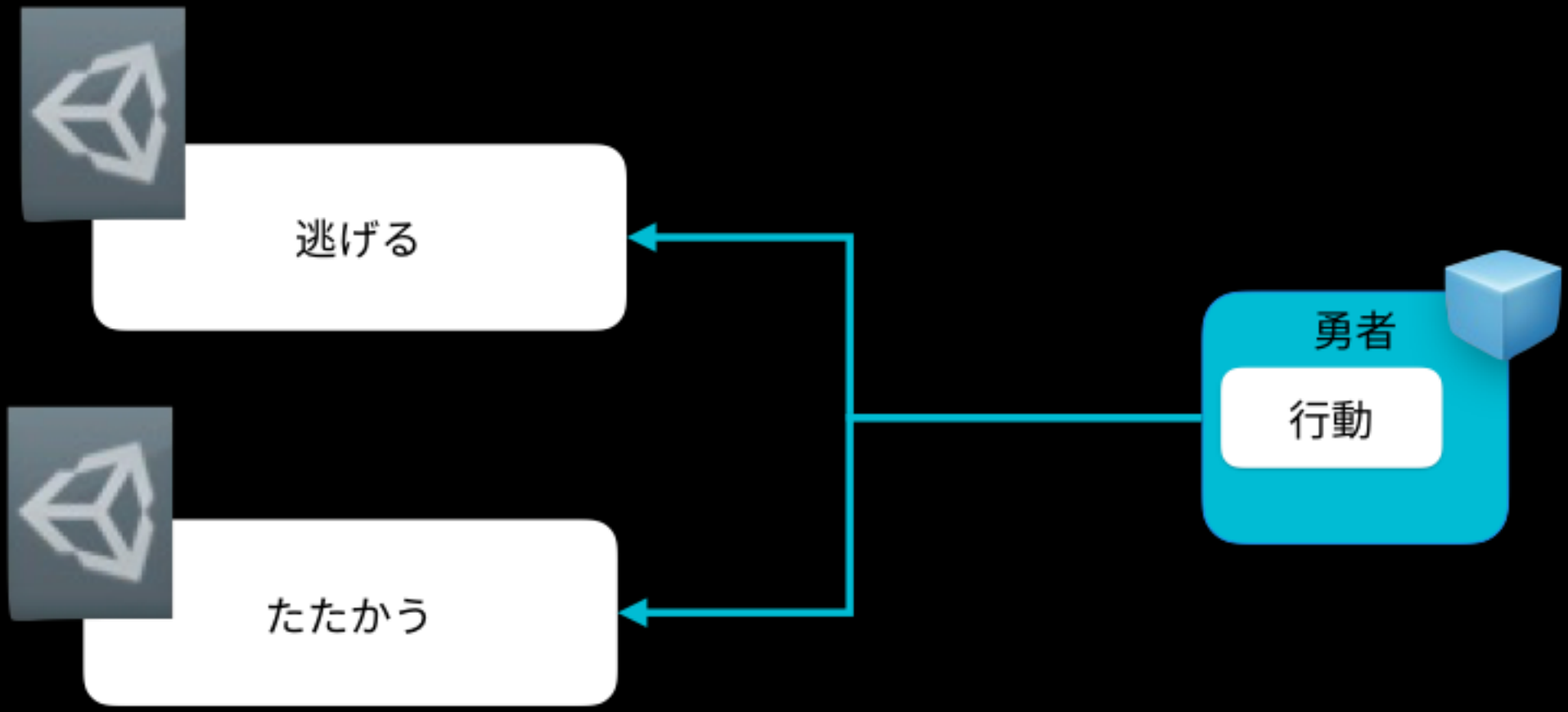
●同一スク

即座に動作に反映される挙動。
static等のシリアライズ出来ない項目は失われる

●スクリプトで全て管理するようにすればシリアライズは
無視出来るかもしれない

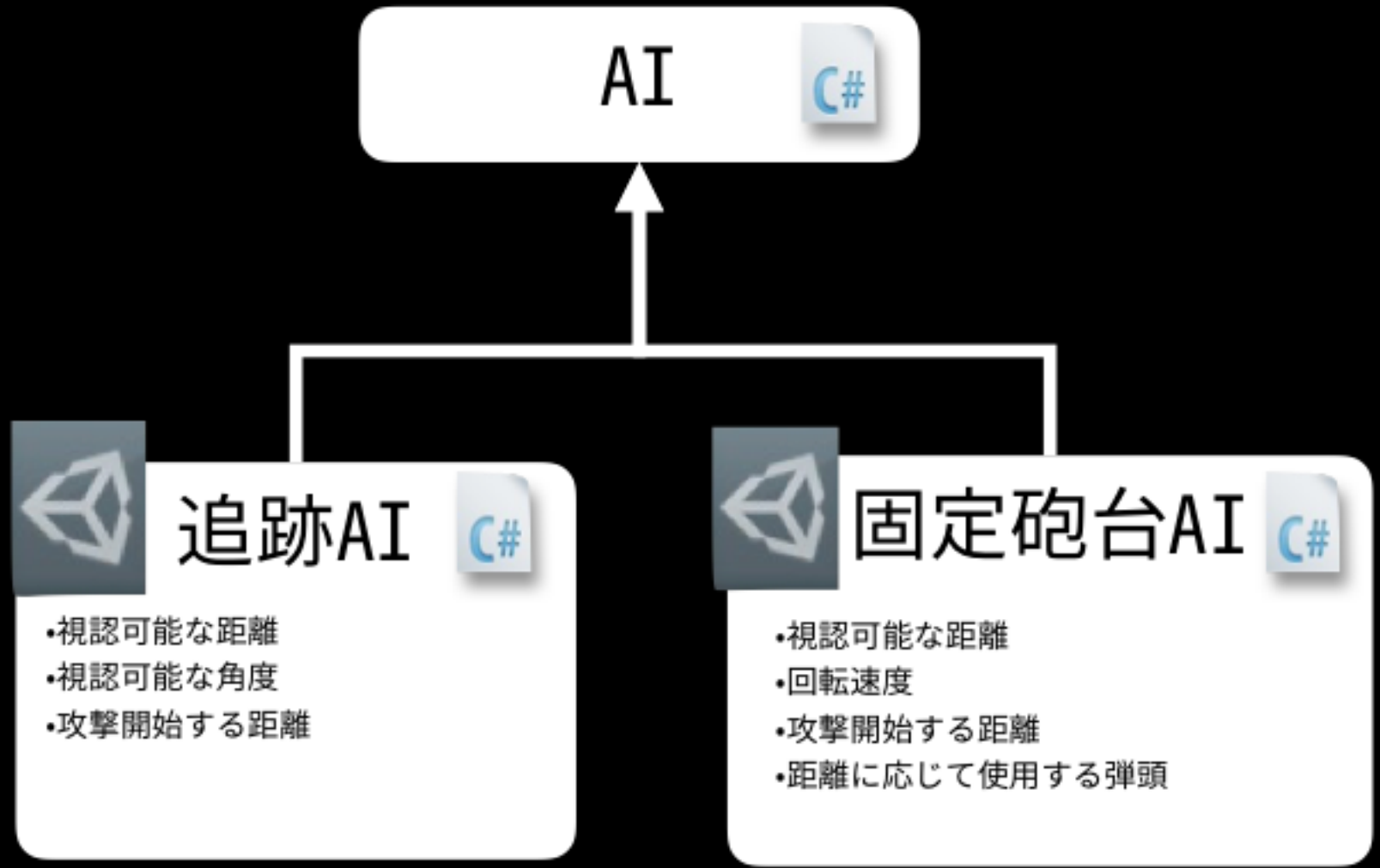


●同一ScriptでもSOを切り替える事でパラメータを切替



たたかう

●挙動の差替もできる

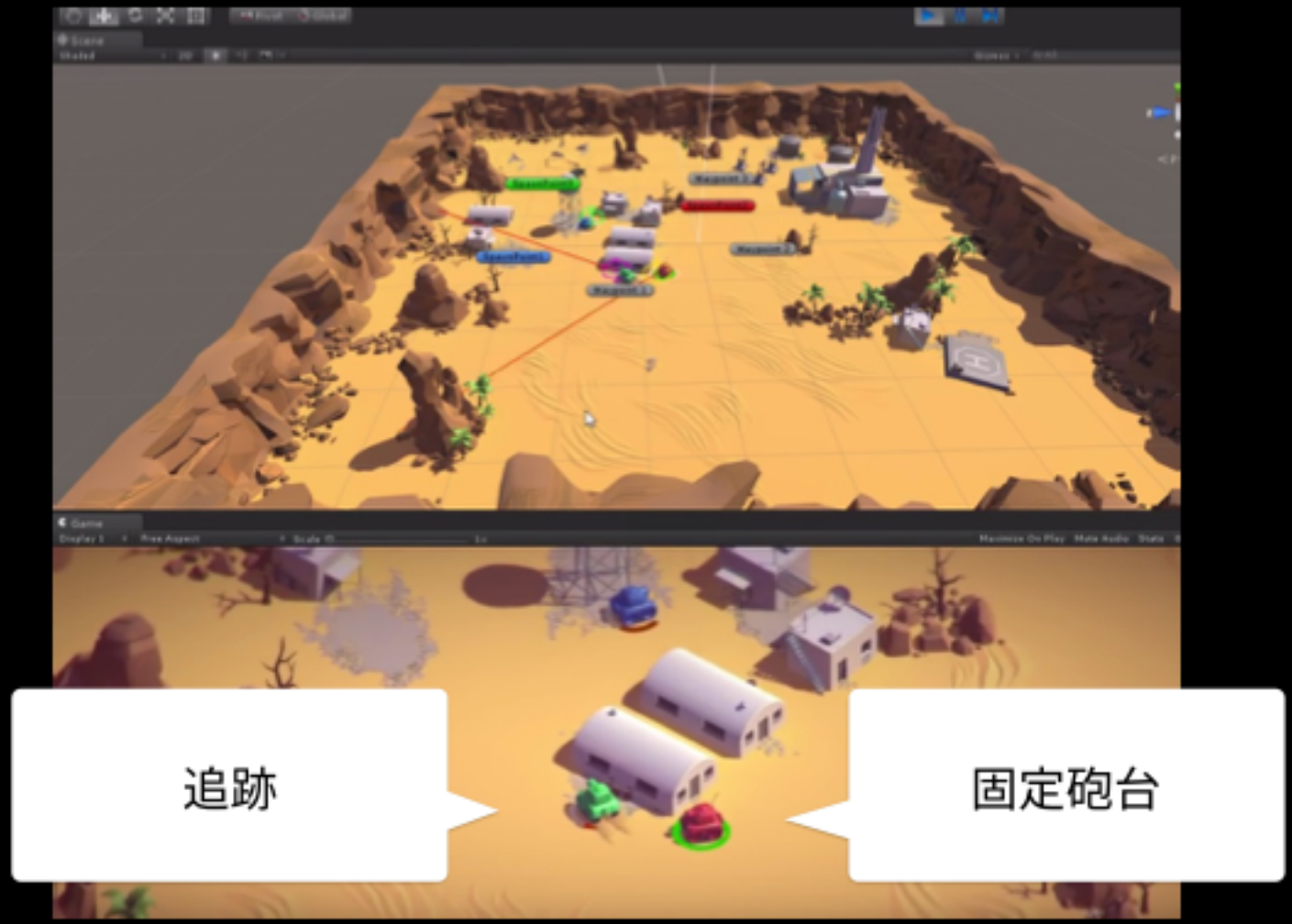


●パラメータと振る舞いを設定

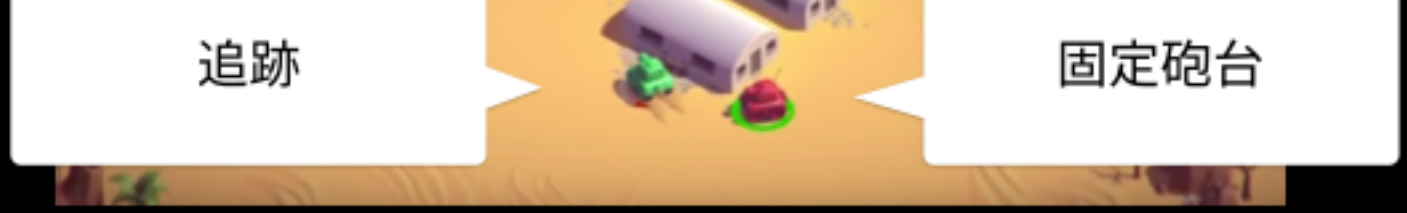
•視認可能な角度
•攻撃開始する距離

•回転速度
•攻撃開始する距離
•距離に応じて使用する弾頭

●パラメータと振る舞いを設定



●ユニット毎にAIを設定



●ユニット毎にAIを設定

•HP
•使用するAI
•ターゲット

追跡AI

•視認可能な距離
•追跡範囲

OR

砲台AI

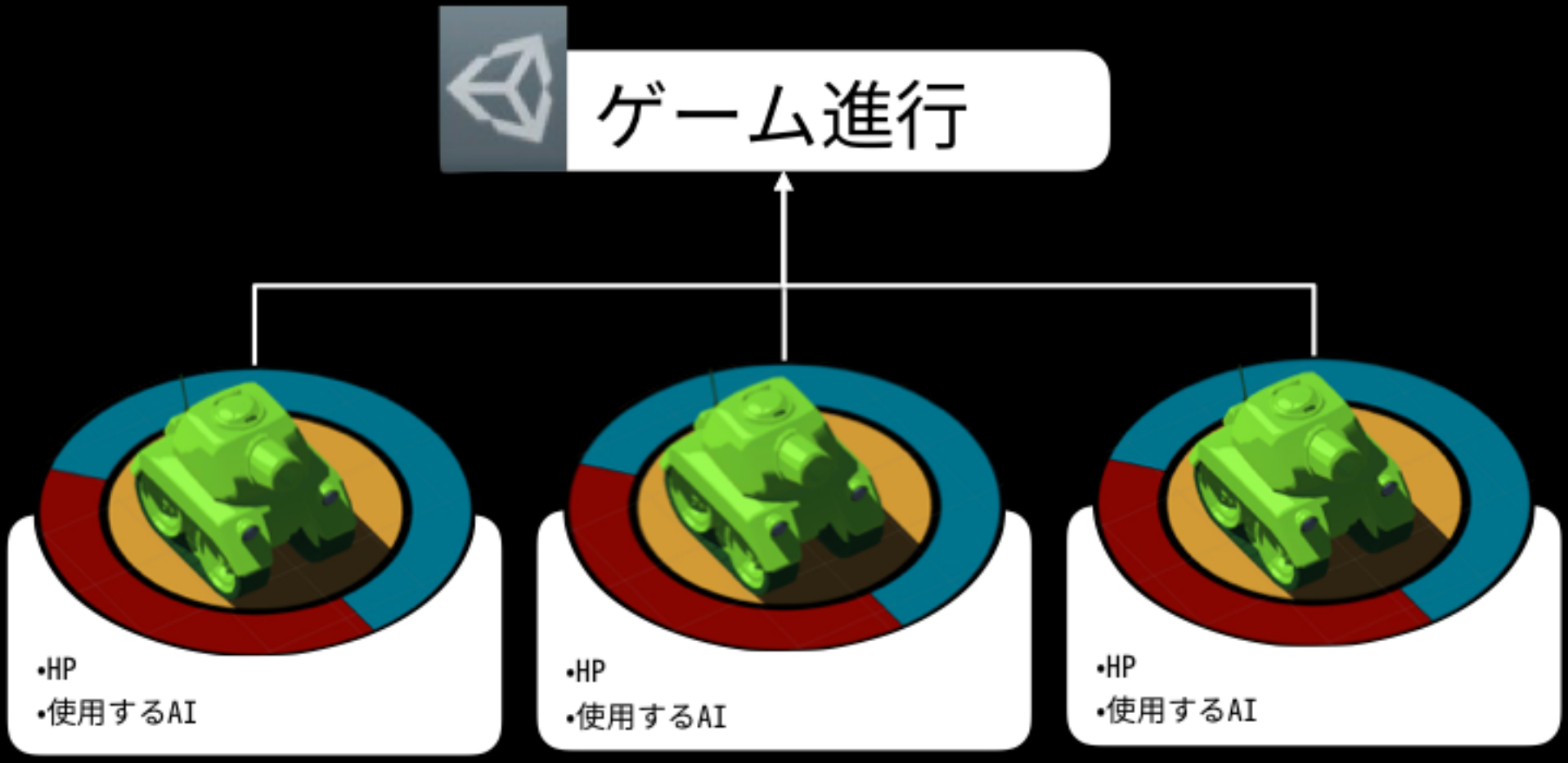
•視認可能な距離
•視認可能な角度
•攻撃開始する距離

●ユニット毎のパラメータはユニット自身が持つ

•視認可能な角度
•攻撃開始する距離

●ユニット毎のパラメータはユニット自身が持つ

（SOを上書きすると、全てのユニットに影響する為）



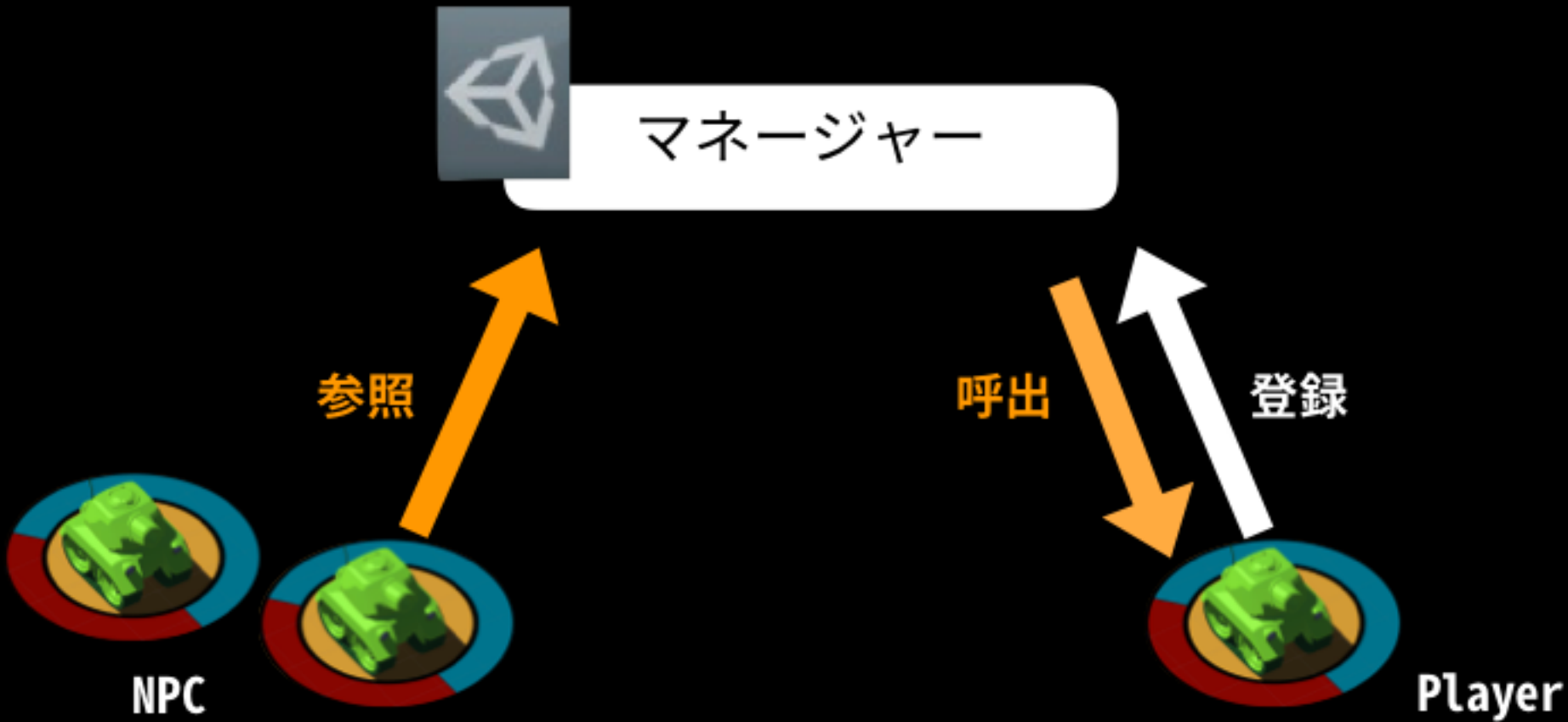
●インスタンスを共有

•HP
•使用するAI

•HP
•使用するAI

•HP
•使用するAI

●インスタンスを共有

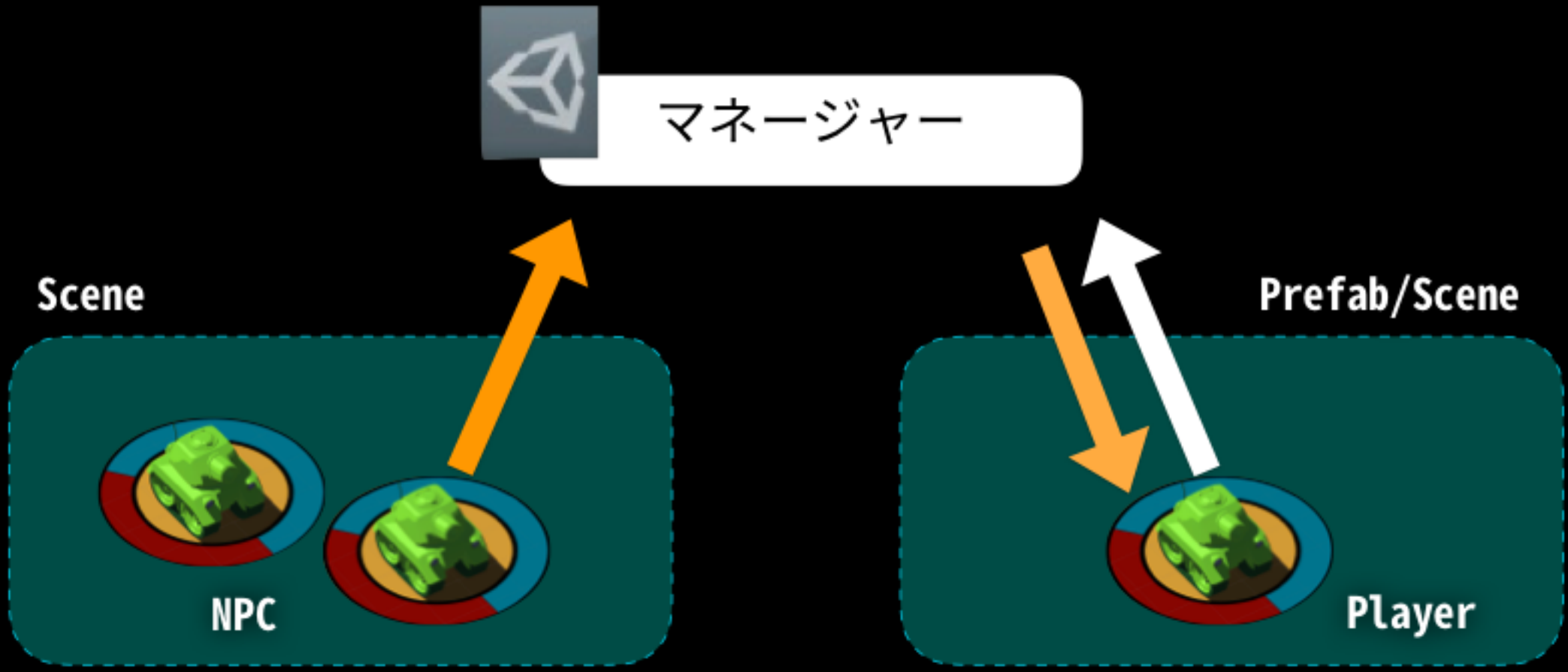


●ScriptableObjectにGameObjectを登録、ScriptableObject経由で呼び出す

NPC

Player

●ScriptableObjectにGameObjectを登録、ScriptableObject経由で呼び出す

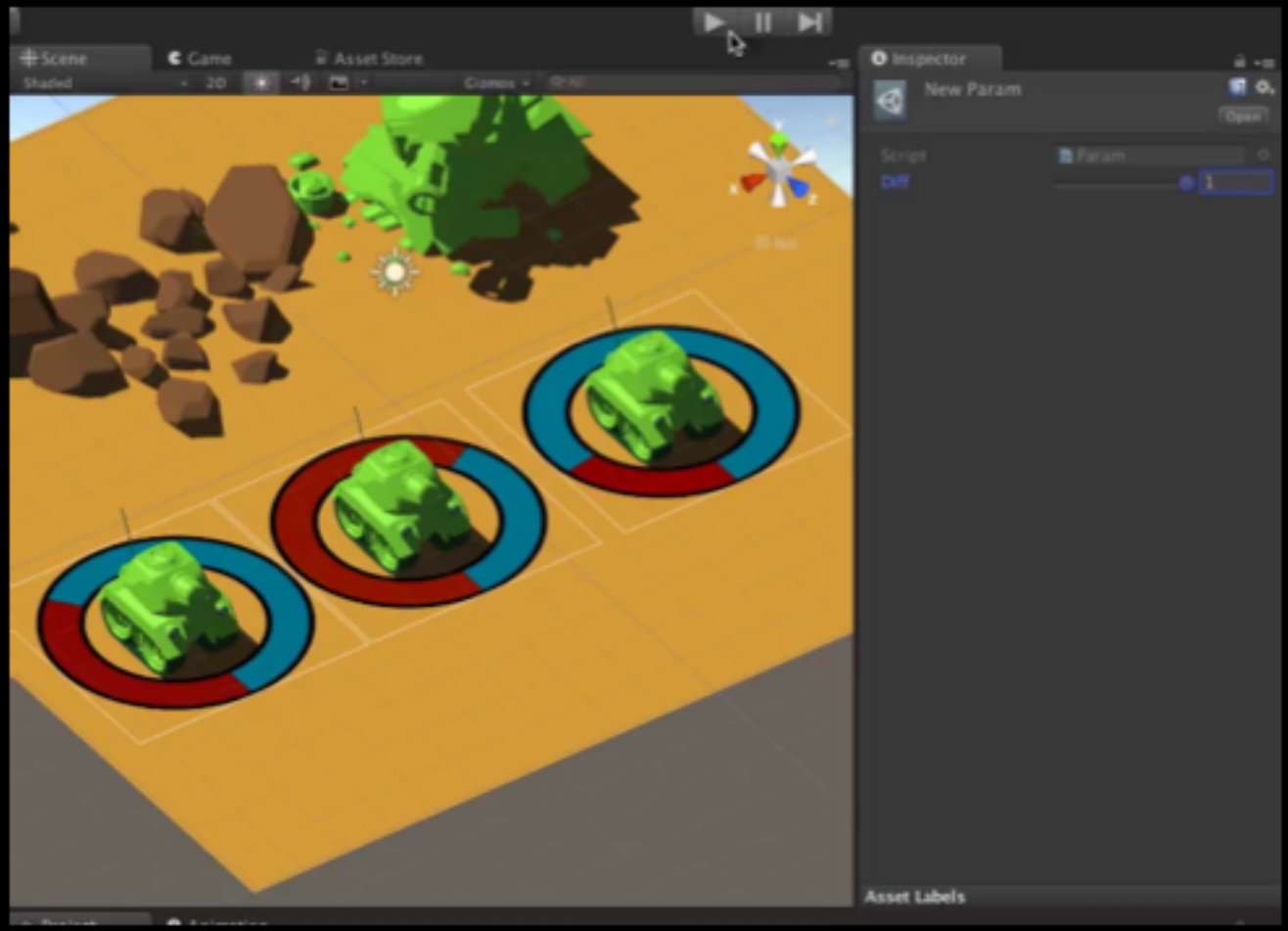


●シーンを跨いだオブジェクトも簡単に参照

NPC

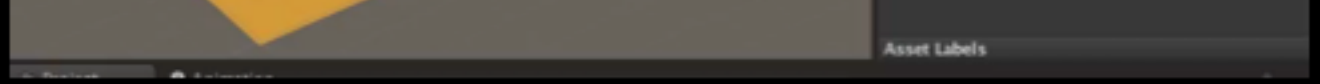
Player

●シーンを跨いだオブジェクトも簡単に参照



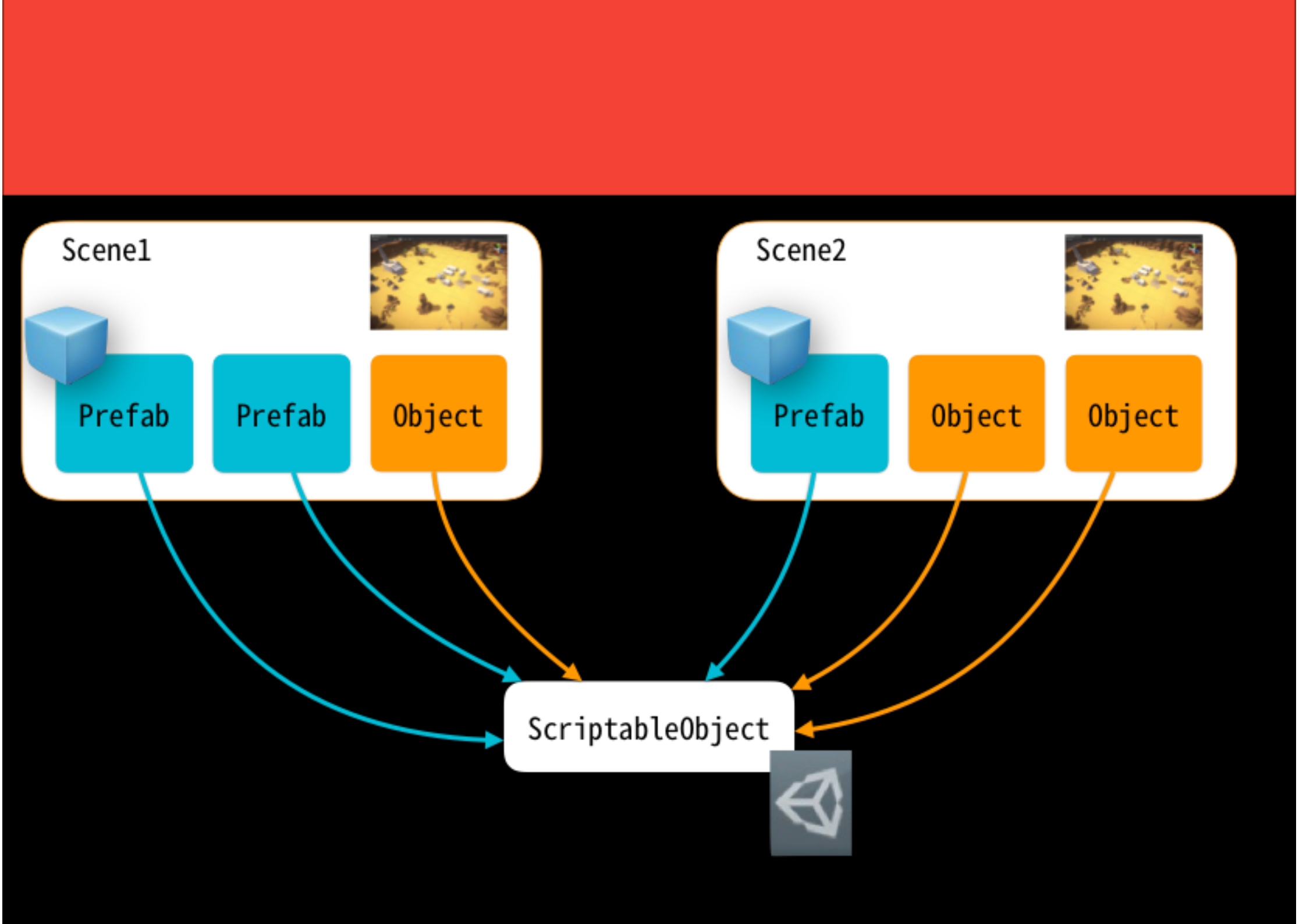
●ScriptableObjectはゲーム停止時にリセットされない

●ゲームをプレイしながらパラメータを調整



- ScriptableObjectはゲーム停止時にリセットされない
- ゲームをプレイしながらパラメータを調整

Demo


Scene1
Prefab
Prefab
Object
Scene2
Prefab
Object
Object
ScriptableObject

```
[CreateAssetMenu]
public class TankConfig : ScriptableObject
{
  public float speed;
  public float rotate;
}
```

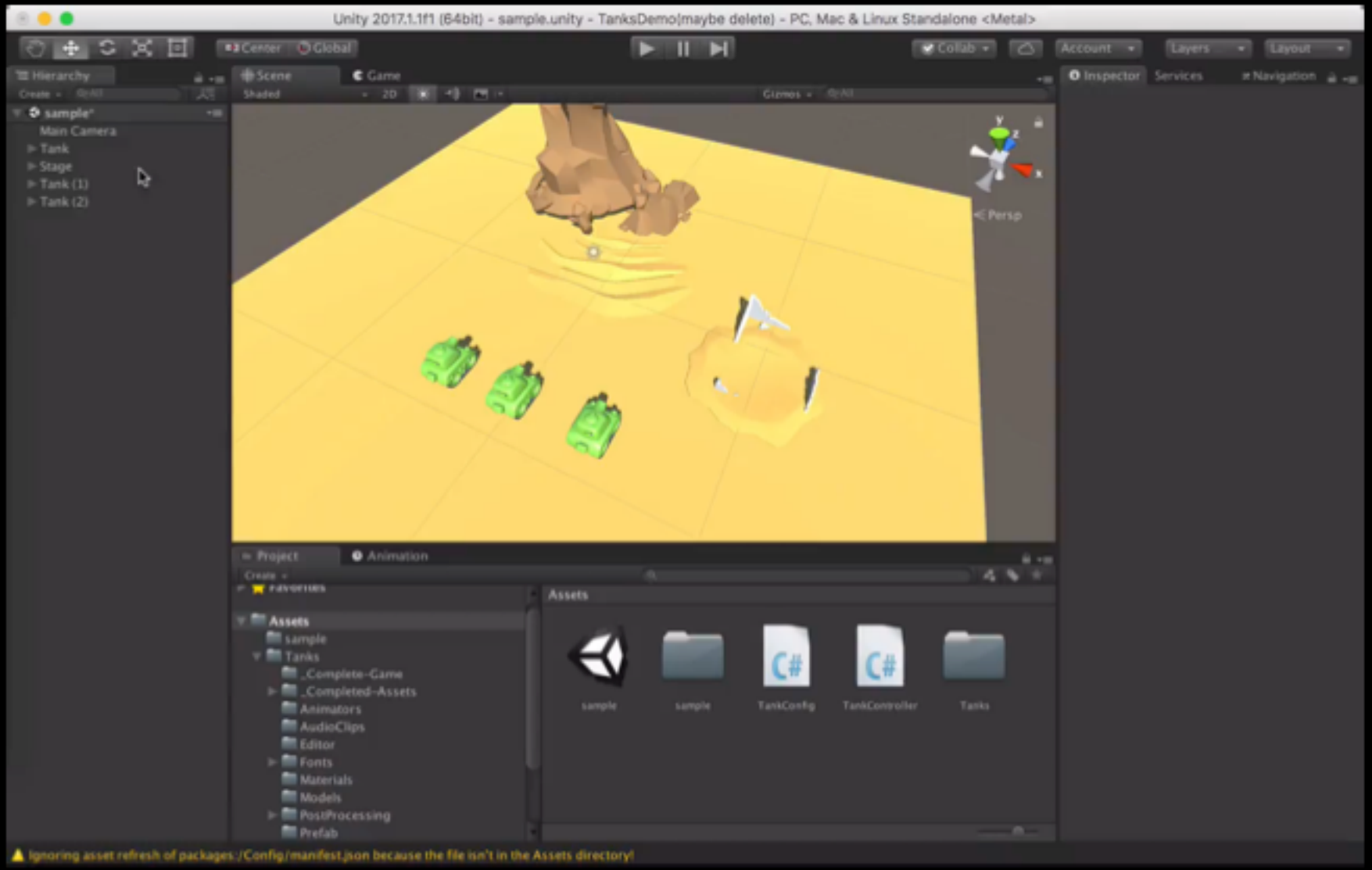
unity

```
public class TankController : MonoBehaviour
{
  [SerializeField] TankConfig config;

  public void Update ()
  {
    var v = Input.GetAxis ("Vertical1");
    var x = Input.GetAxis ("Horizontal1");
    if (v != 0)
      GetComponent<NavMeshAgent> ().Move (transform.forward * (v * config.speed));
    if (x != 0)
      transform.localRotation *= Quaternion.AngleAxis (x * config.rotate, Vector3.up);
  }
}
```

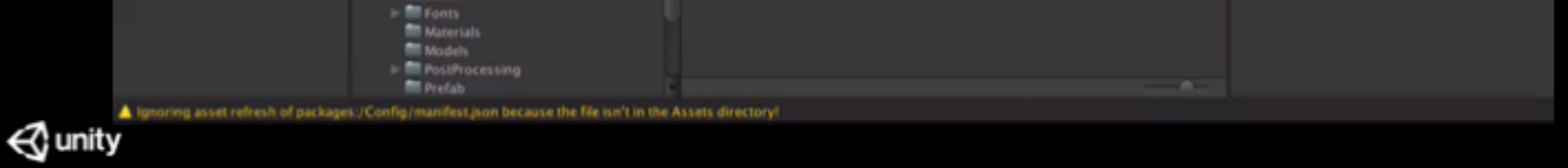
unity

```
    }
}
```

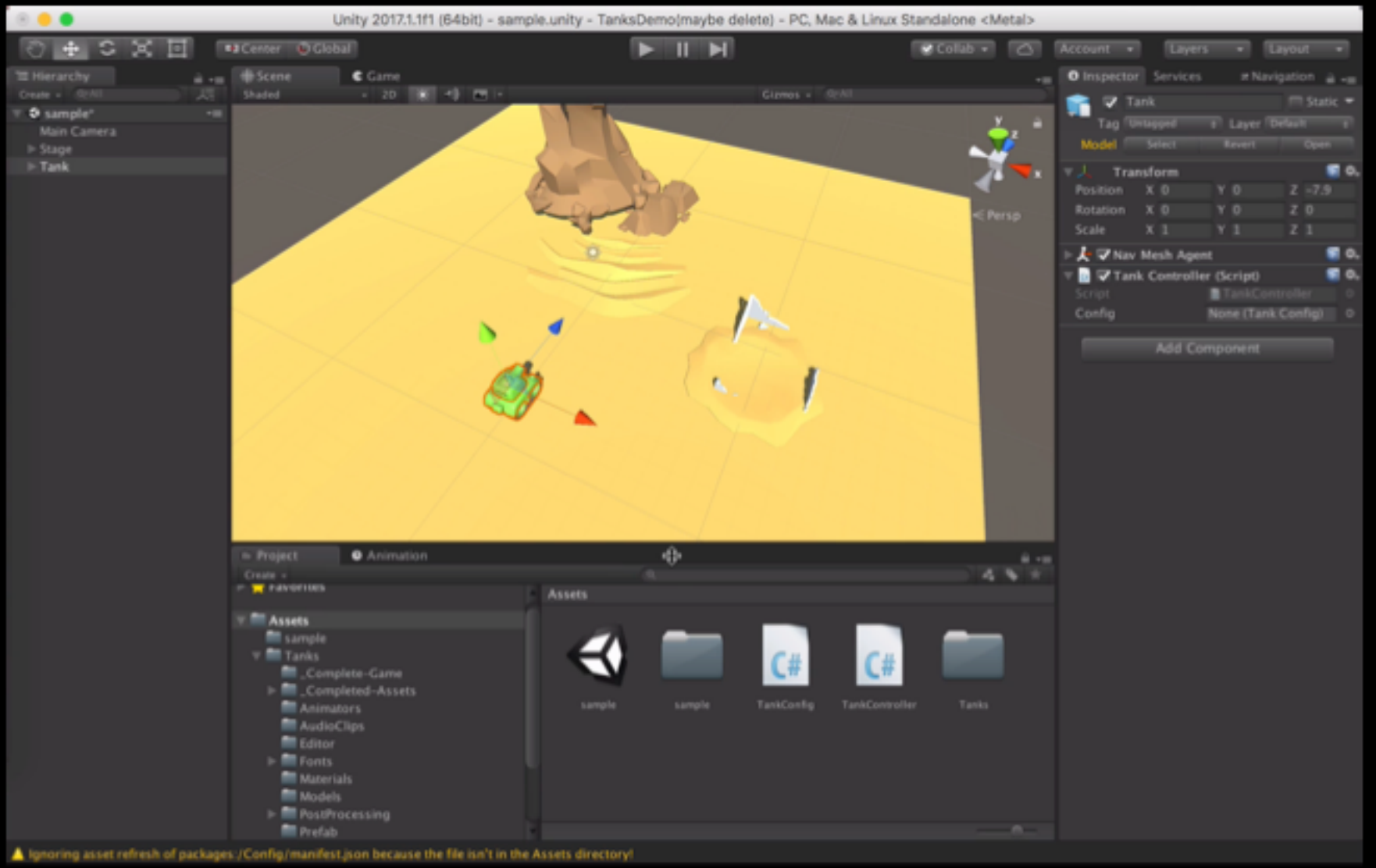
unity


Unity 2017.1.1f1 (64bit) - sample.unity - TanksDemo(maybe delete) - PC, Mac & Linux Standalone <Metal>
Center
Global
Collab
Account
Layers
Layout
Hierarchy
Create
sample*
Main Camera
Tank
Stage
Tank (1)
Tank (2)
Scene
Game
Shaded
2D
Gizmos
Persp
Inspector
Services
Navigation
Project
Animation
Create
Favorites
Assets
sample
Tanks
_Complete-Game
_Completed-Assets
Animators
AudioClips
Editor
Fonts
Materials
Models
PostProcessing
Prefab
Assets
sample
sample
TankConfig
TankController
Tanks
Ignoring asset refresh of packages :/Config/manifest.json because the file isn't in the Assets directory!

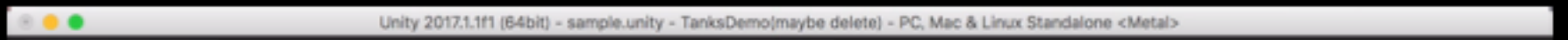
unity

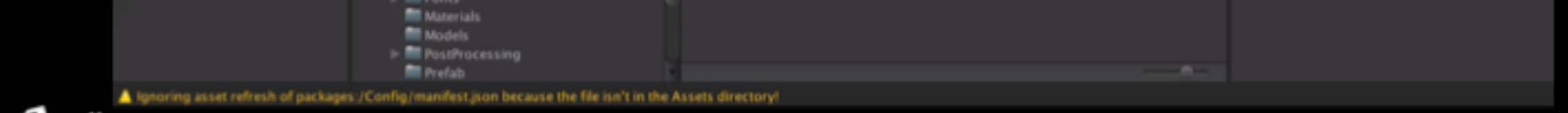

Fonts
Materials
Models
PostProcessing
Prefab
Ignoring asset refresh of packages:/Config/manifest.json because the file isn't in the Assets directory!

unity

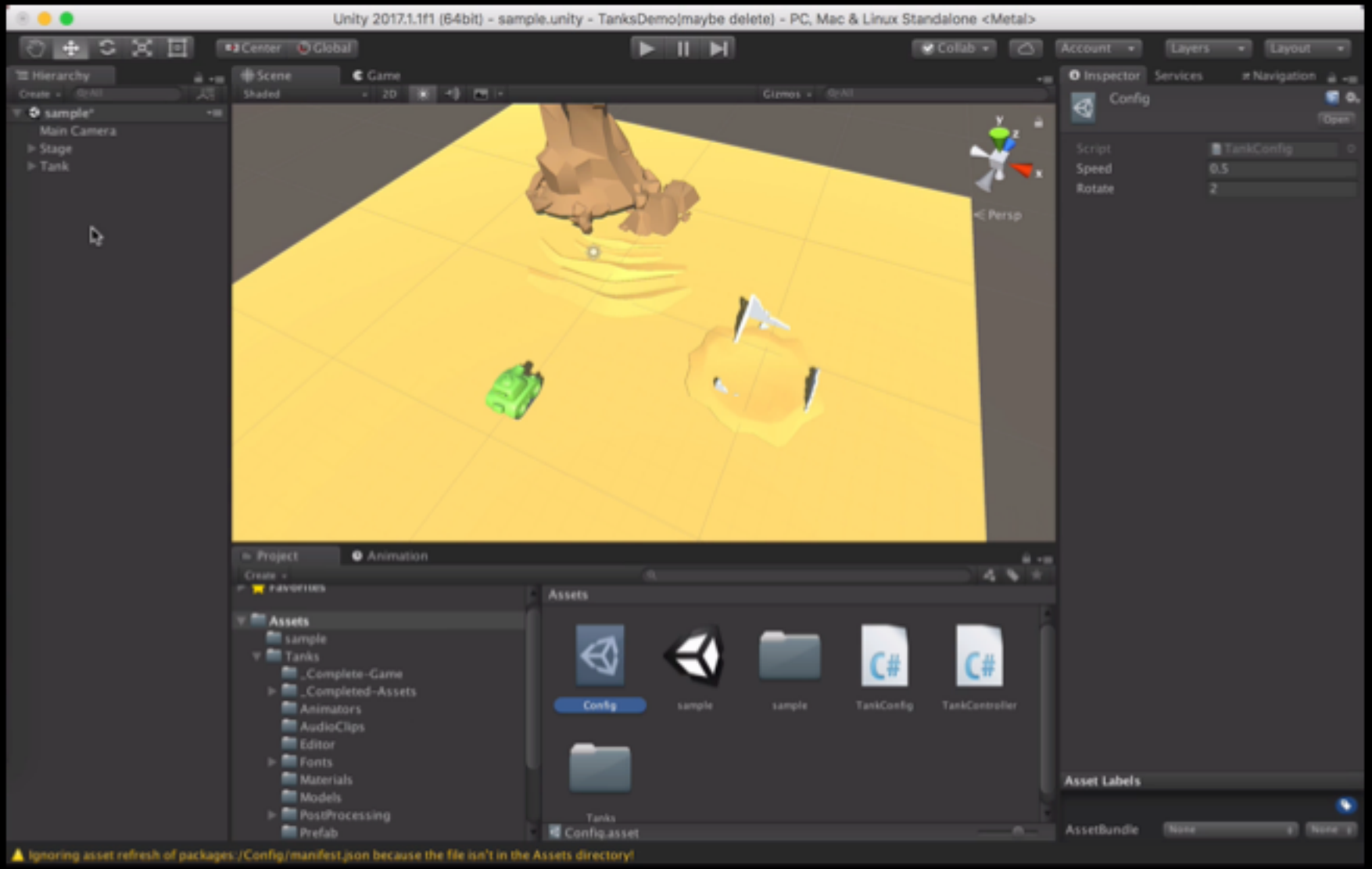

Unity 2017.1.1f1 (64bit) - sample.unity - TanksDemo(maybe delete) - PC, Mac & Linux Standalone <Metal>
Center
Global
Collab
Account
Layers
Layout
Hierarchy
Scene
Game
Inspector
Services
Navigation
sample*
Main Camera
Stage
Tank
Persp
Tank
Static
Transform
Position
Rotation
Scale
Nav Mesh Agent
Tank Controller (Script)
Script
Config
None (Tank Config)
Add Component
Project
Animation
Assets
sample
Tanks
_Complete-Game
_Completed-Assets
Animators
AudioClips
Editor
Fonts
Materials
Models
PostProcessing
Prefab
TankConfig
TankController
Ignoring asset refresh of packages:/Config/manifest.json because the file isn't in the Assets directory!

unity

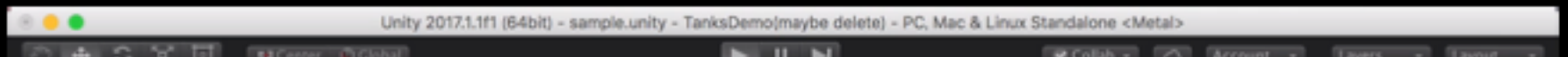

Unity 2017.1.1f1 (64bit) - sample.unity - TanksDemo(maybe delete) - PC, Mac & Linux Standalone <Metal>

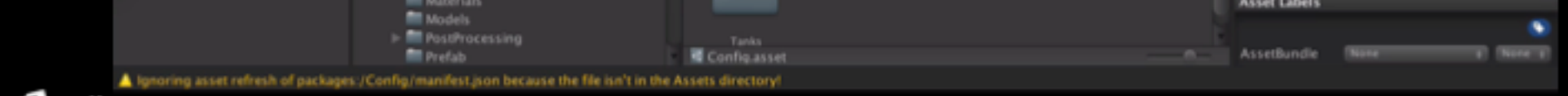

Materials
Models
PostProcessing
Prefab
Ignoring asset refresh of packages:/Config/manifest.json because the file isn't in the Assets directory!

unity

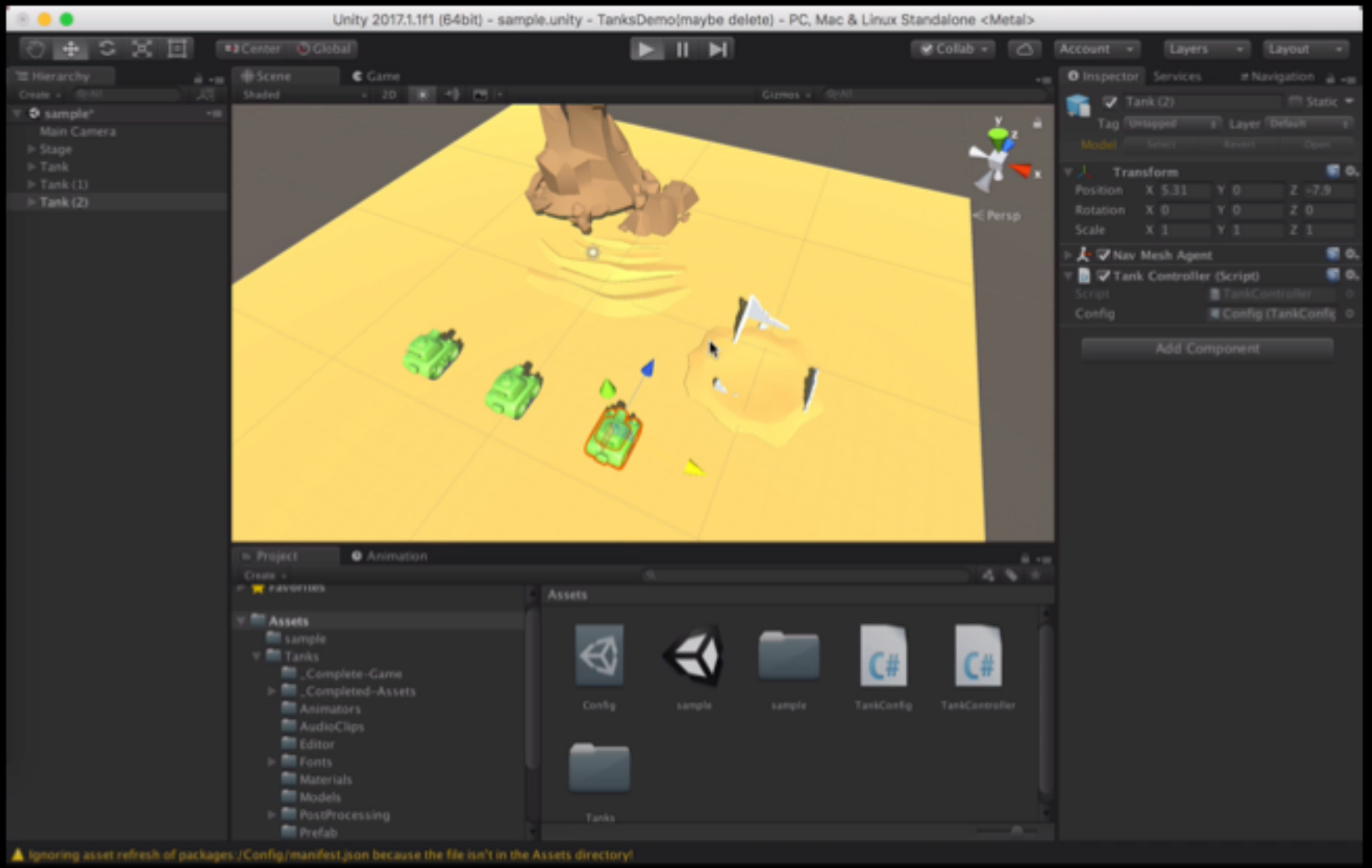

Unity 2017.1.1f1 (64bit) - sample.unity - TanksDemo(maybe delete) - PC, Mac & Linux Standalone <Metal>
Center
Global
Collab
Account
Layers
Layout
Hierarchy
Create
sample*
Main Camera
Stage
Tank
Scene
Game
Shaded
2D
Gizmos
Persp
Inspector
Services
Navigation
Config
Open
Script
TankConfig
Speed
0.5
Rotate
2
Project
Animation
Create
Favorites
Assets
sample
Tanks
_Complete-Game
_Completed-Assets
Animators
AudioClips
Editor
Fonts
Materials
Models
PostProcessing
Prefab
Assets
Config
sample
sample
TankConfig
TankController
Tanks
Config.asset
Asset Labels
AssetBundle
None
None
Ignoring asset refresh of packages:/Config/manifest.json because the file isn't in the Assets directory!

unity


Unity 2017.1.1f1 (64bit) - sample.unity - TanksDemo(maybe delete) - PC, Mac & Linux Standalone <Metal>


Models
PostProcessing
Prefab
Tanks
Config.asset
AssetBundle
None
None
Ignoring asset refresh of packages /Config/manifest.json because the file isn't in the Assets directory!

unity

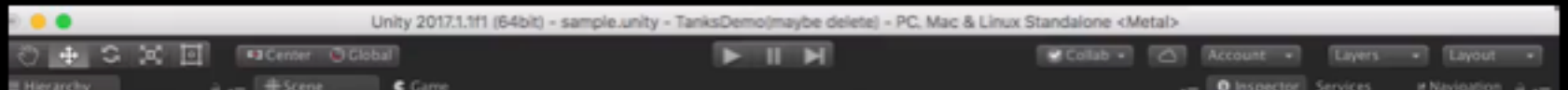

Unity 2017.1.1f1 (64bit) - sample.unity - TanksDemo(maybe delete) - PC, Mac & Linux Standalone <Metal>
Center
Global
Collab
Account
Layers
Layout
Hierarchy
Scene
Game
Inspector
Services
Navigation
sample*
Main Camera
Stage
Tank
Tank (1)
Tank (2)
Persp
Tank (2)
Static
Transform
Position
Rotation
Scale
Nav Mesh Agent
Tank Controller (Script)
Add Component
Project
Animation
Assets
sample
Tanks
_Complete-Game
_Completed-Assets
Animators
AudioClips
Editor
Fonts
Materials
Models
PostProcessing
Prefab
Config
sample
sample
TankConfig
TankController
Tanks
Ignoring asset refresh of packages /Config/manifest.json because the file isn't in the Assets directory!

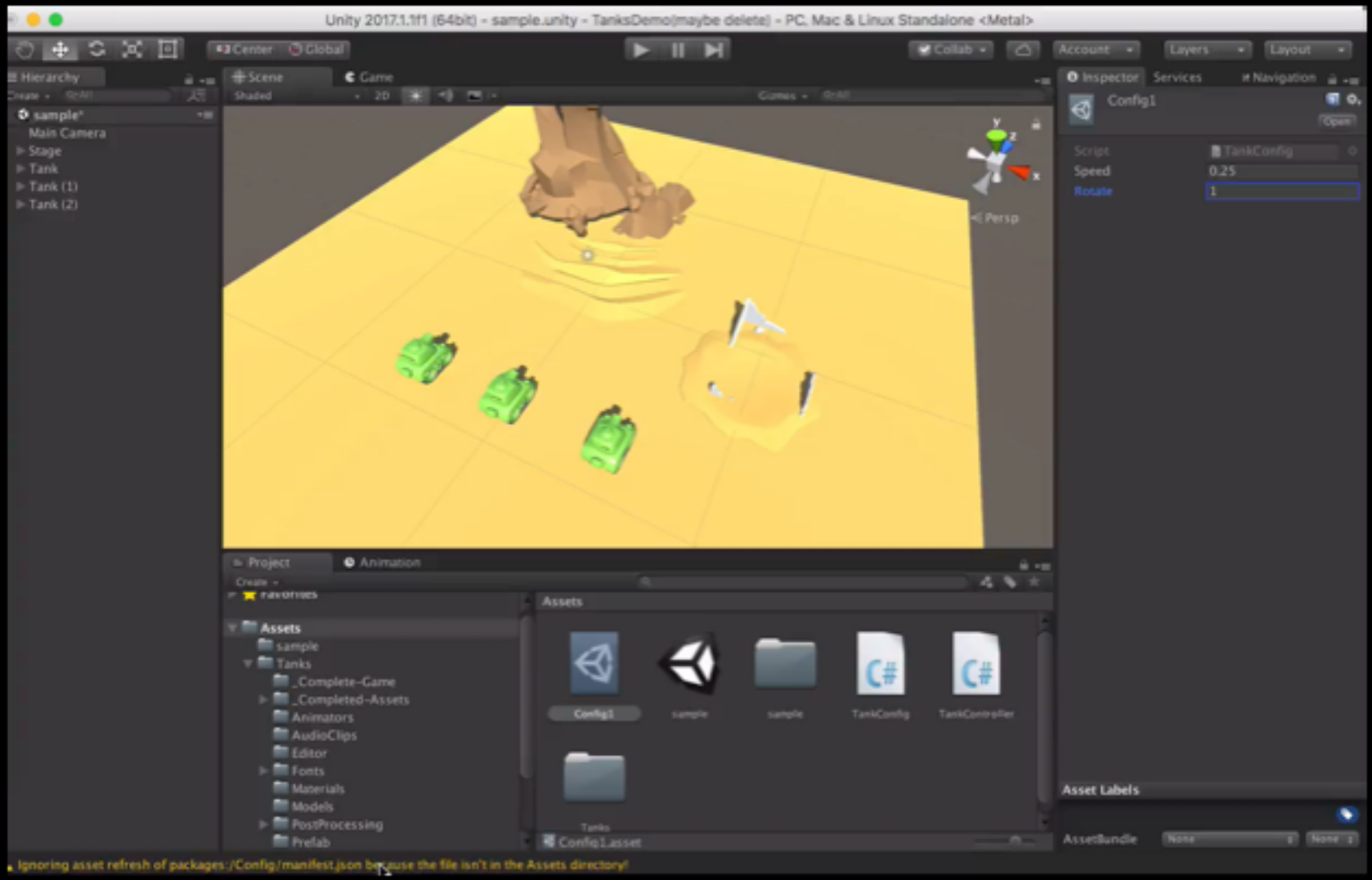
unity

unity

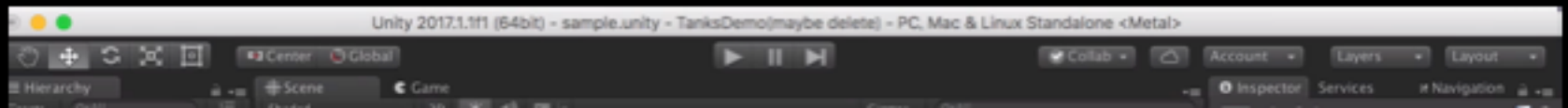
NPC Config 1
弱い

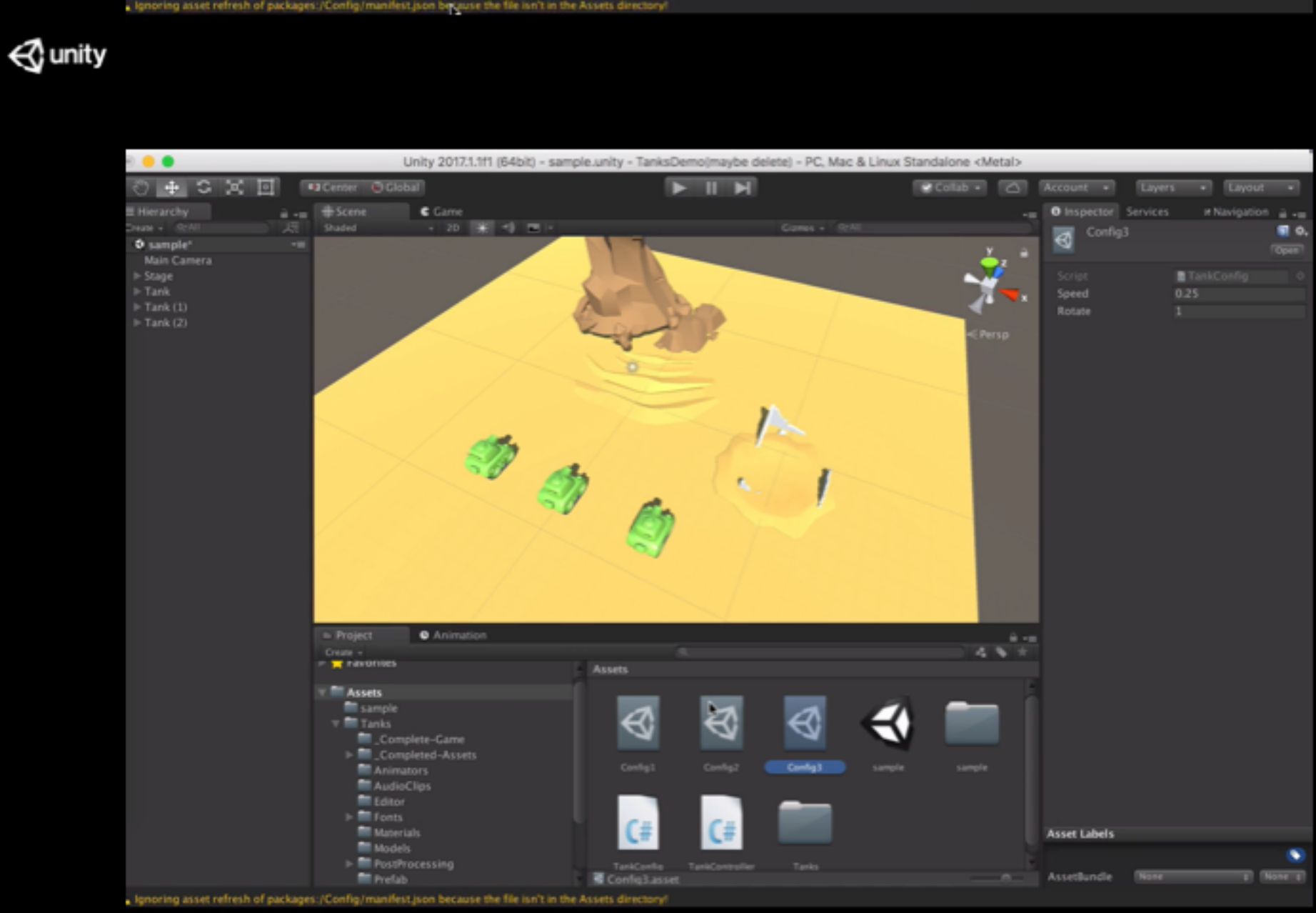
NPC Config 2
強い

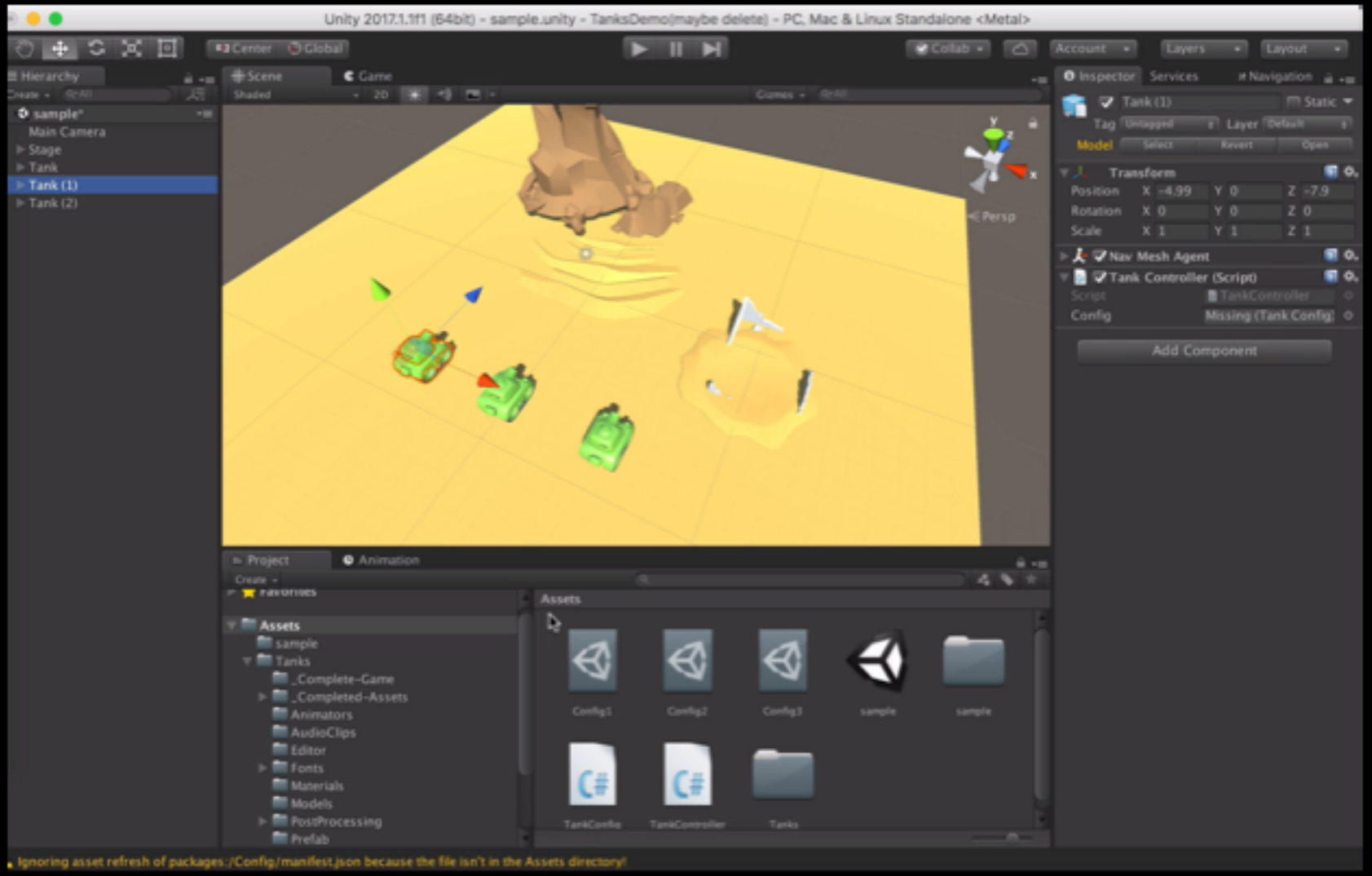


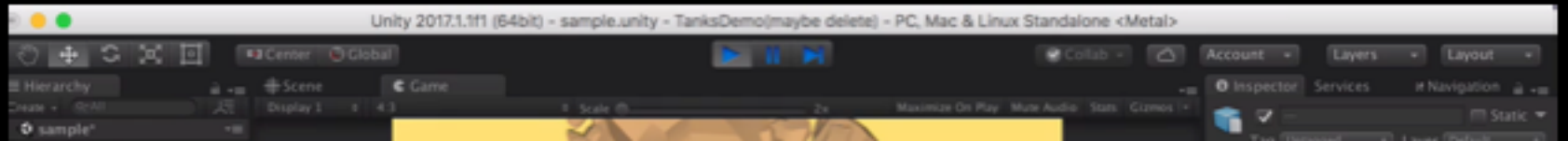

Unity 2017.1.1f1 (64bit) - sample.unity - TanksDemo(maybe delete) - PC, Mac & Linux Standalone <Metal>
Hierarchy
sample*
Main Camera
Stage
Tank
Tank (1)
Tank (2)
Scene
Game
Inspector
Services
Navigation
Config1
Script
TankConfig
Speed
0.25
Rotate
1
Persp
Project
Animation
Assets
sample
Tanks
_Complete-Game
_Completed-Assets
Animators
AudioClips
Editor
Fonts
Materials
Models
PostProcessing
Prefab
Config1
sample
sample
TankConfig
TankController
Tanks
Config1.asset
Asset Labels
AssetBundle
None
None
Ignoring asset refresh of packages ./Config/manifest.json because the file isn't in the Assets directory!

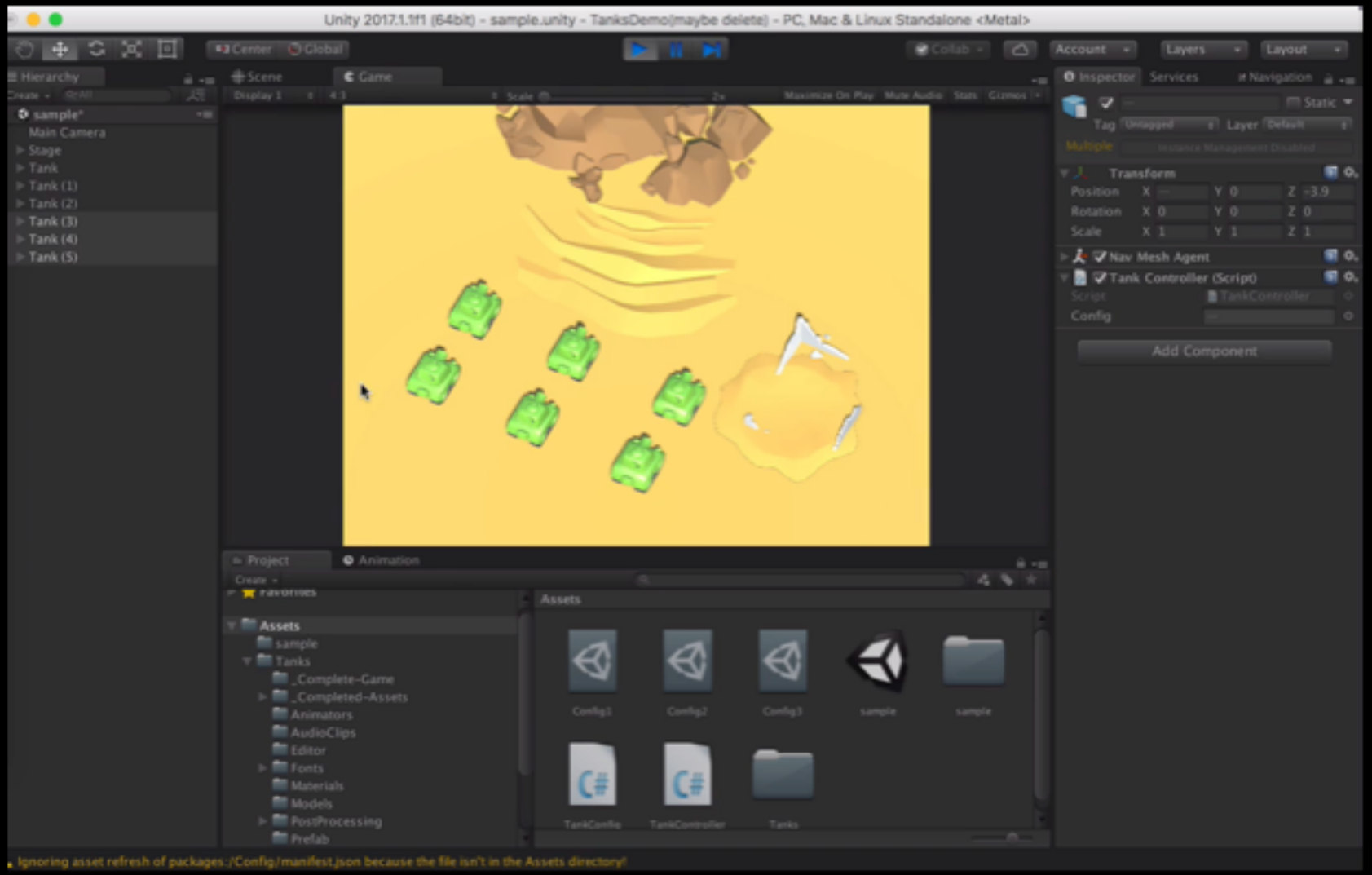
unity

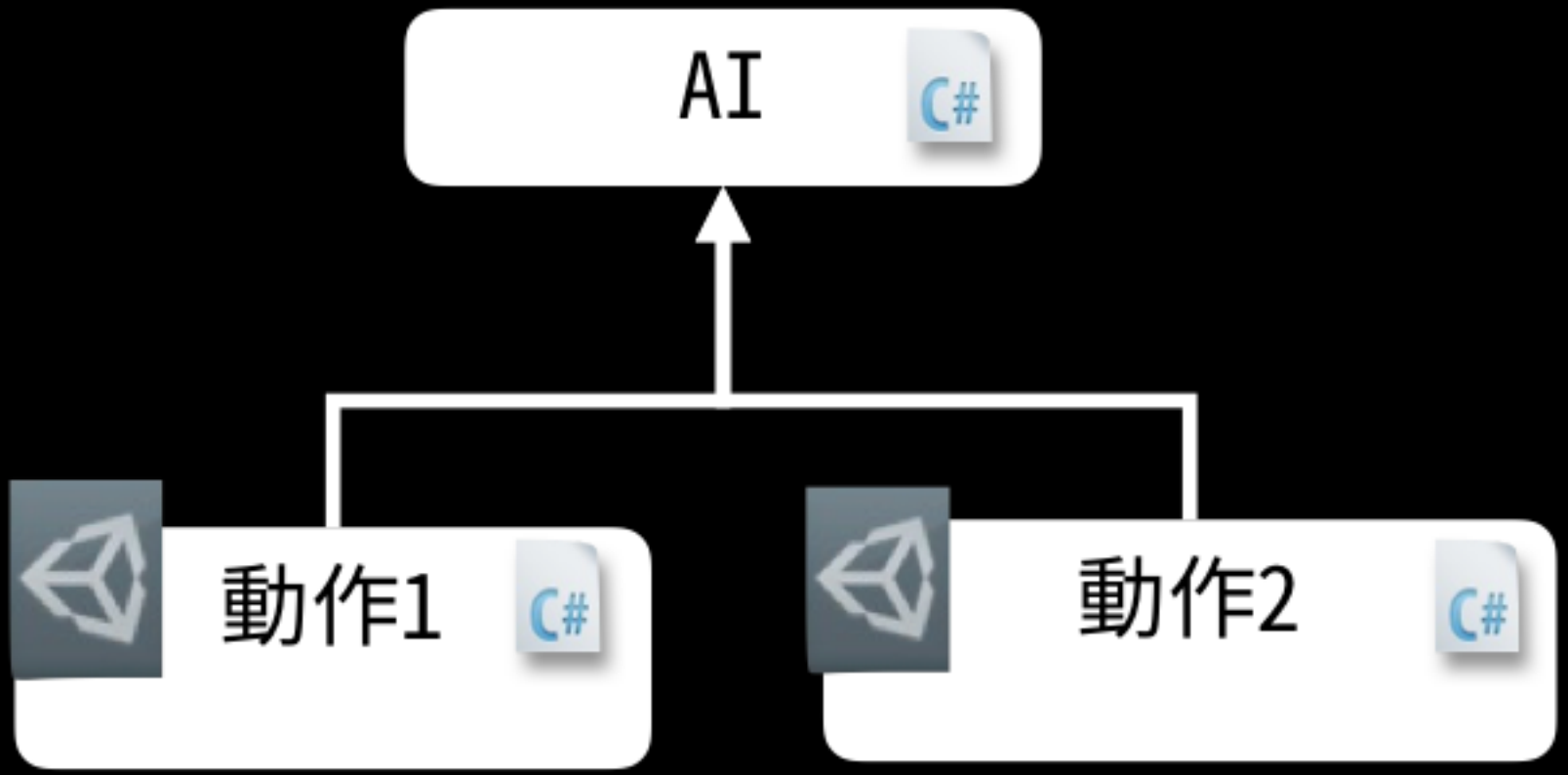

Unity 2017.1.1f1 (64bit) - sample.unity - TanksDemo(maybe delete) - PC, Mac & Linux Standalone <Metal>


Ignoring asset refresh of packages: /Config/manifest.json because the file isn't in the Assets directory!
unity
Unity 2017.1.1f1 (64bit) - sample.unity - TanksDemo(maybe delete) - PC, Mac & Linux Standalone <Metal>
Center
Global
Collab
Account
Layers
Layout
Hierarchy
Scene
Game
Inspector
Services
Navigation
sample*
Main Camera
Stage
Tank
Tank (1)
Tank (2)
Config3
Open
Script
TankConfig
Speed
0.25
Rotate
1
Persp
Project
Animation
Assets
sample
Tanks
_Complete-Game
_Completed-Assets
Animators
AudioClips
Editor
Fonts
Materials
Models
PostProcessing
Prefab
Config1
Config2
Config3
sample
sample
TankConfig
TankController
Tanks
Config3.asset
Asset Labels
AssetBundle
None
None
Ignoring asset refresh of packages: /Config/manifest.json because the file isn't in the Assets directory!
unity
Unity 2017.1.1f1 (64bit) - sample.unity - TanksDemo(maybe delete) - PC, Mac & Linux Standalone <Metal>
Center
Global
Collab
Account
Layers
Layout
Hierarchy
Scene
Game
Inspector
Services
Navigation
Tank (1)
Static


Unity 2017.1.1f1 (64bit) - sample.unity - TanksDemo(maybe delete) - PC, Mac & Linux Standalone <Metal>
Hierarchy
Scene
Game
Inspector
Services
Navigation
sample*
Main Camera
Stage
Tank
Tank (1)
Tank (2)
Persp
Transform
Position
Rotation
Scale
Nav Mesh Agent
Tank Controller (Script)
Config
Missing (Tank Config)
Add Component
Project
Animation
Assets
sample
Tanks
_Complete-Game
_Completed-Assets
Animators
AudioClips
Editor
Fonts
Materials
Models
PostProcessing
Prefab
Config1
Config2
Config3
TankConfig
TankController
Ignoring asset refresh of packages:/Config/manifest.json because the file isn't in the Assets directory!


Unity 2017.1.1f1 (64bit) - sample.unity - TanksDemo(maybe delete) - PC, Mac & Linux Standalone <Metal>


Unity 2017.1.1f1 (64bit) - sample.unity - TanksDemo(maybe delete) - PC, Mac & Linux Standalone <Metal>
Hierarchy
sample*
Main Camera
Stage
Tank
Tank (1)
Tank (2)
Tank (3)
Tank (4)
Tank (5)
Scene
Game
Inspector
Services
Navigation
Transform
Position
Rotation
Scale
Nav Mesh Agent
Tank Controller (Script)
Add Component
Project
Animation
Assets
sample
Tanks
_Complete-Game
_Completed-Assets
Animators
AudioClips
Editor
Fonts
Materials
Models
PostProcessing
Prefab
Config1
Config2
Config3
sample
sample
TankConfig
TankController
Tanks
Ignoring asset refresh of packages ./Config/manifest.json because the file isn't in the Assets directory!


AI
動作1
動作2

```
public class TankController : MonoBehaviour
{
  [SerializeField] TankAIBase ai;

  public void Update ()
  {
    ai.Do(gameObject);
  }
}
```

```
public abstract class TankAIBase : ScriptableObject
{
  public virtual void Do (GameObject obj) { }
}
```

```
[CreateAssetMenu]
public class TanksStand : TankAIBase{}
```

```
[CreateAssetMenu]
public class TankRotate : TankAIBase
{
  public float rotate = 2;

  public override void Do (GameObject obj)
  {
    obj.transform.localRotation *= Quaternion.AngleAxis (rotate, Vector3.up);
  }
}
```

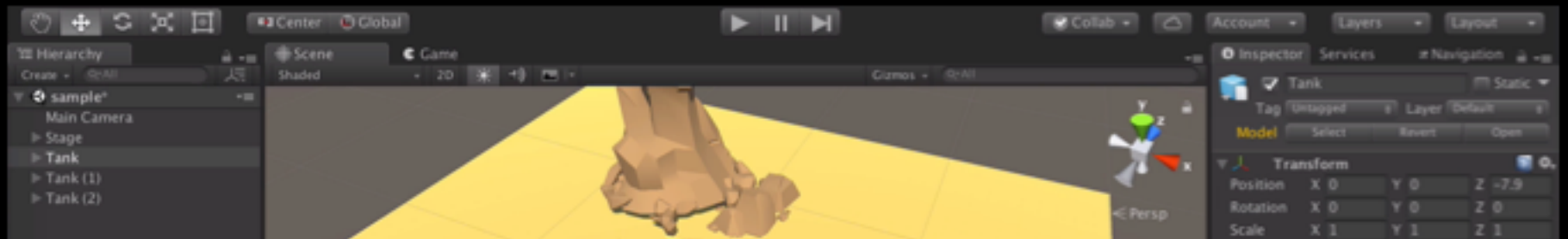
unity

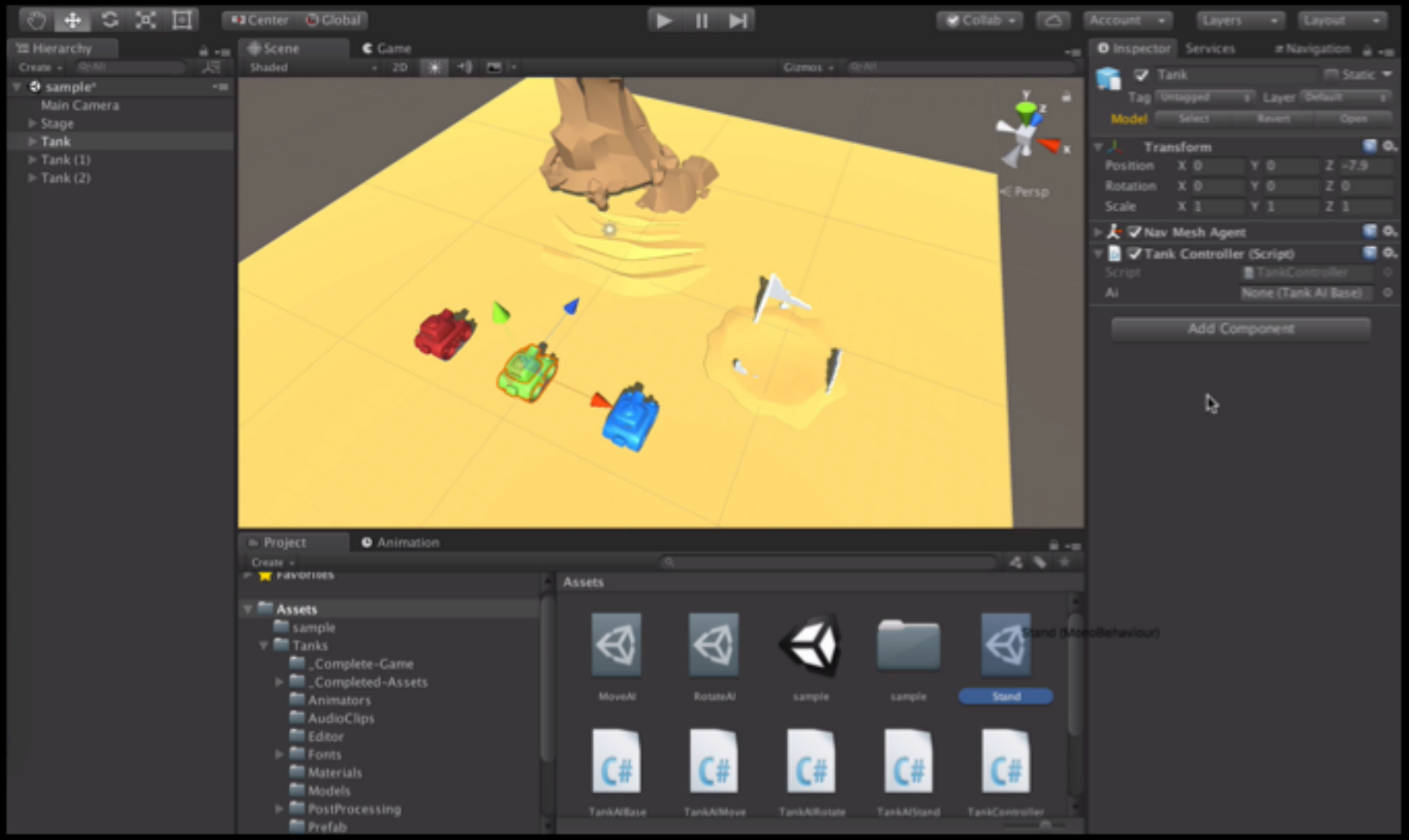
```
[CreateAssetMenu]
public class TankMove : TankAIBase
{
  public float speed = 0.5f;

  public override void Do (GameObject obj)
  {
    obj.GetComponent<NavMeshAgent>().Move(obj.transform.forward * speed);
  }
}
```

unity

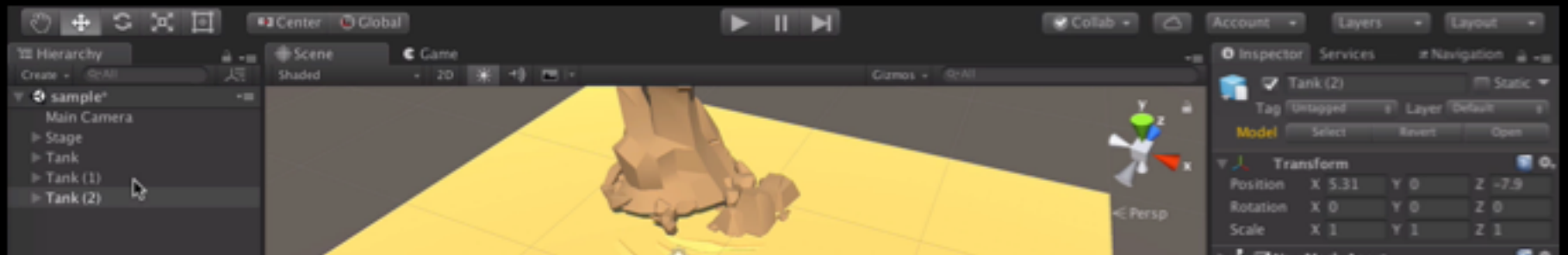

Center
Global
Collab
Account
Layers
Layout
Hierarchy
Create
sample*
Main Camera
Stage
Tank
Tank (1)
Tank (2)
Scene
Game
Shaded
2D
Gizmos
Persp
Inspector
Services
Navigation
Tank (2)
Static
Tag
Untagged
Layer
Default
Model
Select
Revert
Open
Transform
Position
X 5.31
Y 0
Z -7.9
Rotation
X 0
Y 0
Z 0
Scale
X 1
Y 1
Z 1
Nav Mesh Agent
Tank Controller (Script)
Script
TankController
Ai
None (Tank AI Base)
Add Component
Project
Animation
Create
Favorites
Assets
sample
Tanks
_Complete-Game
_Completed-Assets
Animators
AudioClips
Editor
Fonts
Materials
Models
PostProcessing
Prefabs
Assets
sample
sample
TankAIBase
TankAIMove
TankAIRotate
TankAIStand
TankController
Tanks

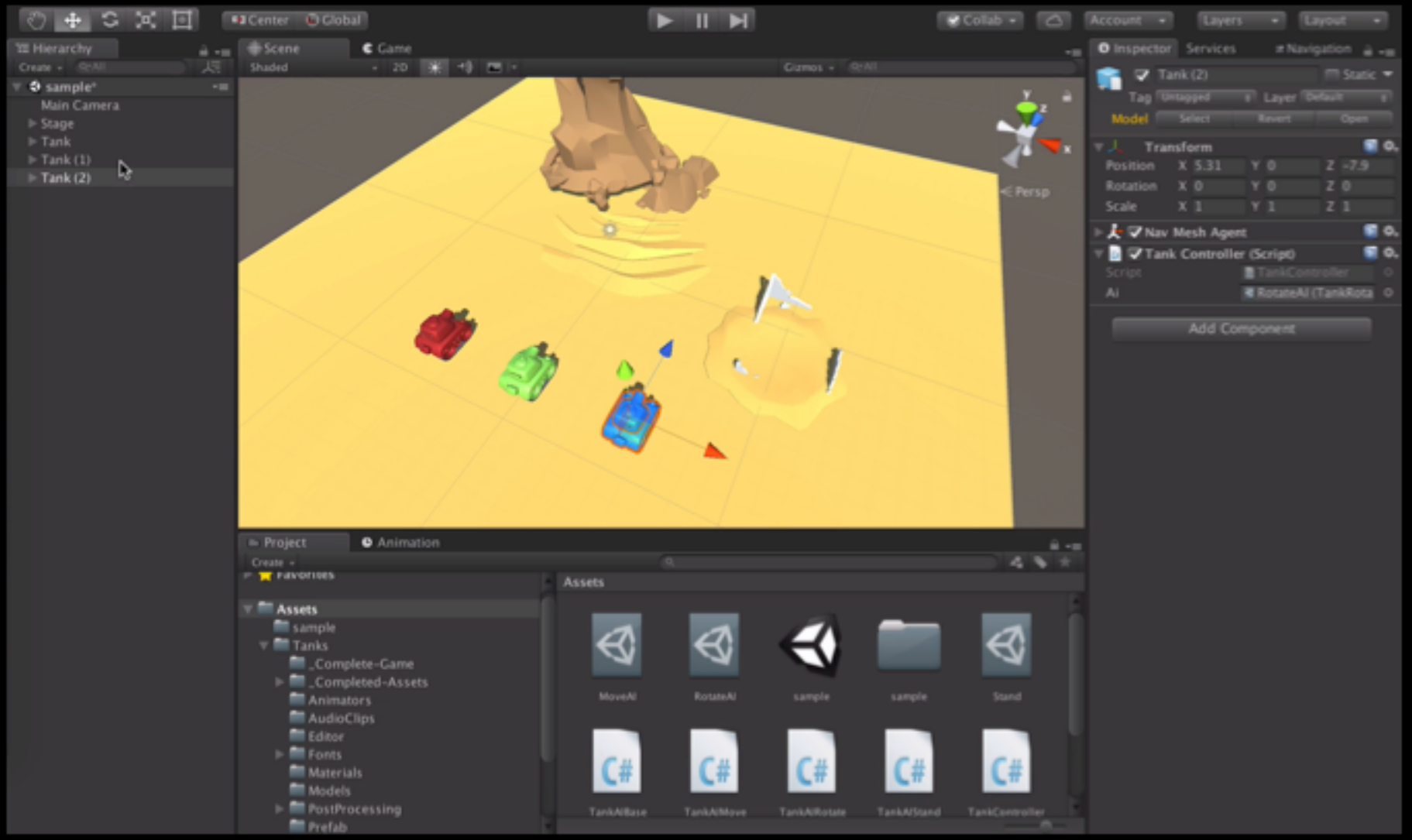
unity


Center
Global
Collab
Account
Layers
Layout
Hierarchy
Create
sample*
Main Camera
Stage
Tank
Tank (1)
Tank (2)
Scene
Game
Shaded
2D
Gizmos
Persp
Inspector
Services
Navigation
Tank
Static
Tag
Untagged
Layer
Default
Model
Select
Revert
Open
Transform
Position
X 0
Y 0
Z -7.9
Rotation
X 0
Y 0
Z 0
Scale
X 1
Y 1
Z 1

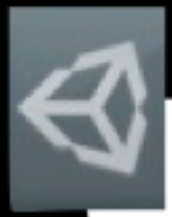

Center
Global
Collab
Account
Layers
Layout
Hierarchy
Create
sample*
Main Camera
Stage
Tank
Tank (1)
Tank (2)
Scene
Game
Shaded
2D
Gizmos
Persp
Inspector
Services
Navigation
Tank
Static
Tag
Untagged
Layer
Default
Model
Select
Revert
Open
Transform
Position
Rotation
Scale
Nav Mesh Agent
Tank Controller (Script)
Script
TankController
Ai
None (Tank AI Base)
Add Component
Project
Animation
Create
Favorites
Assets
sample
Tanks
_Complete-Game
_Completed-Assets
Animators
AudioClips
Editor
Fonts
Materials
Models
PostProcessing
Prefab
Assets
MoveAI
RotateAI
sample
sample
Stand
Stand (MonoBehaviour)
TankAIBase
TankAIMove
TankAIRotate
TankAIStand
TankController

unity

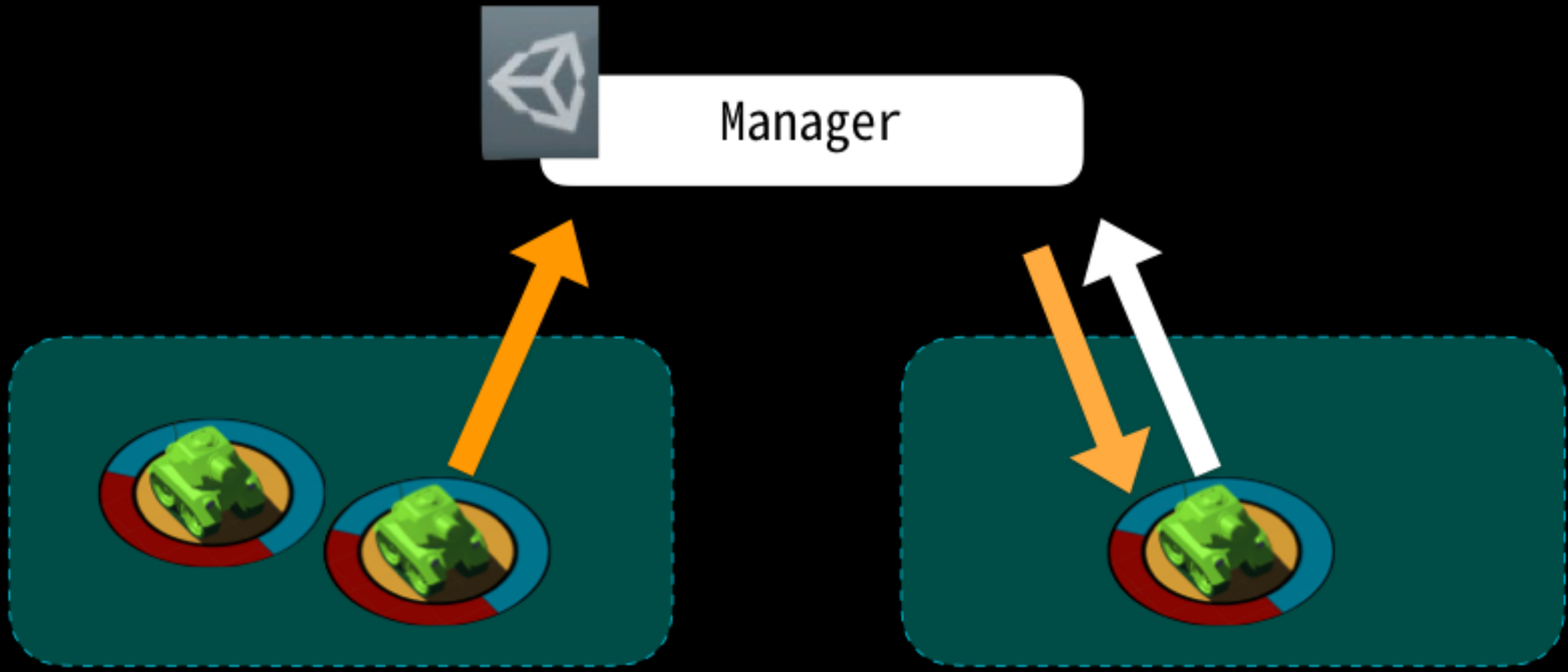

Center
Global
Collab
Account
Layers
Layout
Hierarchy
Create
sample*
Main Camera
Stage
Tank
Tank (1)
Tank (2)
Scene
Game
Shaded
2D
Gizmos
Persp
Inspector
Services
Navigation
Tank (2)
Static
Tag
Untagged
Layer
Default
Model
Select
Revert
Open
Transform
Position
Rotation
Scale



unity



Manager




```
public class TankController : MonoBehaviour
{
   public void Move(float accel)
```

```
public class TankController : MonoBehaviour
{
   public void Move(float accel)
   {
      GetComponent<NavMeshAgent>().Move(transform.forward * (accel * 0.2f));
   }
}
```

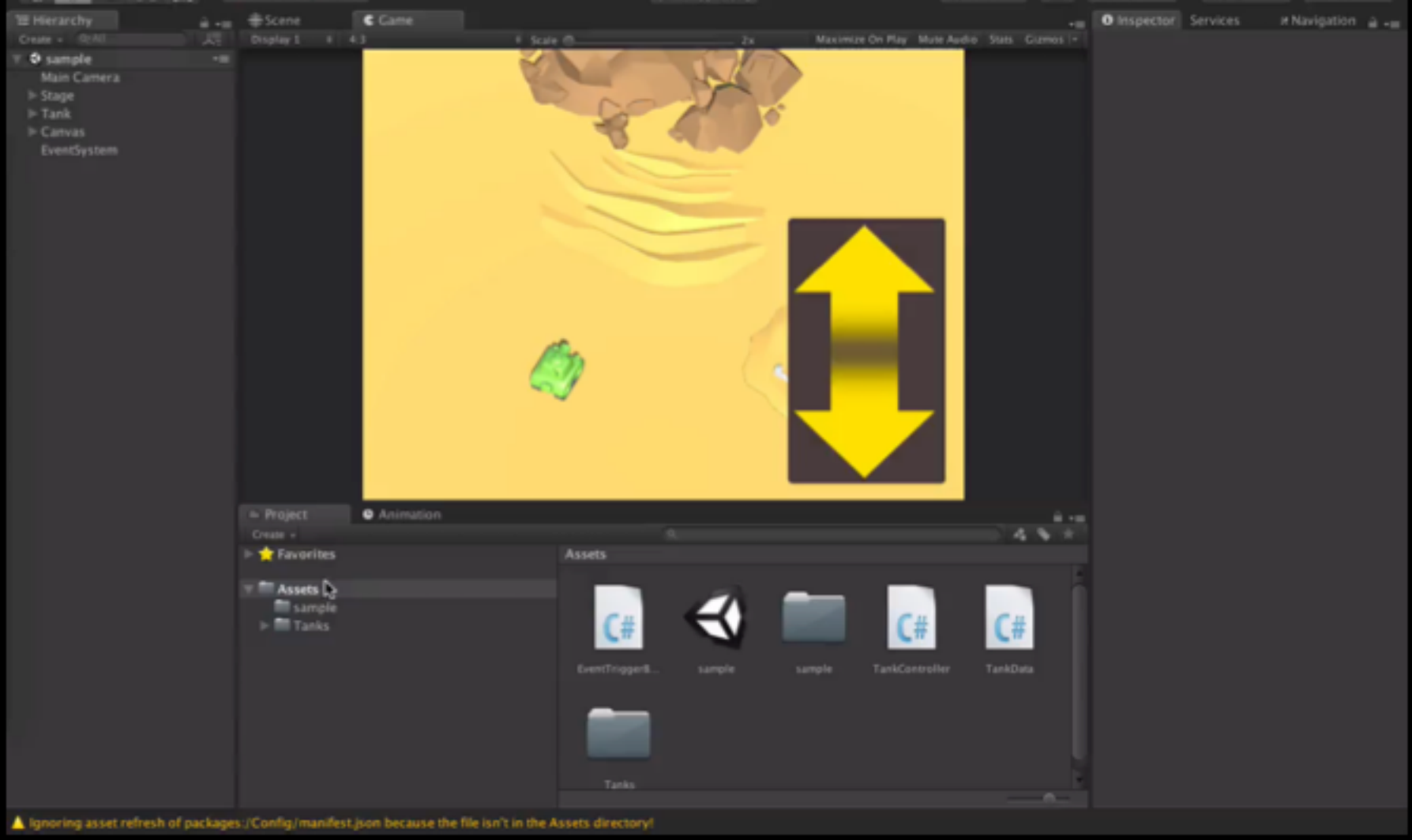
```
[CreateAssetMenu]
public class TankData : ScriptableObject
{
   public TankController tankController{get; set;}

   public void Up ()   { tankController.Move(5); }
   public void Down () { tankController.Move(-5); }
}
```

unity

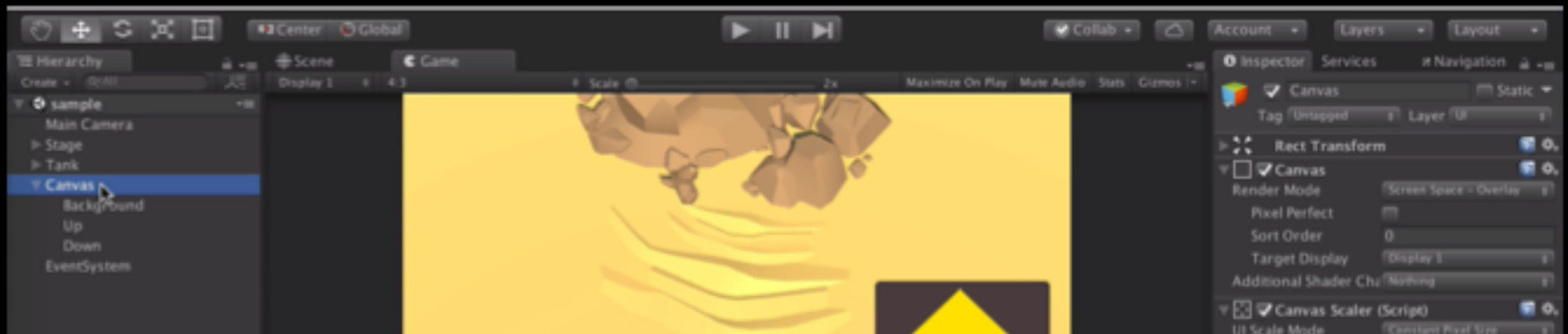
```
[DefaultExecutionOrder (-100)]
public class EventTriggerBehaviour : MonoBehaviour
{
```

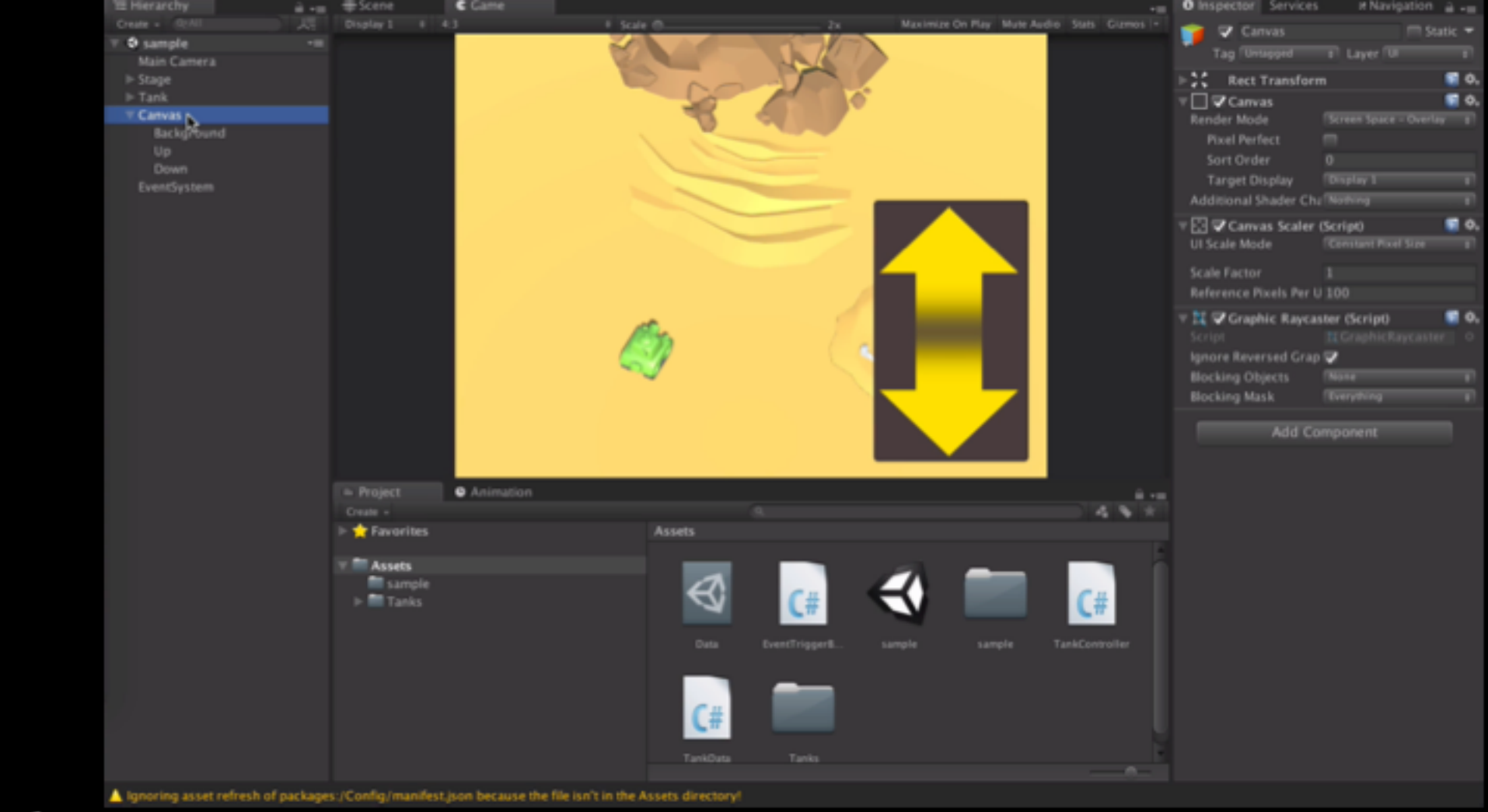
```
[DefaultExecutionOrder (-100)]
public class EventTriggerBehaviour : MonoBehaviour
{
   [SerializeField]
   UnityEvent onAwake = new UnityEvent (), onDestroy = new UnityEvent ();

   void Awake ()
   {
      onAwake.Invoke ();
   }

   void OnDestroy ()
   {
      onDestroy.Invoke ();
   }
}
```

unity


Hierarchy
Scene
Game
Inspector
Services
Navigation
sample
Main Camera
Stage
Tank
Canvas
EventSystem
Project
Animation
Favorites
Assets
sample
Tanks
EventTriggerE...
sample
sample
TankController
TankData
Tanks
Ignoring asset refresh of packages :/Config/manifest.json because the file isn't in the Assets directory!

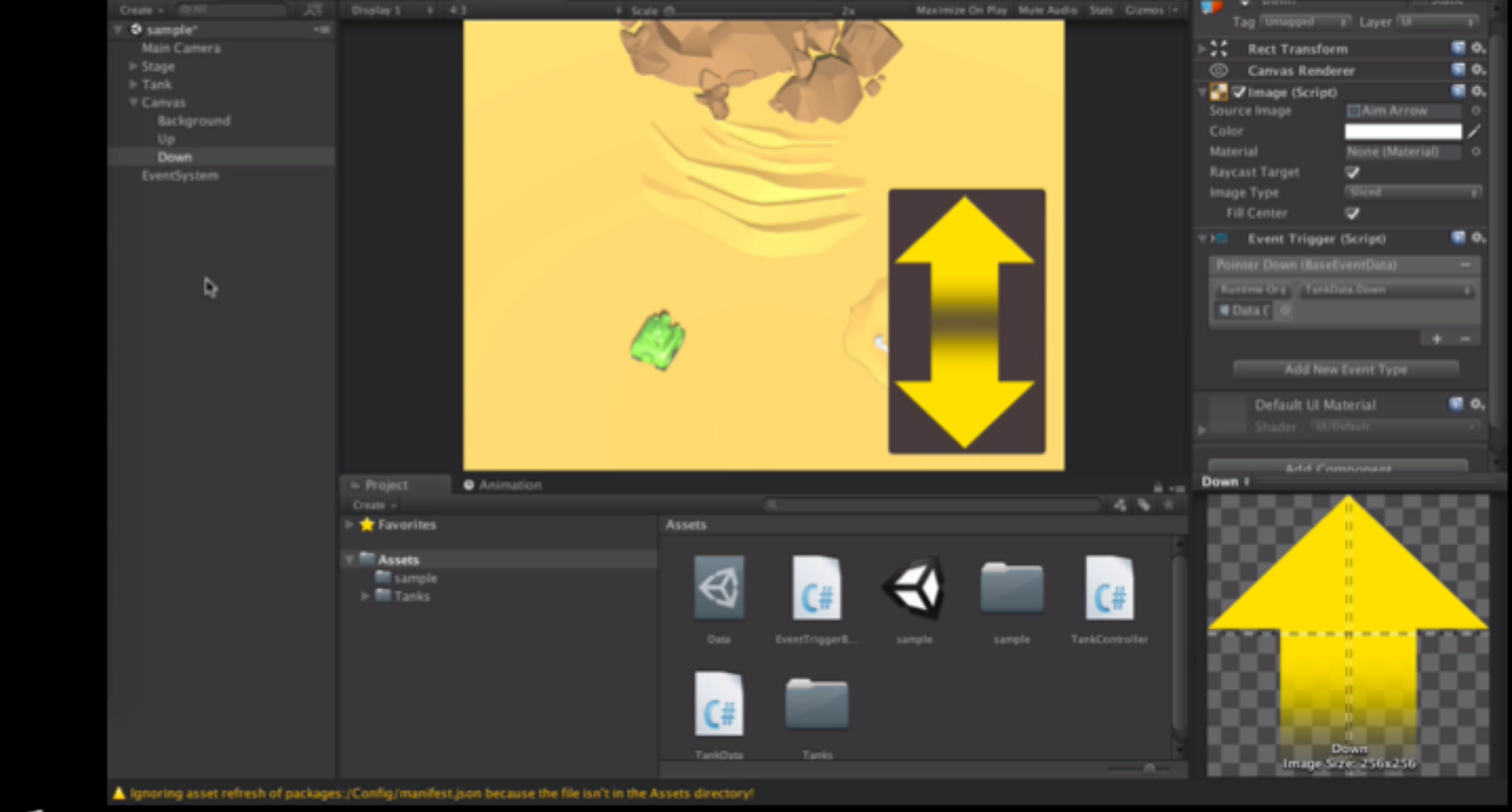
unity


Center
Global
Collab
Account
Layers
Layout
Hierarchy
Scene
Game
Inspector
Services
Navigation
sample
Main Camera
Stage
Tank
Canvas
Background
Up
Down
EventSystem
Canvas
Static
Tag
Layer
Rect Transform
Canvas
Render Mode
Screen Space - Overlay
Pixel Perfect
Sort Order
Target Display
Canvas Scaler (Script)


Hierarchy
sample
Main Camera
Stage
Tank
Canvas
Background
Up
Down
EventSystem
Scene
Game
Inspector
Services
Navigation
Canvas
Rect Transform
Render Mode
Pixel Perfect
Sort Order
Target Display
Canvas Scaler (Script)
UI Scale Mode
Scale Factor
Reference Pixels Per U 100
Graphic Raycaster (Script)
Ignore Reversed Grap
Blocking Objects
Blocking Mask
Add Component
Project
Animation
Favorites
Assets
sample
Tanks
TankController
TankData
Ignoring asset refresh of packages:/Config/manifest.json because the file isn't in the Assets directory!

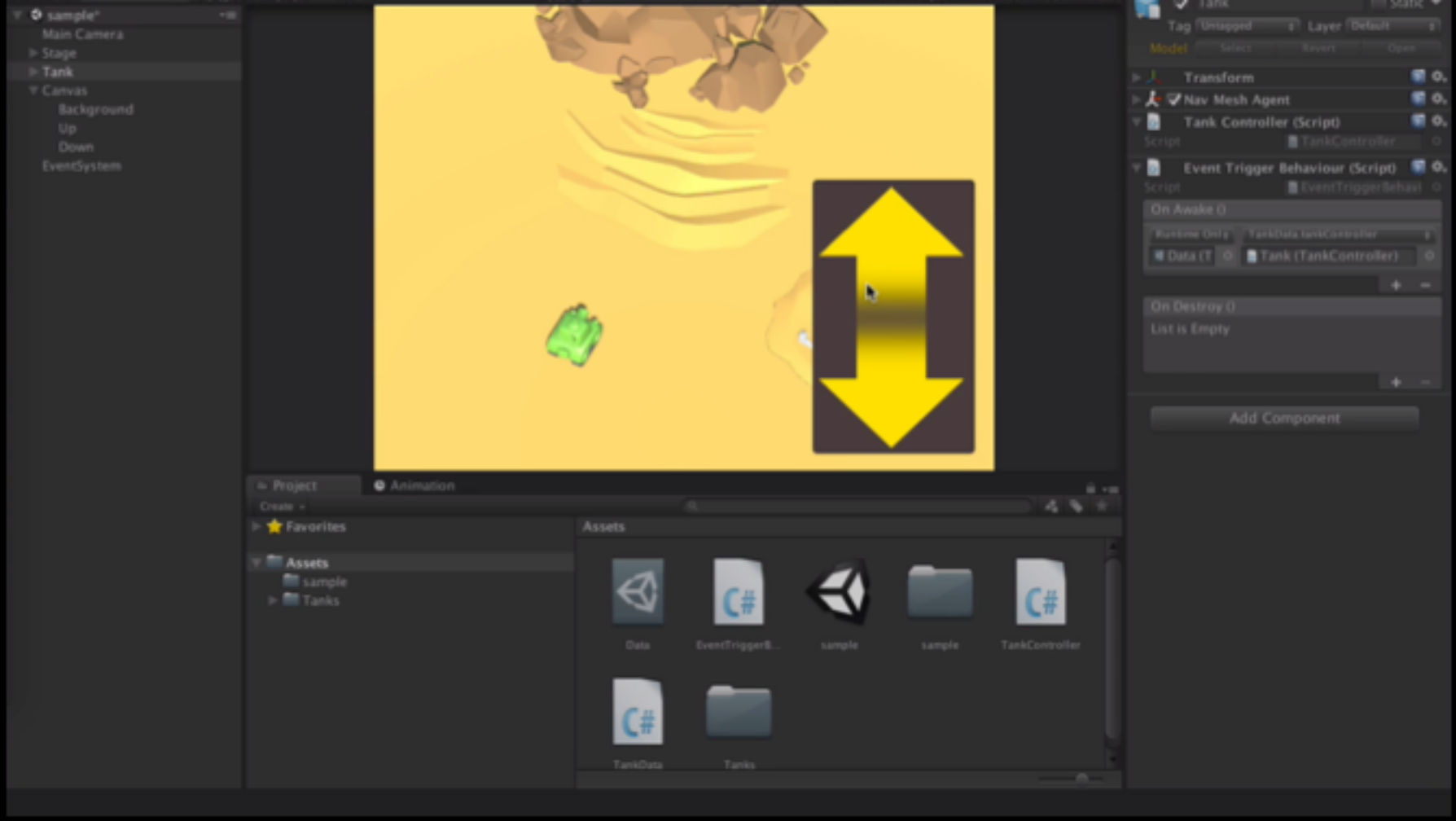
unity


Center
Global
Collab
Account
Layers
Layout
Hierarchy
sample*
Main Camera
Stage
Tank
Canvas
Background
Up
Down
EventSystem
Scene
Game
Inspector
Services
Navigation
Rect Transform
Canvas Renderer
Image (Script)
Source Image
Aim Arrow
Color
Material
None (Material)
Raycast Target
Image Type
Sliced
Fill Center
Event Trigger (Script)

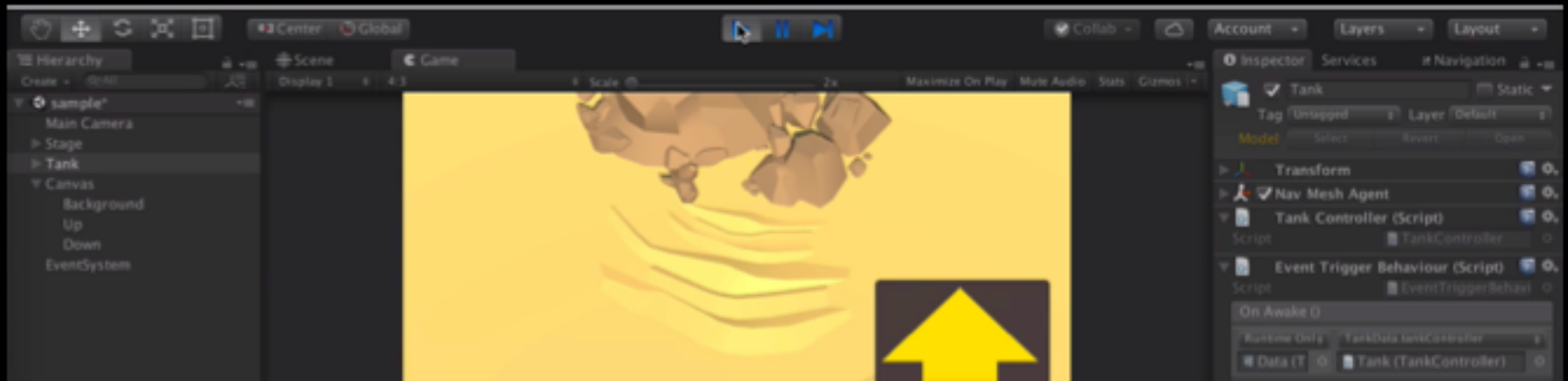

sample*
Main Camera
Stage
Tank
Canvas
Background
Up
Down
EventSystem
Image (Script)
Source Image
Aim Arrow
Color
Material
None (Material)
Raycast Target
Image Type
Fill Center
Event Trigger (Script)
Pointer Down (BaseEventData)
Add New Event Type
Default UI Material
Project
Animation
Favorites
Assets
sample
Tanks
Data
sample
sample
TankController
TankData
Tanks
Down
Image Size: 256x256
Ignoring asset refresh of packages:/Config/manifest.json because the file isn't in the Assets directory!

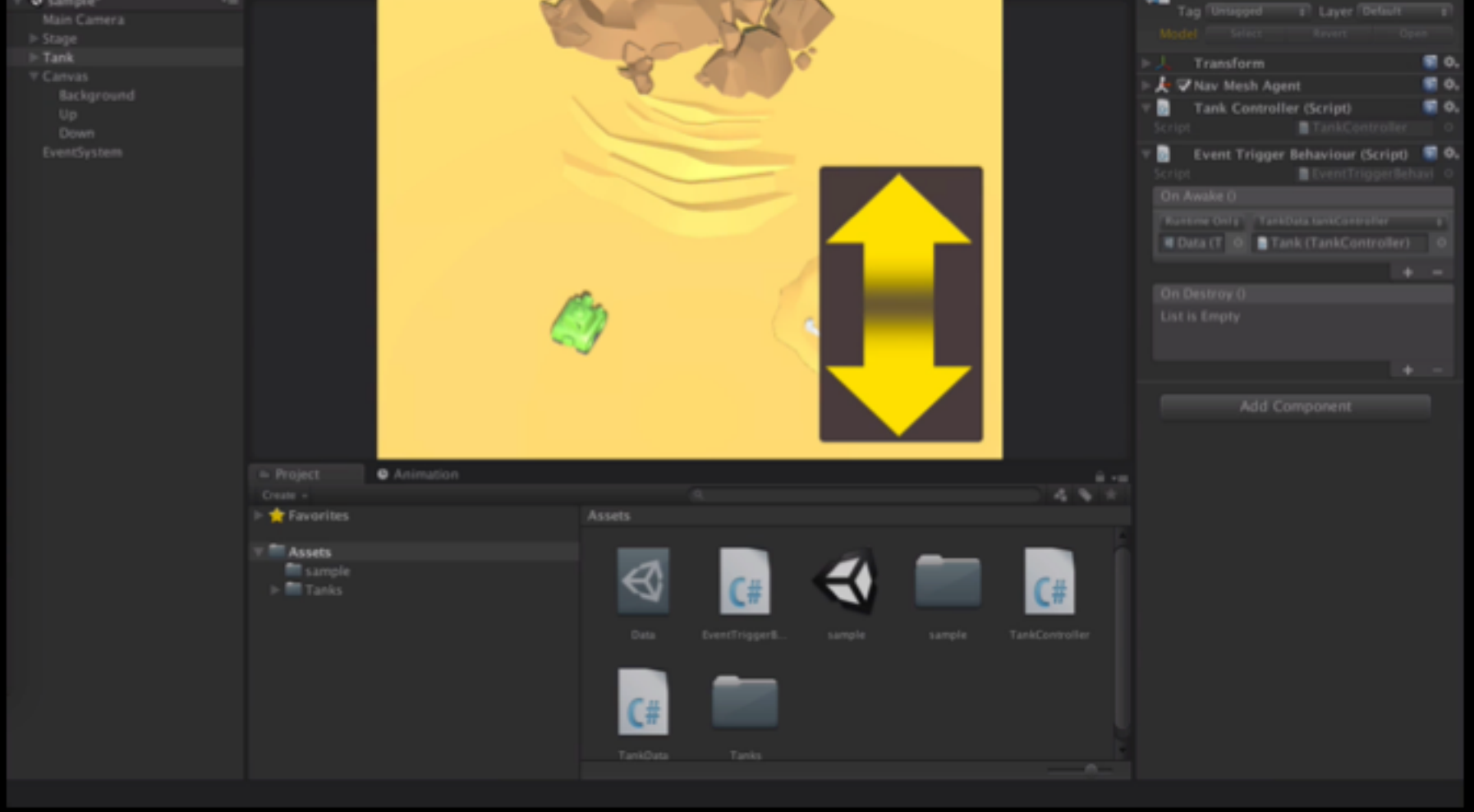
unity


Hierarchy
Scene
Game
Inspector
Services
Navigation
sample*
Main Camera
Stage
Tank
Canvas
Background
Up
Down
EventSystem
Transform
Nav Mesh Agent
Tank Controller (Script)
Event Trigger Behaviour (Script)
On Awake ()

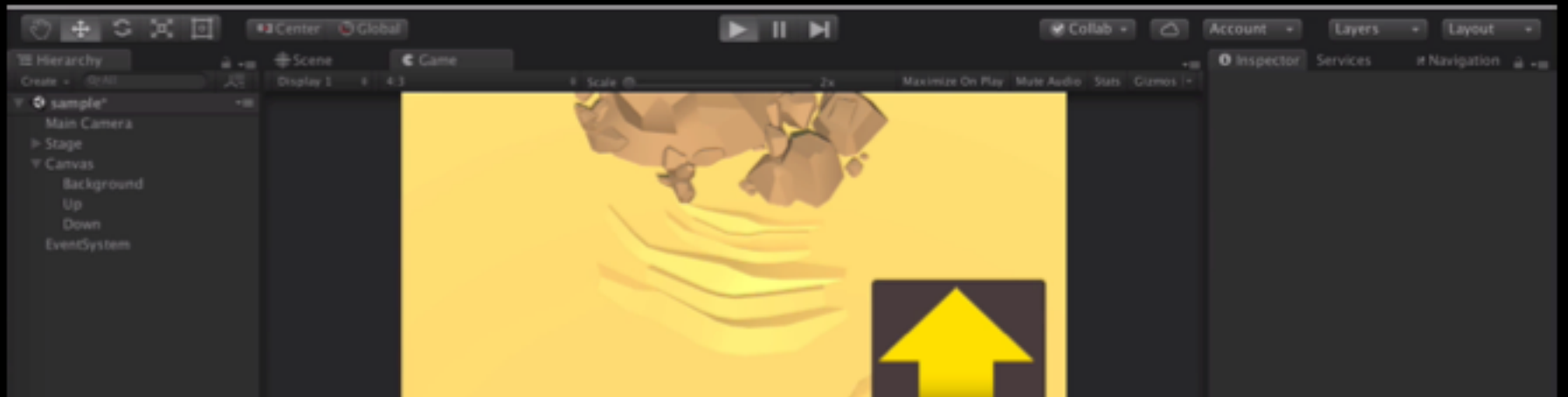

sample*
Main Camera
Stage
Tank
Canvas
Background
Up
Down
EventSystem
Transform
Nav Mesh Agent
Tank Controller (Script)
Event Trigger Behaviour (Script)
On Awake ()
On Destroy ()
List is Empty
Add Component
Project
Animation
Create
Favorites
Assets
sample
Tanks
Data
sample
sample
TankController
TankData
Tanks

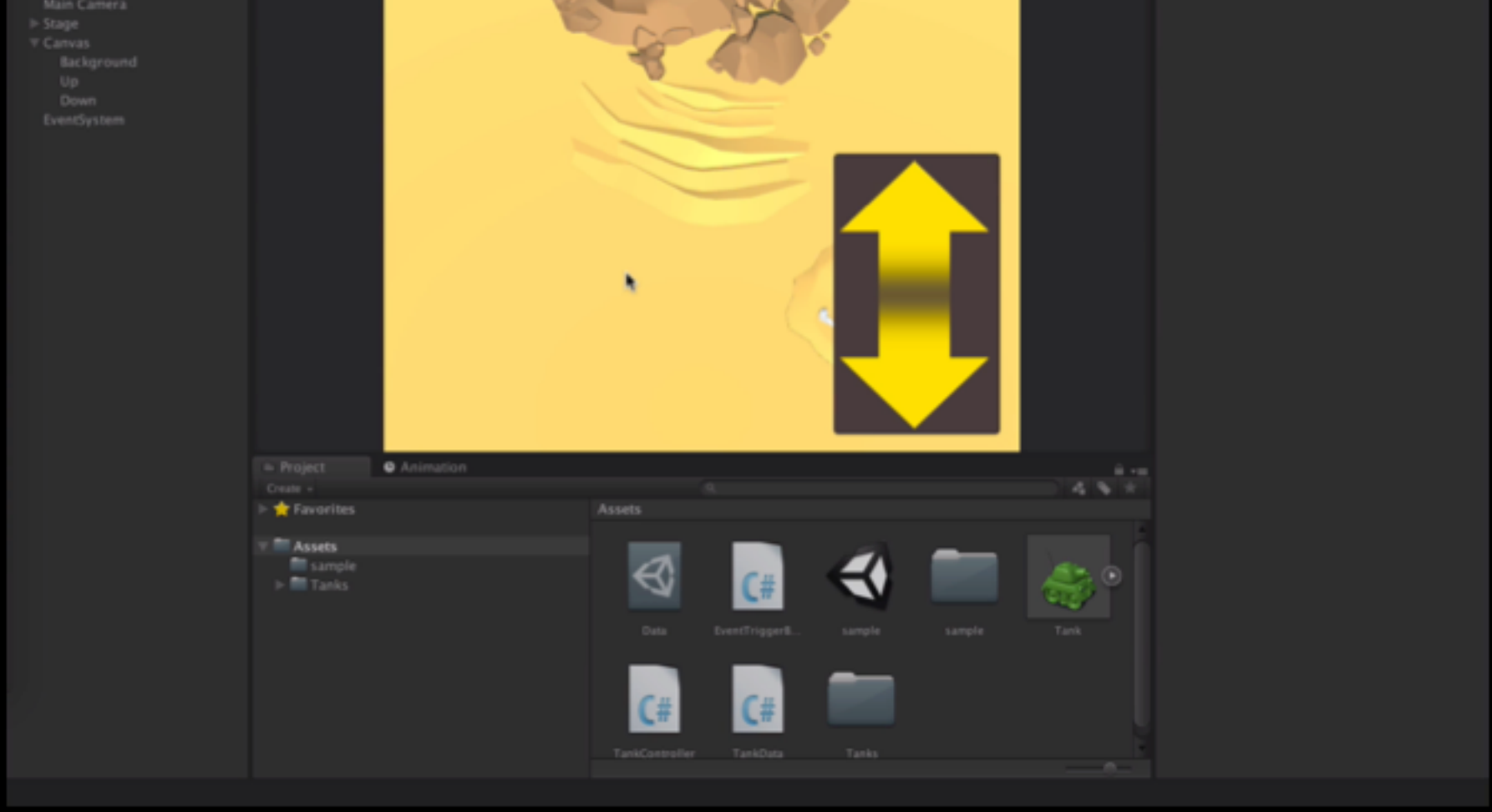
unity


Center
Global
Collab
Account
Layers
Layout
Hierarchy
Scene
Game
Inspector
Services
Navigation
sample*
Main Camera
Stage
Tank
Canvas
Background
Up
Down
EventSystem
Transform
Nav Mesh Agent
Tank Controller (Script)
Event Trigger Behaviour (Script)
On Awake ()

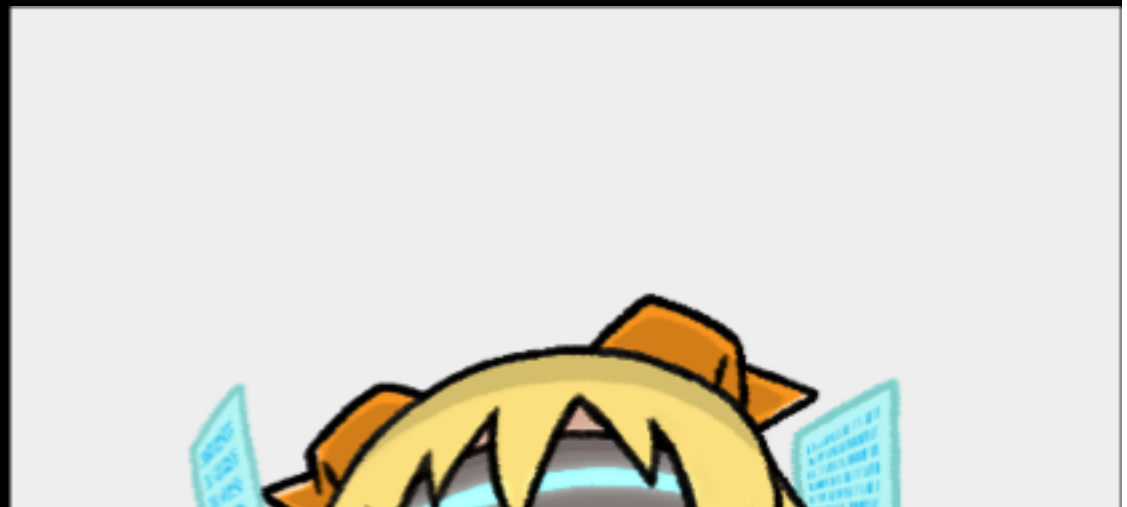

Main Camera
Stage
Tank
Canvas
Background
Up
Down
EventSystem
Transform
Nav Mesh Agent
Tank Controller (Script)
Event Trigger Behaviour (Script)
On Awake ()
On Destroy ()
List is Empty
Add Component
Project
Animation
Create
Favorites
Assets
sample
Tanks
Data
sample
sample
TankController
TankData
Tanks

unity


Center
Global
Collab
Account
Layers
Layout
Hierarchy
Scene
Game
Inspector
Services
Navigation
Create
Display 1
Maximize On Play
Mute Audio
Stats
Gizmos
sample*
Main Camera
Stage
Canvas
Background
Up
Down
EventSystem

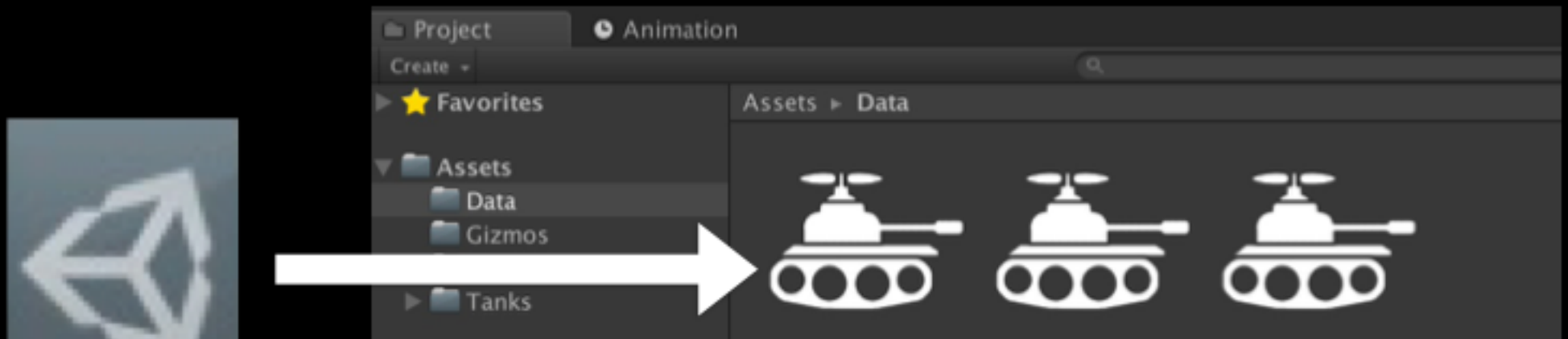

Main Camera
Stage
Canvas
Background
Up
Down
EventSystem
Project
Animation
Create
Favorites
Assets
sample
Tanks
Assets
Data
EventTriggerB...
sample
sample
Tank
TankController
TankData
Tanks

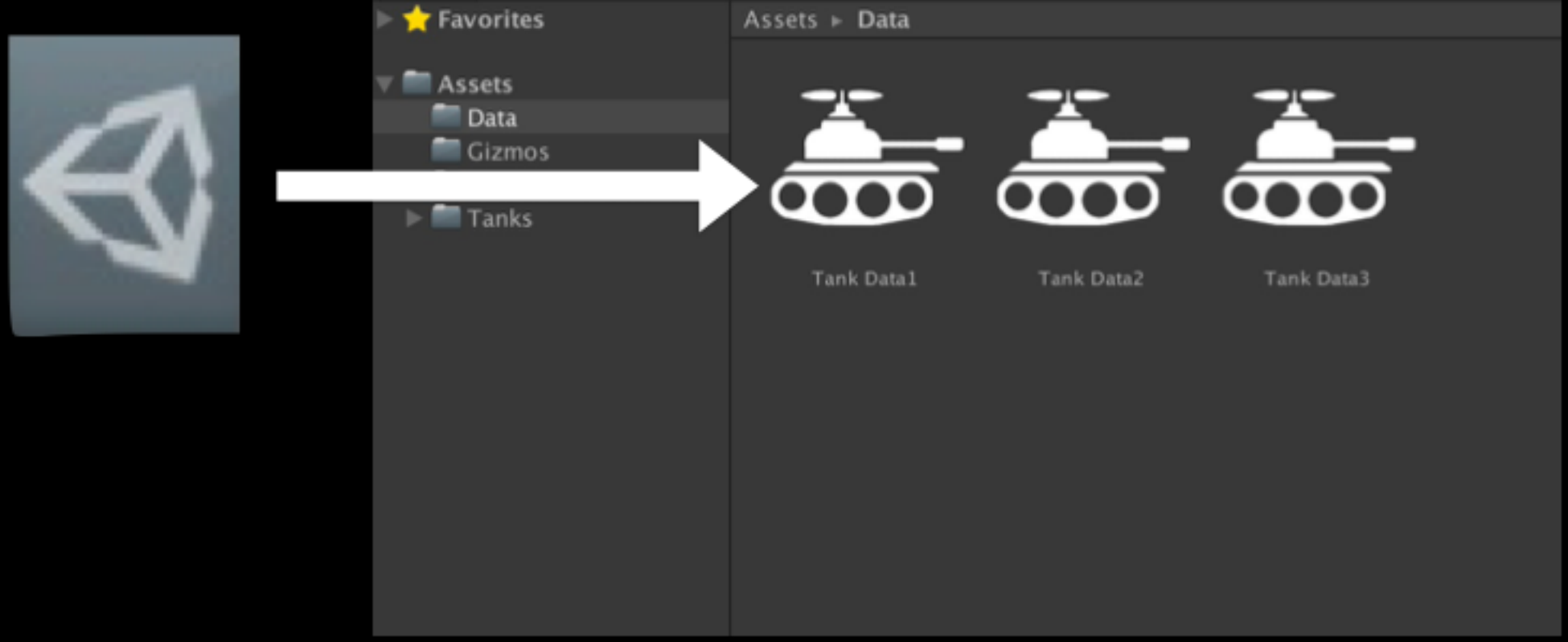
unity



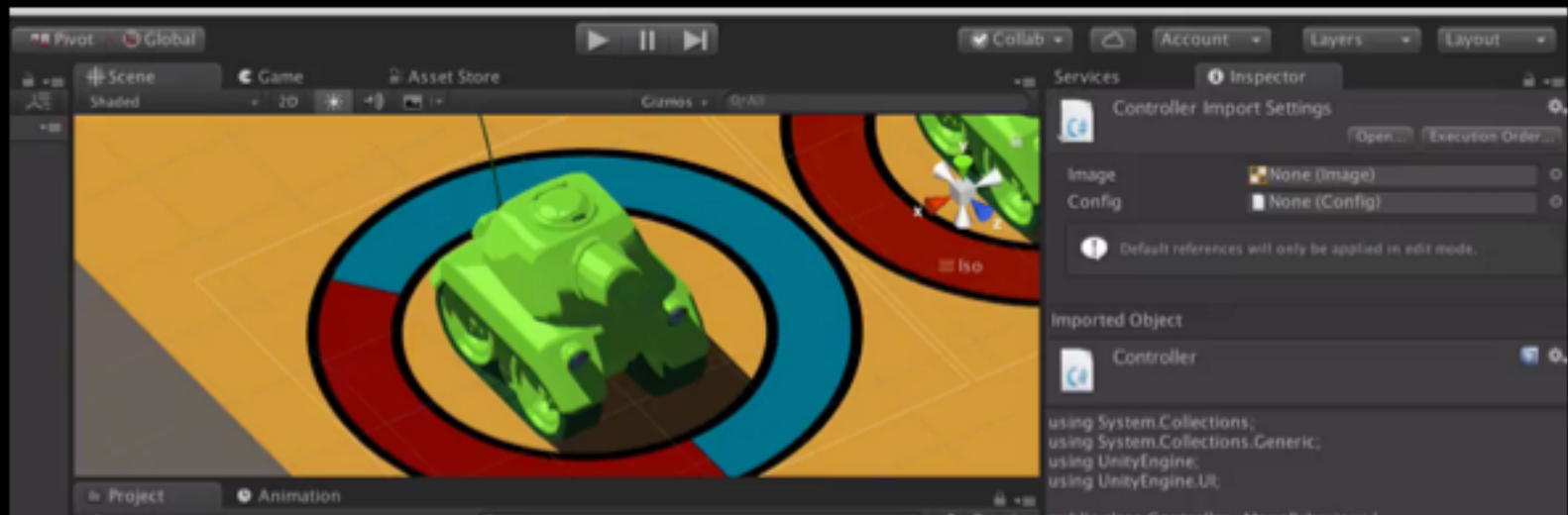
応用

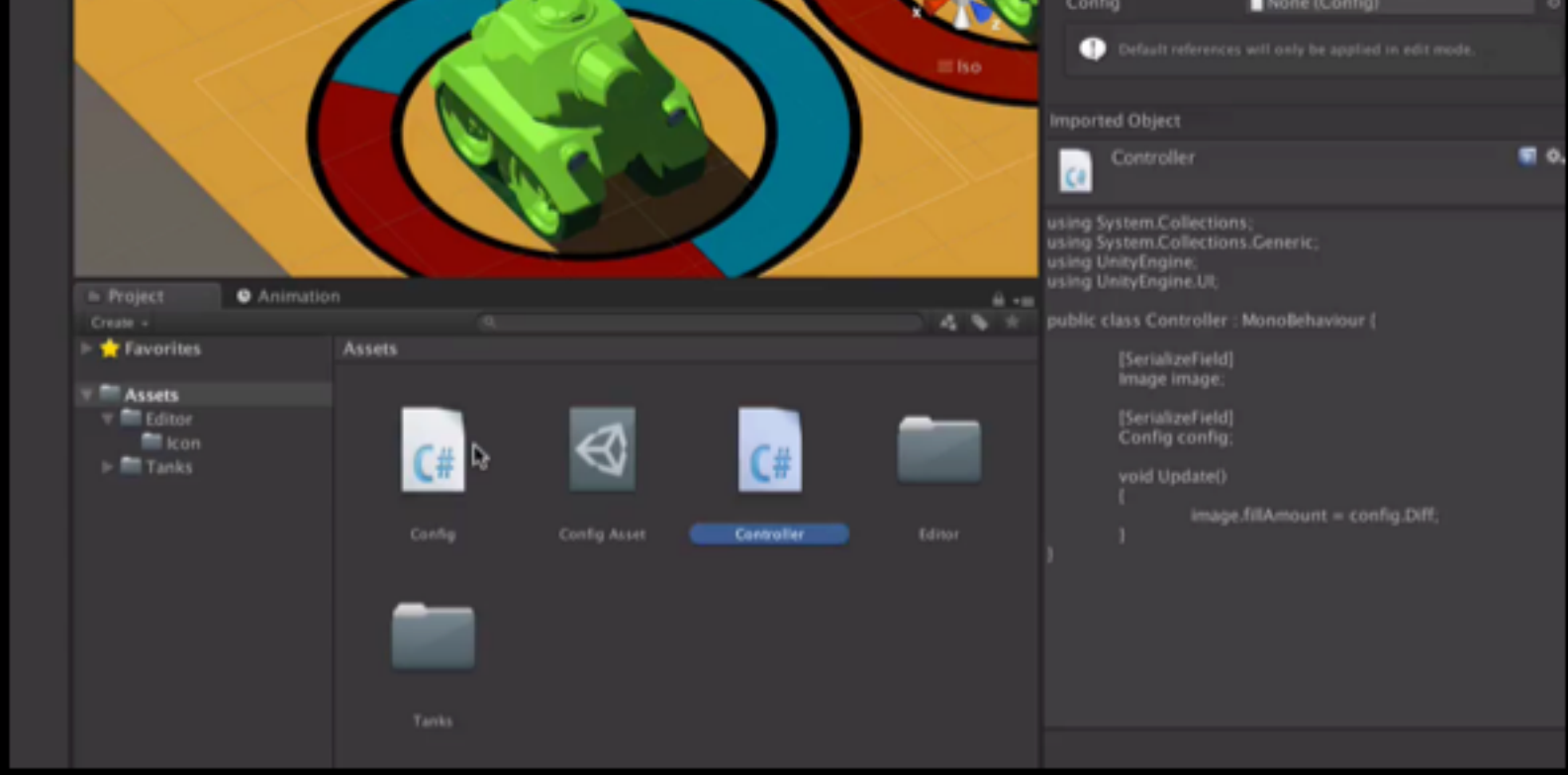
もっとScriptable Objectを
使いこなすTips



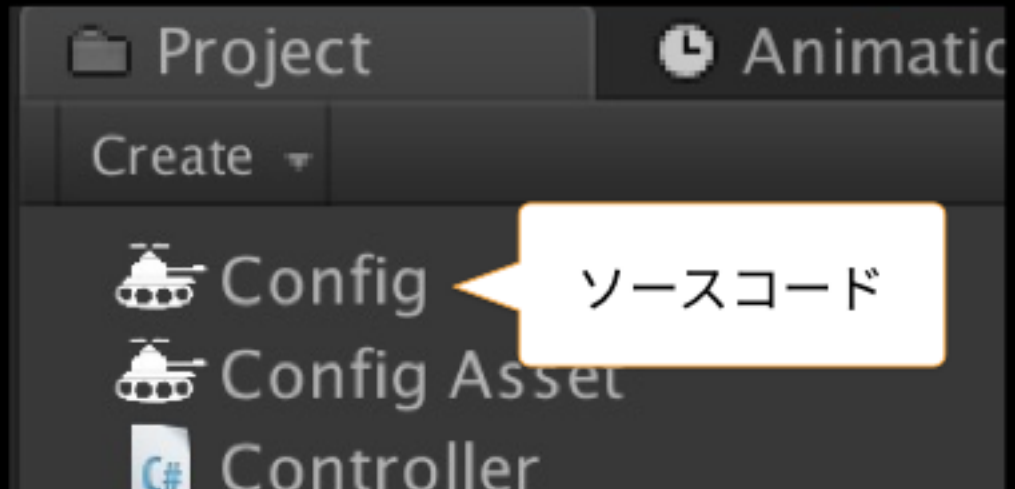


●Scriptable Objectのアイコンを変えて、一覧性を上げる

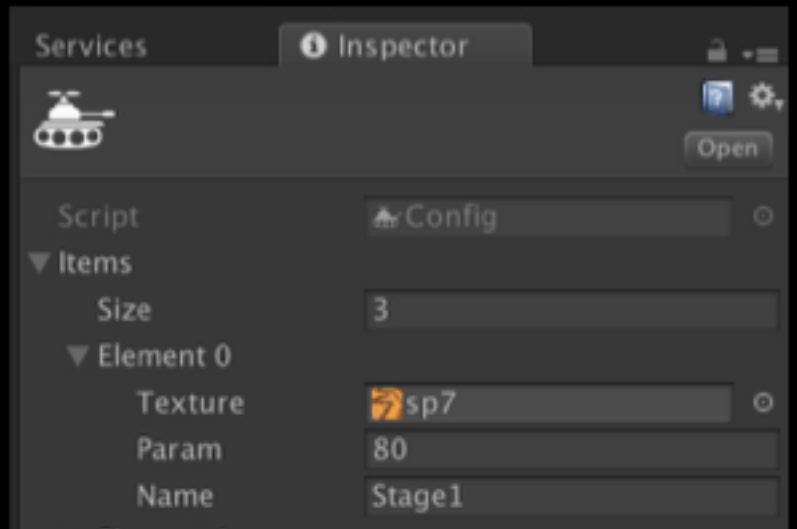


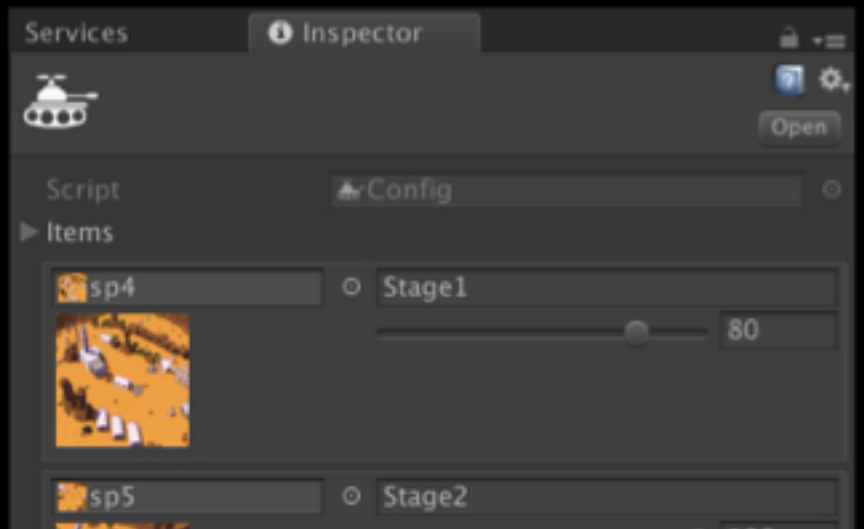


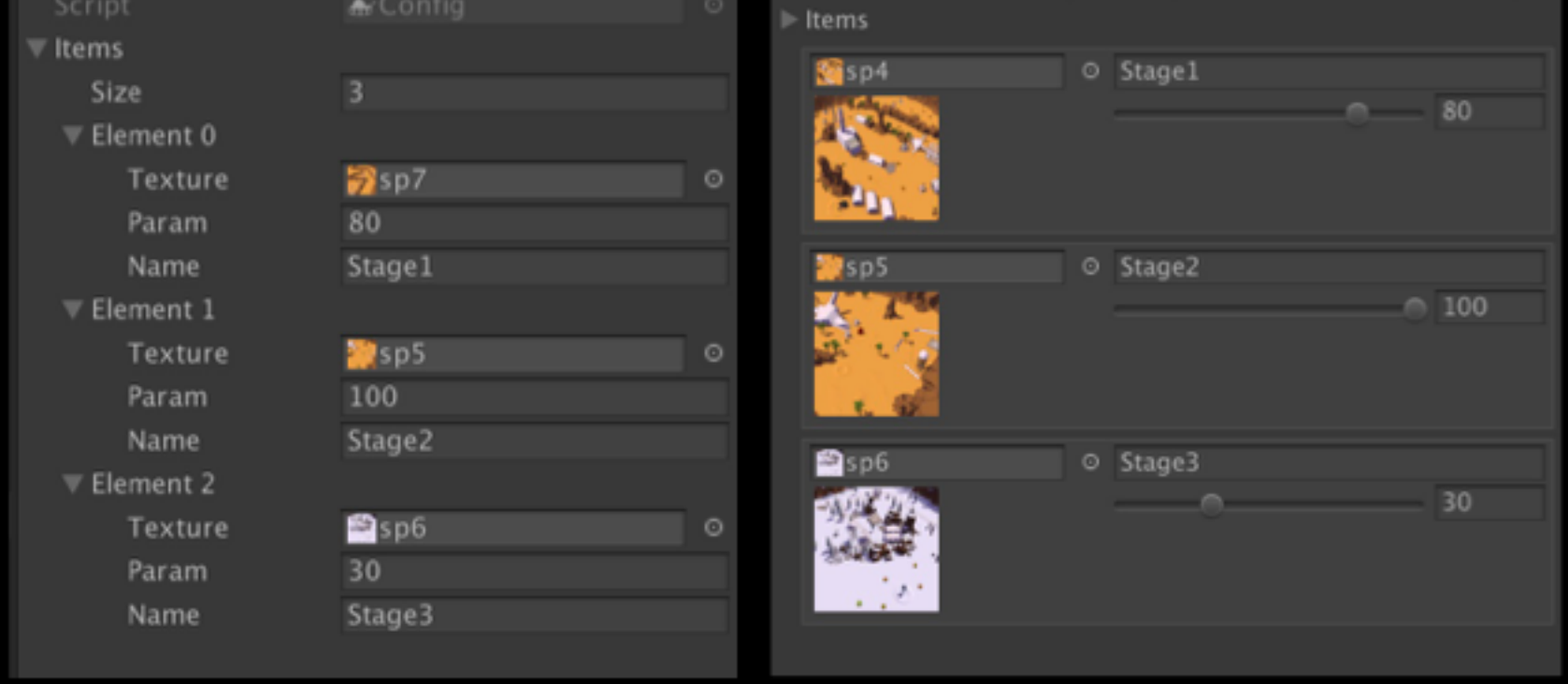
●ScriptableObjectのコードアイコンを変更すると、SOのアイコンが変わる



●Gizmosフォルダ以下に「**[クラス名] Icon**」ファイルを置くとアイコンが変化







●エディター拡張でScriptableObjectのUIを変更

ScriptableObjectのUIを拡張

拡張するクラスを登録

```
[CreateAssetMenu]
public class Config : ScriptableObject
{
```

```
[CustomEditor (typeof(Config))]
public class ConfigEditor : Editor
{
```

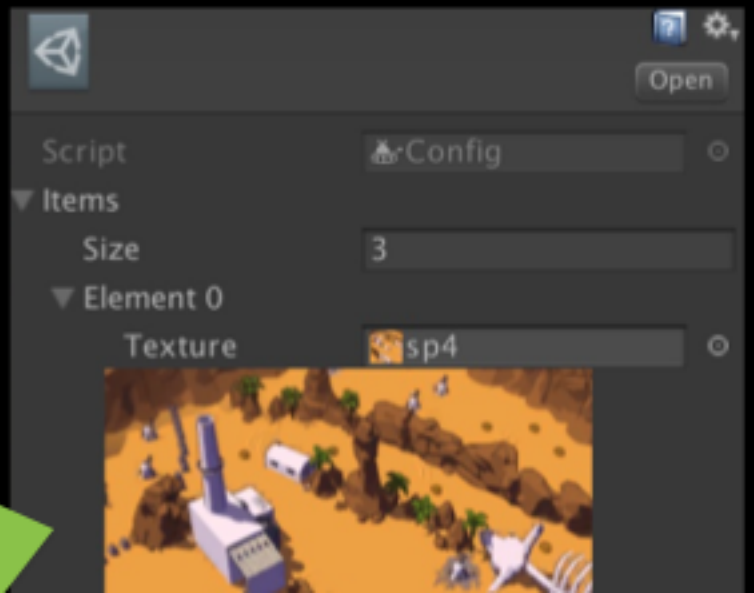
```
[CreateAssetMenu]
public class Config : ScriptableObject
{
  public Item[] items = new Item[0];

  [System.Serializable]
  public class Item
  {
    public Texture texture;
    public int param;
    public string name;
  }
}
```

```
[CustomEditor (typeof(Config))]
public class ConfigEditor : Editor
{
  public override void OnInspectorGUI ()
  {
    base.OnInspectorGUI();
    var config = (Config)target;
    // レイアウト記述
  }
}
```

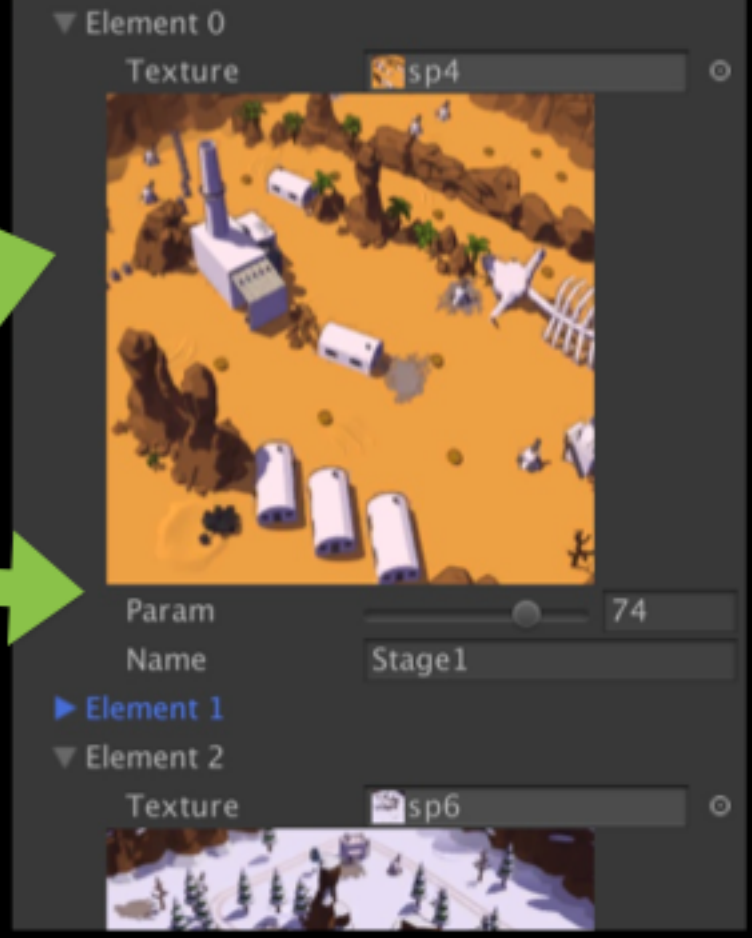
レイアウトを拡張するコードを記述

PropertyDrawerを使う

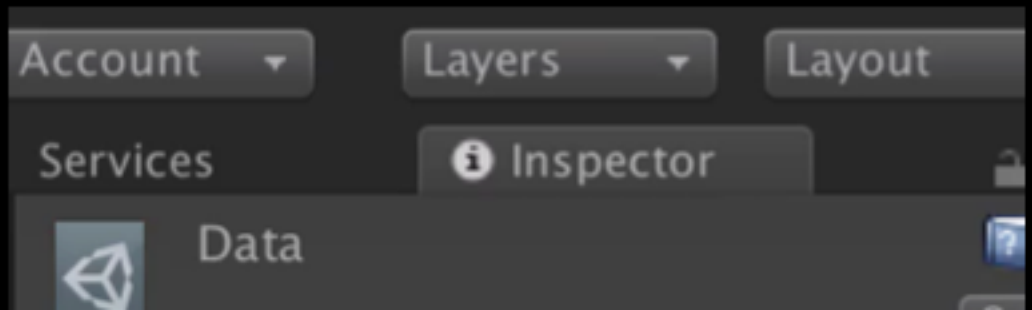
```
[CreateAssetMenu]
public class Config : ScriptableObject
{
  public Item[] items = new Item[0];

  [System.Serializable]
```



```
[CreateAssetMenu]
public class Config : ScriptableObject
{
   public Item[] items = new Item[0];

   [System.Serializable]
   public class Item
   {
      [PreviewTexture]
      public Texture texture;
      [Range(0, 100)]
      public int param;
      public string name;
   }
}
```

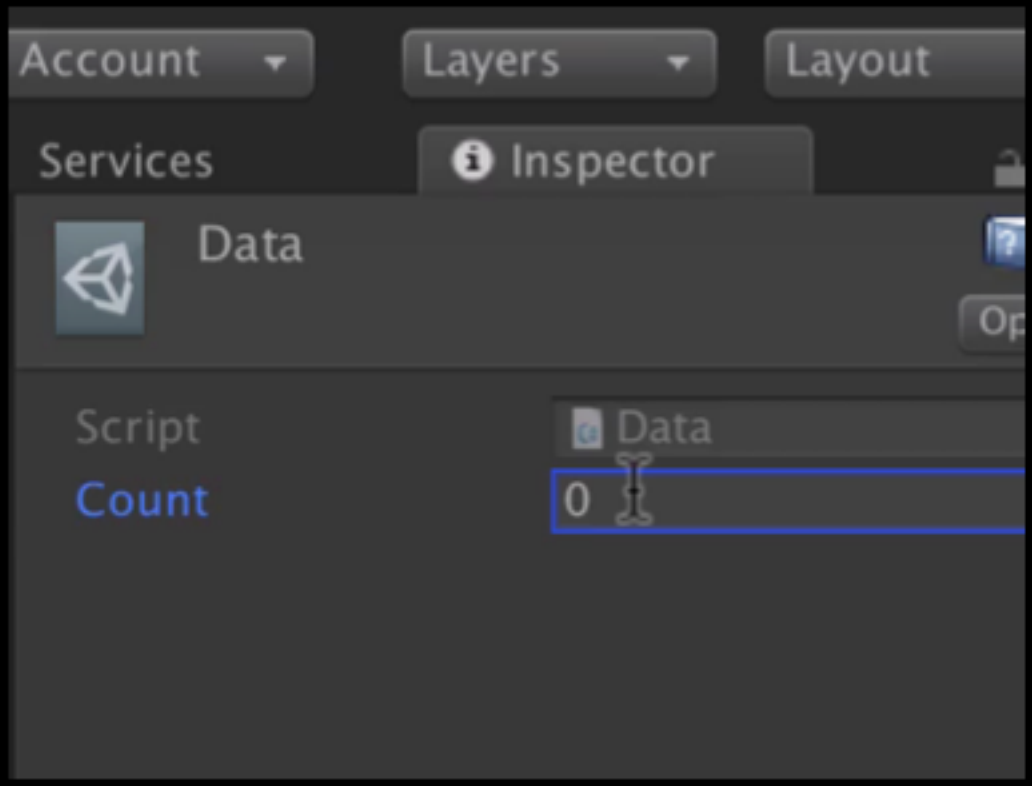


OnValidateを使う



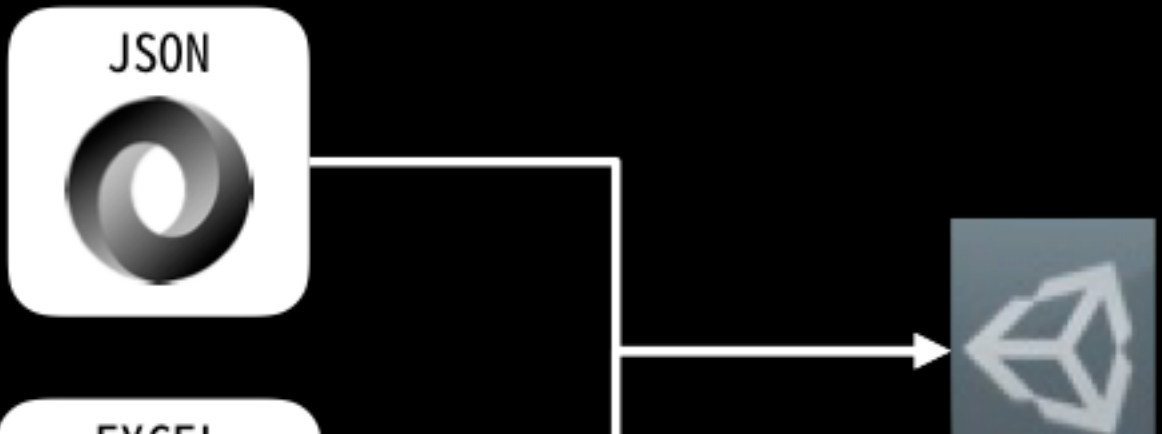

```
[CreateAssetMenu]
public class Data : Sc
{
   [SerializeField] int
```

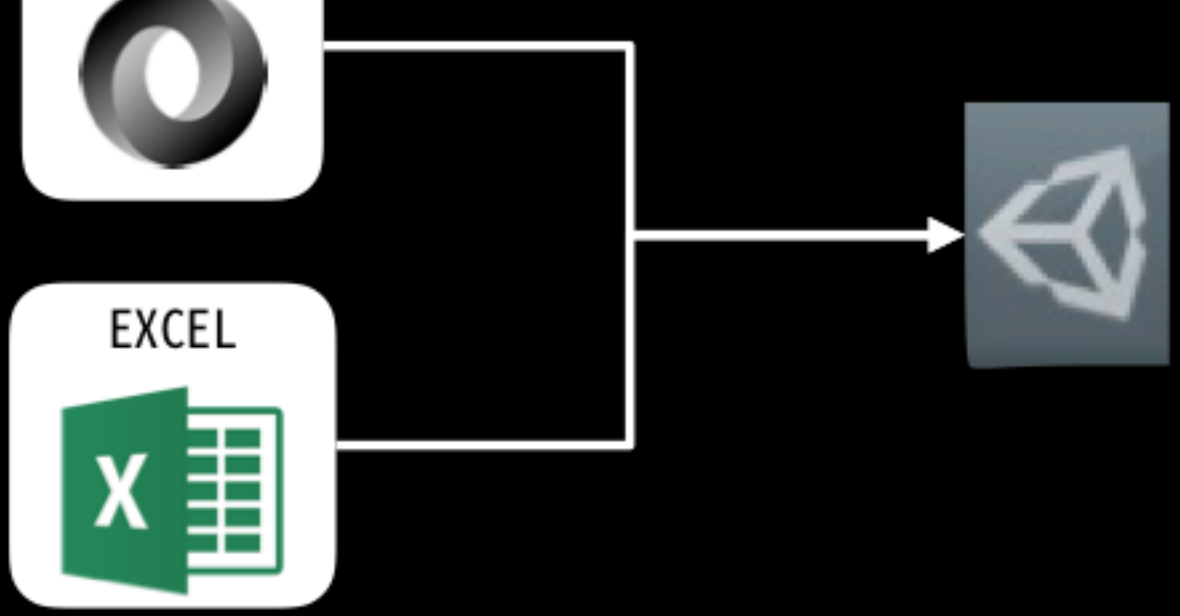
エディターで値を
変更時に呼ばれる




```
[CreateAssetMenu]
public class Data : Sc
{
  [SerializeField] int

  void OnValidate ()
  {
    count = Mathf.Clamp (count, 0, 99);
  }
}
```

エディターで値を
変更時に呼ばれる





●アセットを外部ファイルから作る

何故EXCEL？

●数値編集・入力で最も優れたツール

●データテーブルを作るのに便利

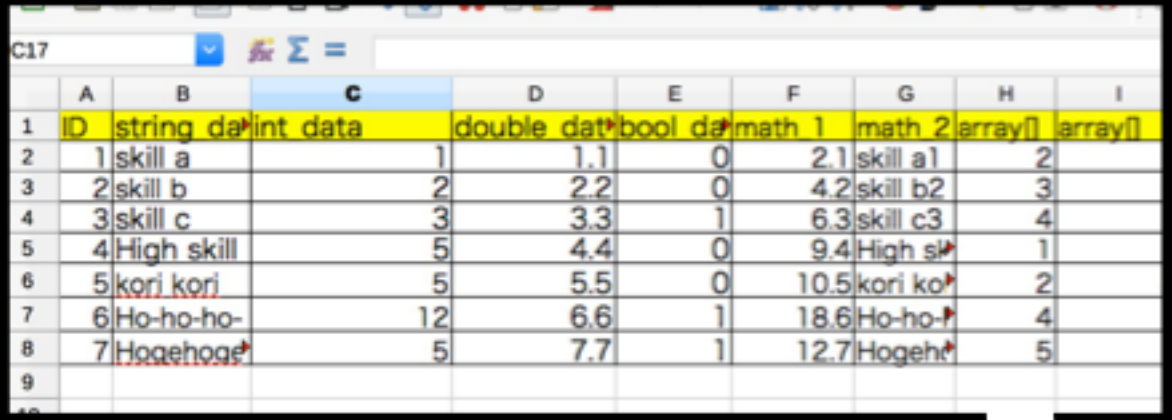
●バランスの良いゲームは、

●数値編集・入力で最も優れたツール

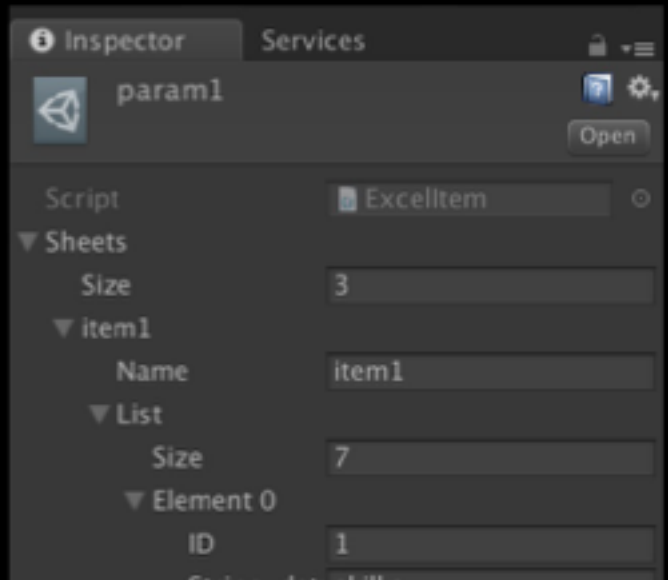
●データテーブルを作るのに便利

●バランスの良いゲームは、
ランダムではなく都合の良いテーブルを使う事がある

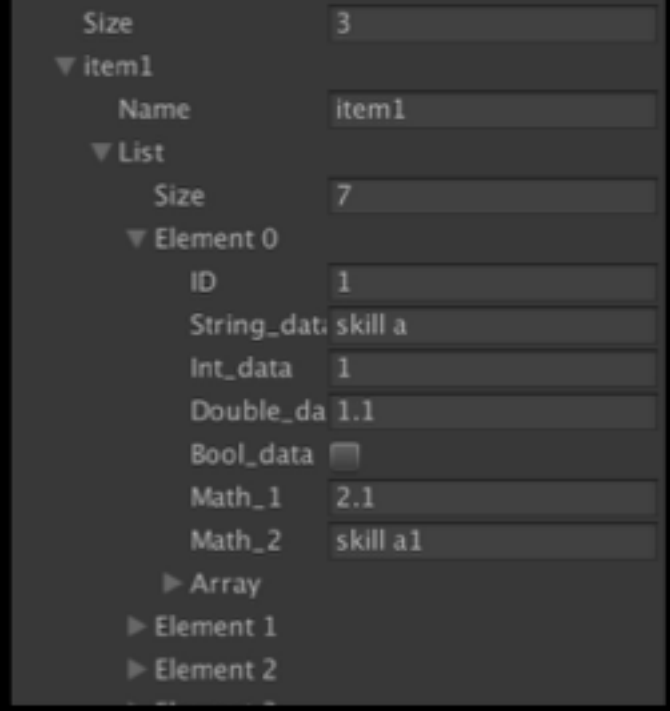
●Excelは実行時に読むようなデータ構造ではない



	A	B	C	D	E	F	G	H	I
1	ID	string_da	int_data	double_dat	bool_da	math_1	math_2	array[]	array[]
2	1	skill a	1	1.1	0	2.1	skill a1	2	
3	2	skill b	2	2.2	0	4.2	skill b2	3	
4	3	skill c	3	3.3	1	6.3	skill c3	4	
5	4	High skill	5	4.4	0	9.4	High sk	1	
6	5	kori kori	5	5.5	0	10.5	kori ko	2	
7	6	Ho-ho-ho-	12	6.6	1	18.6	Ho-ho-h	4	
8	7	Hogehoge	5	7.7	1	12.7	Hogeho	5	
9									



4	3	skill c	3	3.3	1	6.3	skill c3	4
5	4	High skill	5	4.4	0	9.4	High sk	1
6	5	kori kori	5	5.5	0	10.5	kori ko	2
7	6	Ho-ho-ho-	12	6.6	1	18.6	Ho-ho-h	4
8	7	Hogehoge	5	7.7	1	12.7	Hogeh	5
9								



●エンジンが読みやすいScriptable Objectに変換する

AssetPostprocessorで
アセットのインポート処理に
割り込み

既にあればLoadAssetAtPath
無ければCreateAssetで
ScriptableObjectを取得

既にあればLoadAssetAtPath
無ければCreateAssetで
ScriptableObjectを取得

EditorUtility.SetDirtyで
ScriptableObjectを変更

●エディターでExcelのインポート時、ScriptableObjectを生成

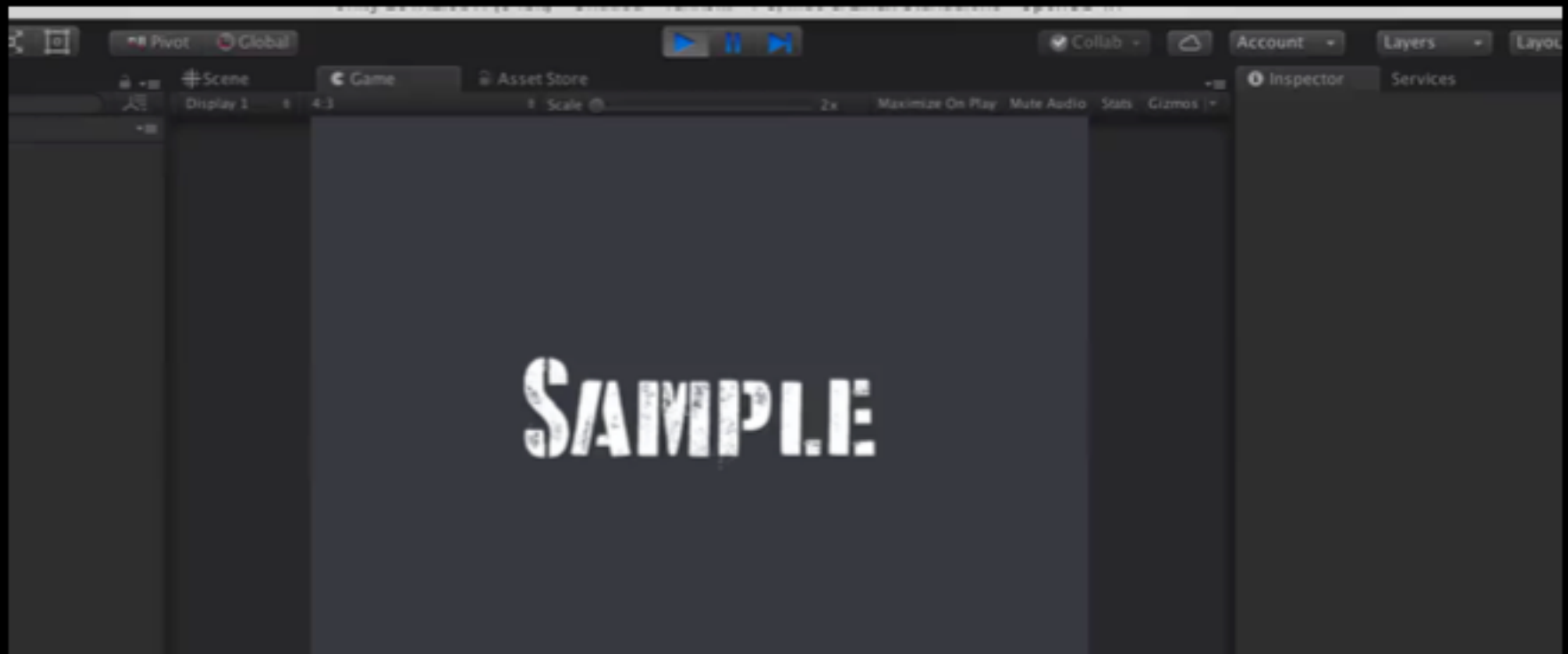
インポート処理に割込

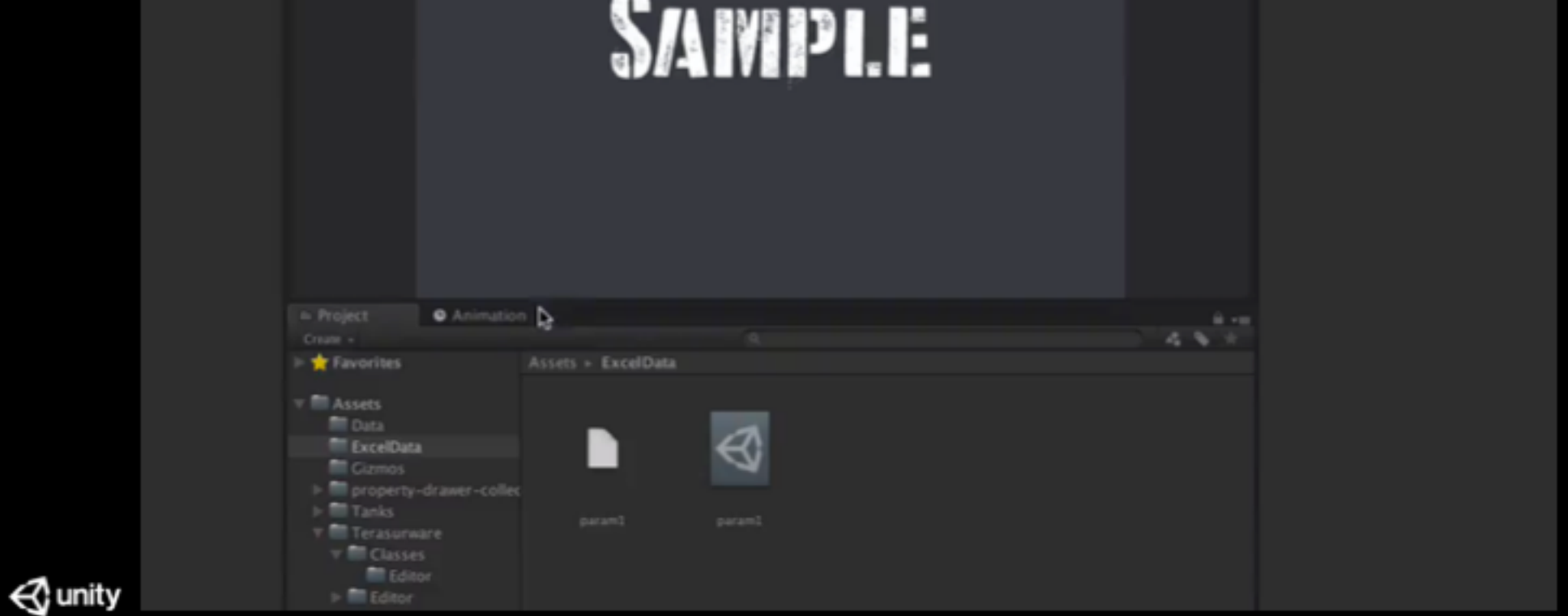
```
public class param1_importer : AssetPostprocessor
 {
  static void OnPostprocessAllAssets (
        string[] importedAssets, string[] deletedAssets, string[] movedAssets,
        string[] movedFromAssetPaths)
  {
      ExcelItem data = (ExcelItem)AssetDatabase.LoadAssetAtPath (exportPath, typeof(ExcelItem));
      if (data == null) {
        data = ScriptableObject.CreateInstance<ExcelItem> ();
        AssetDatabase.CreateAsset ((ScriptableObject)data, exportPath);
      }
```

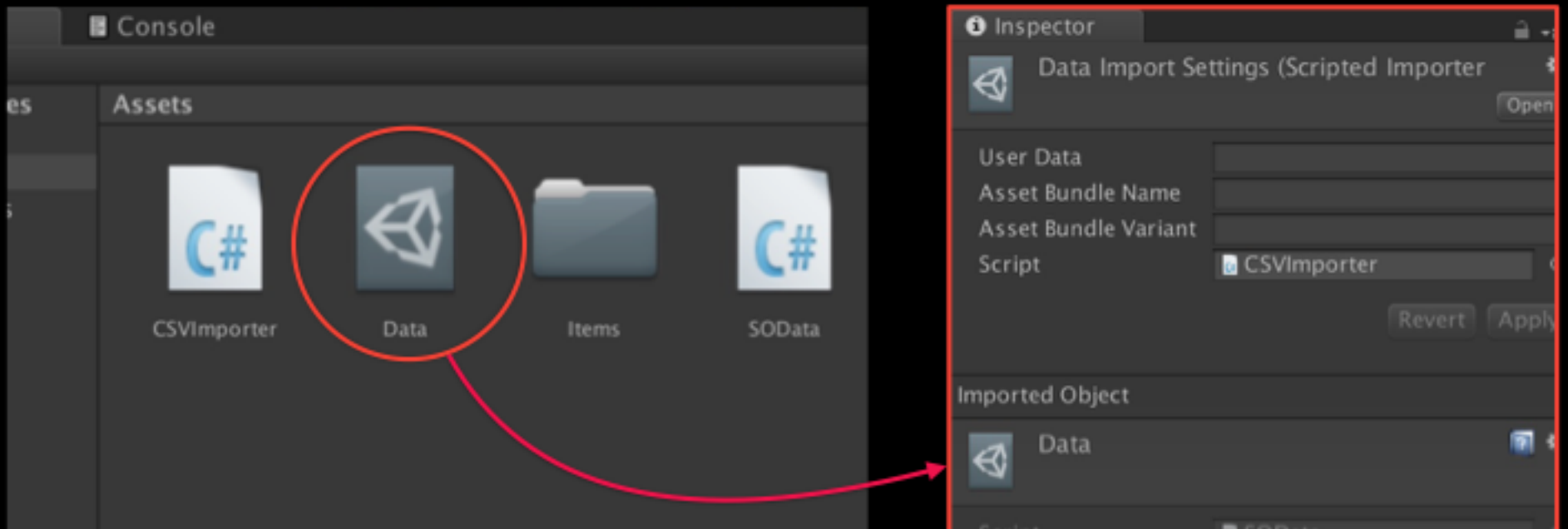
```
    {
        ExcelItem data = (ExcelItem)AssetDatabase.LoadAssetAtPath (exportPath, typeof(ExcelItem));
        if (data == null) {
          data = ScriptableObject.CreateInstance<ExcelItem> ();
          AssetDatabase.CreateAsset ((ScriptableObject)data, exportPath);
        }
        // インポート処理

        ScriptableObject obj = AssetDatabase.LoadAssetAtPath (
          exportPath, typeof(ScriptableObject)) as ScriptableObject;
        EditorUtility.SetDirty (obj);
    }
}
```

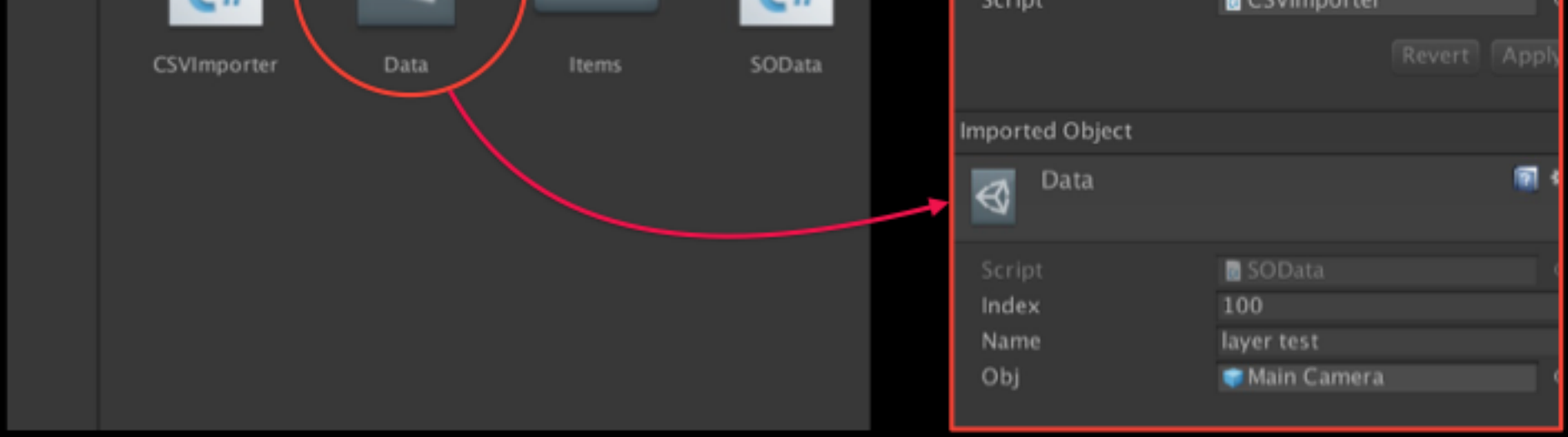
ScriptableObject作成

変更を反映



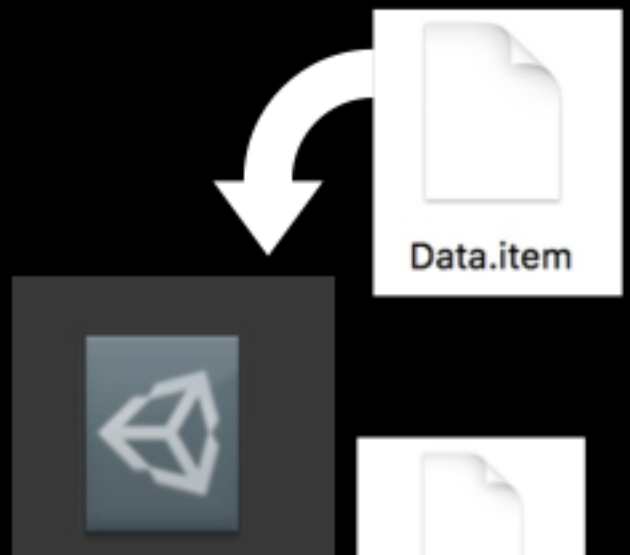

SAMPLE
Project
Animation
Create
Favorites
Assets
Data
ExcelData
Gizmos
property-drawer-collec
Tanks
Terasurware
Classes
Editor
Editor
Assets ▸ ExcelData
param1
param1
unity


Console
Assets
CSVImporter
Data
Items
SOData
Inspector
Data Import Settings (Scripted Importer
Open
User Data
Asset Bundle Name
Asset Bundle Variant
Script
CSVImporter
Revert
Apply
Imported Object
Data



●ScriptedImporterでアセットをScriptableObjectとして使う

以前



ScriptedImporter

Data.item

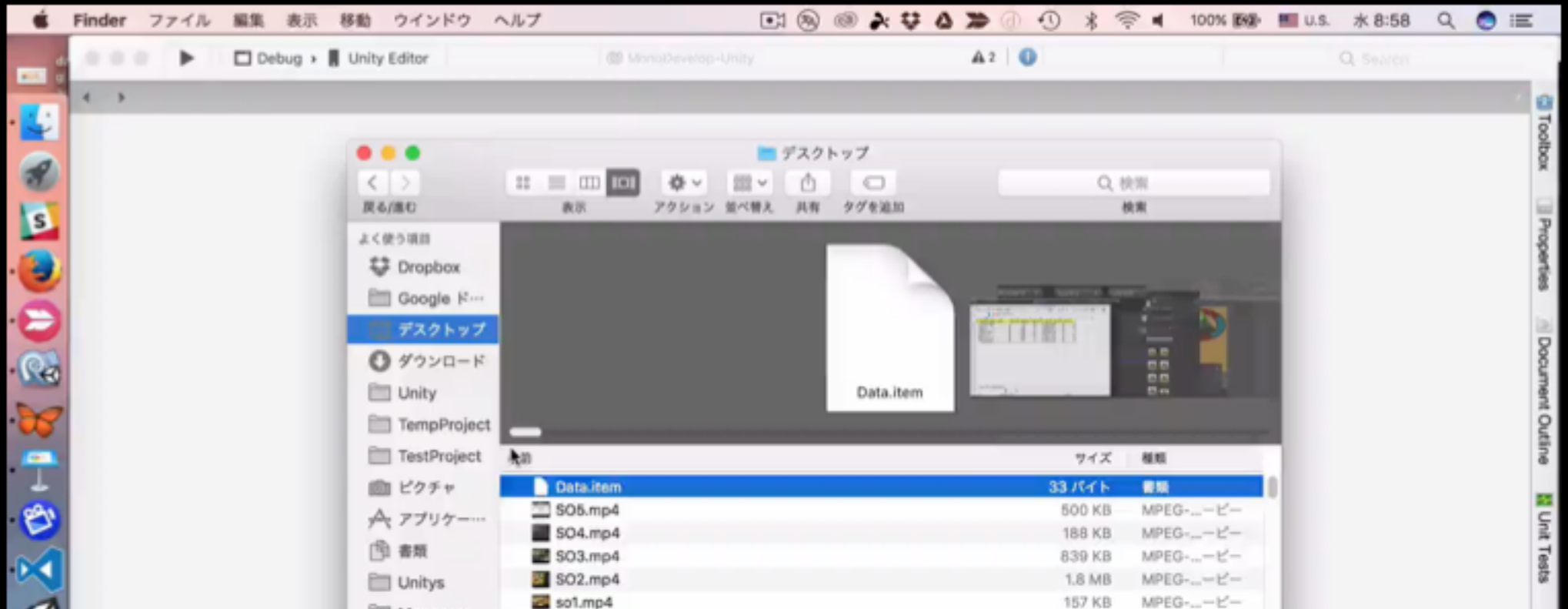
DataAsset

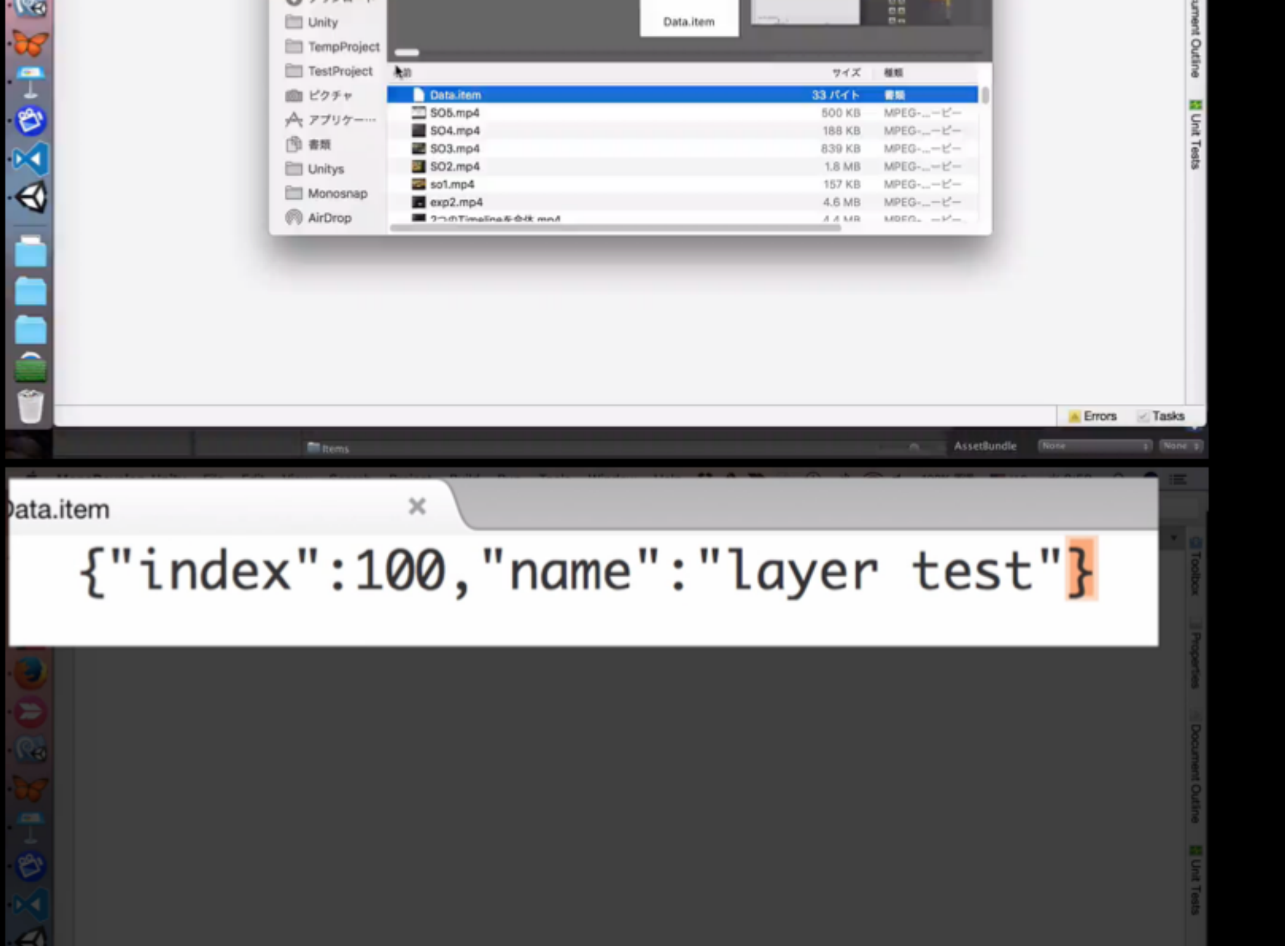
DataAsset.asset

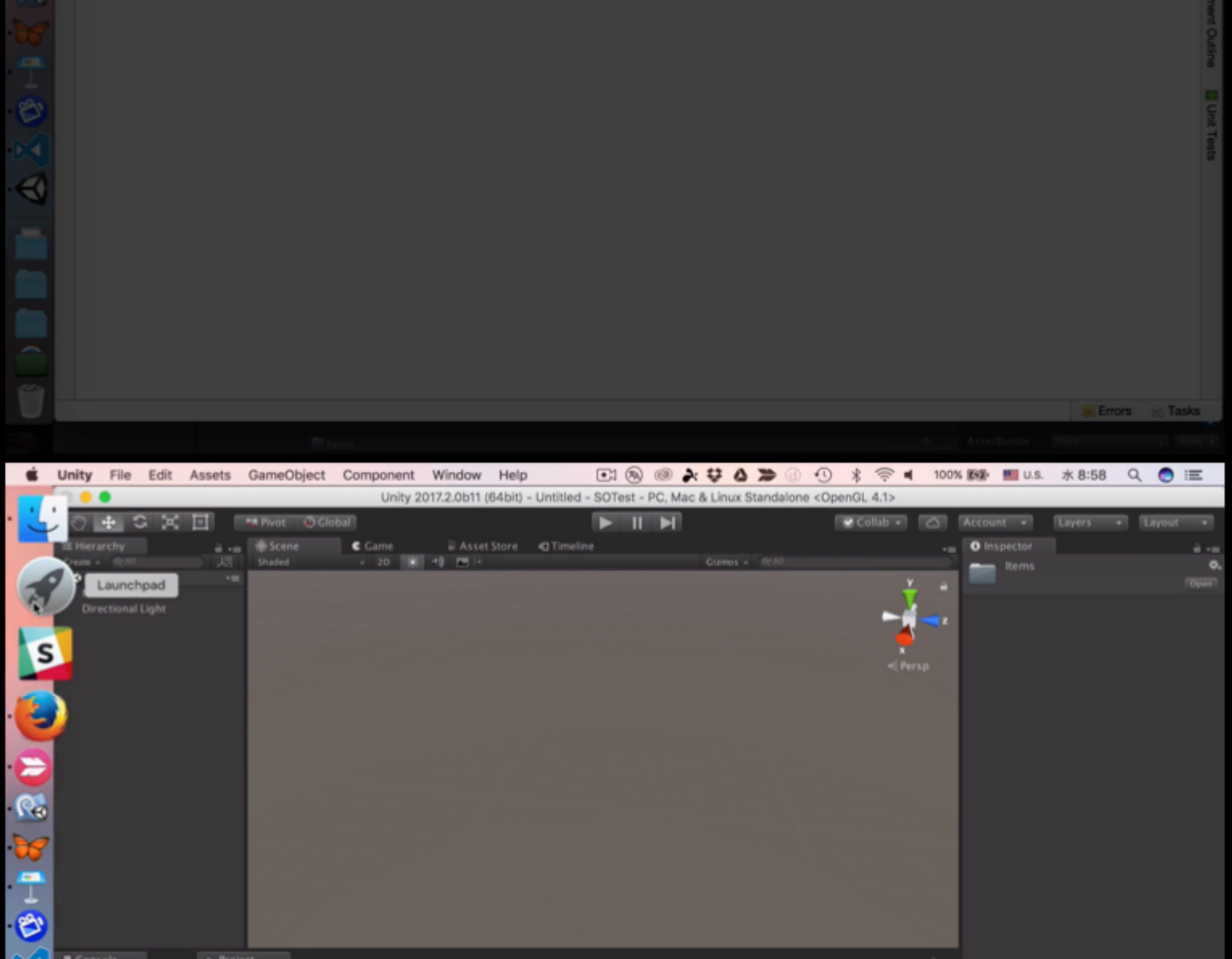
Data

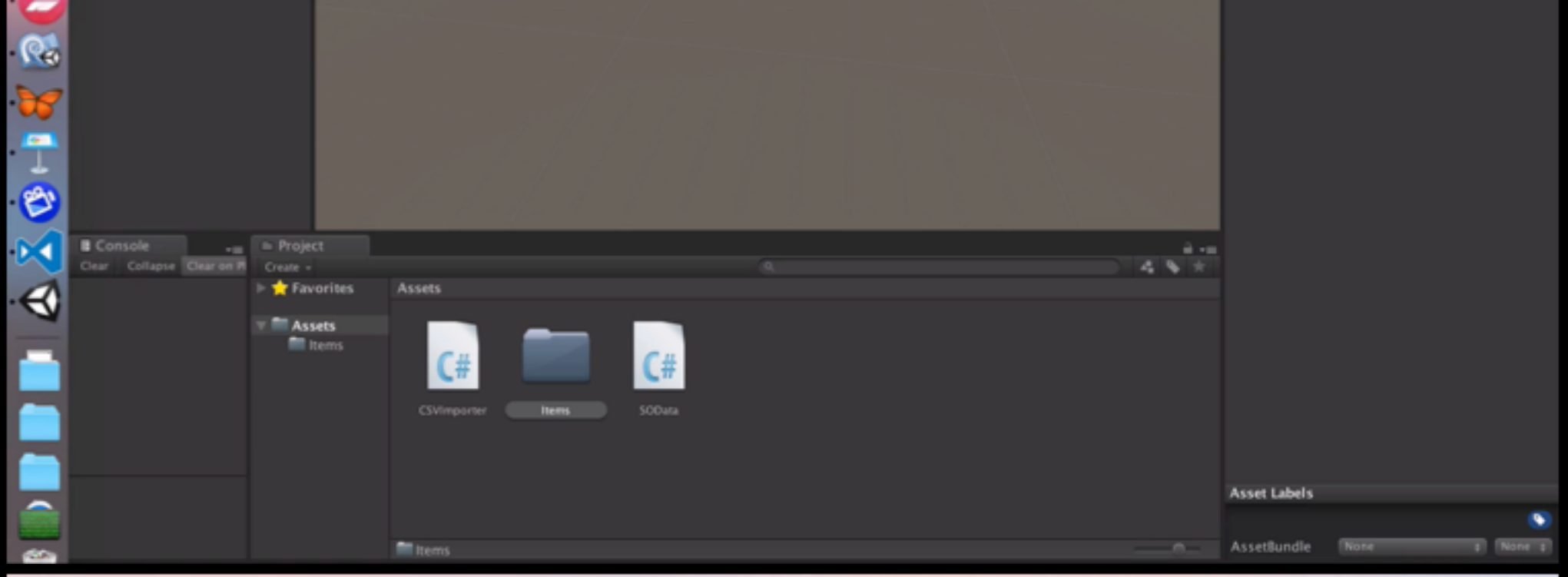
Data.item

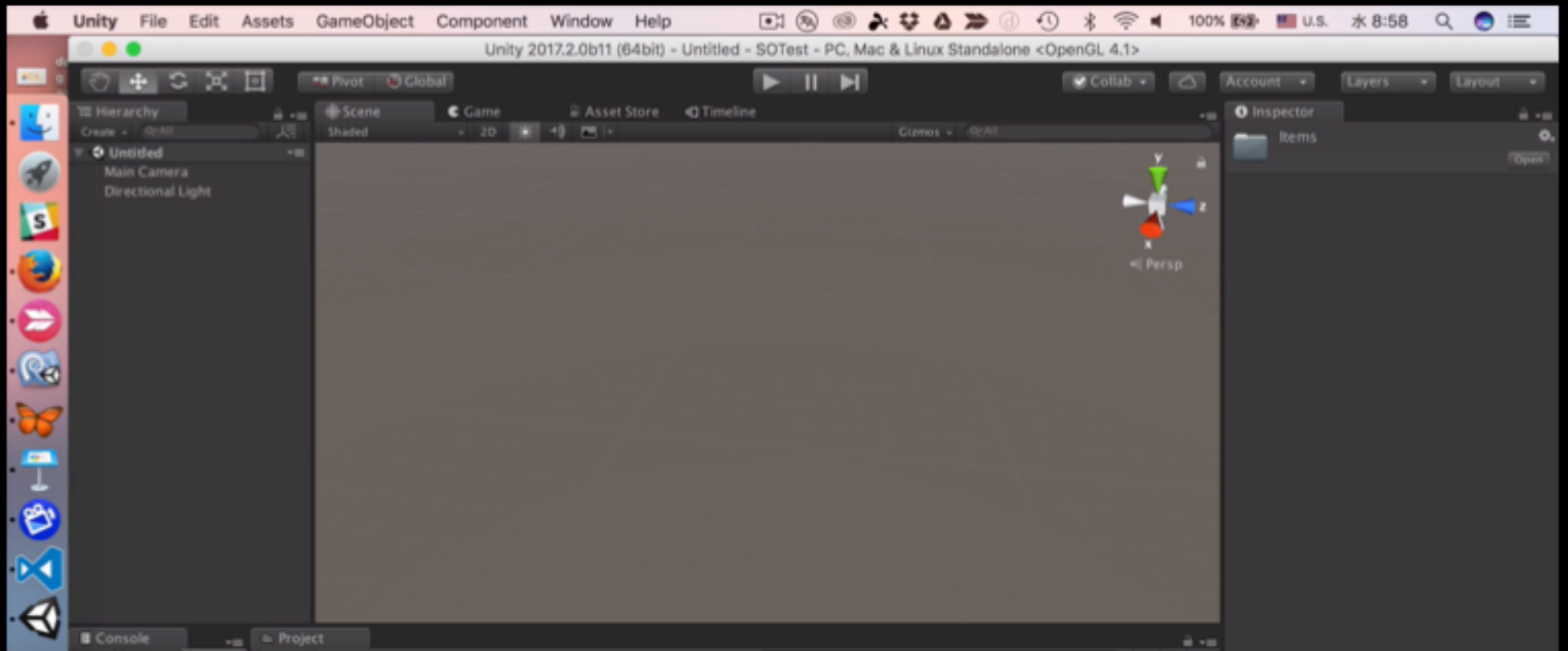
●アセットのように使える

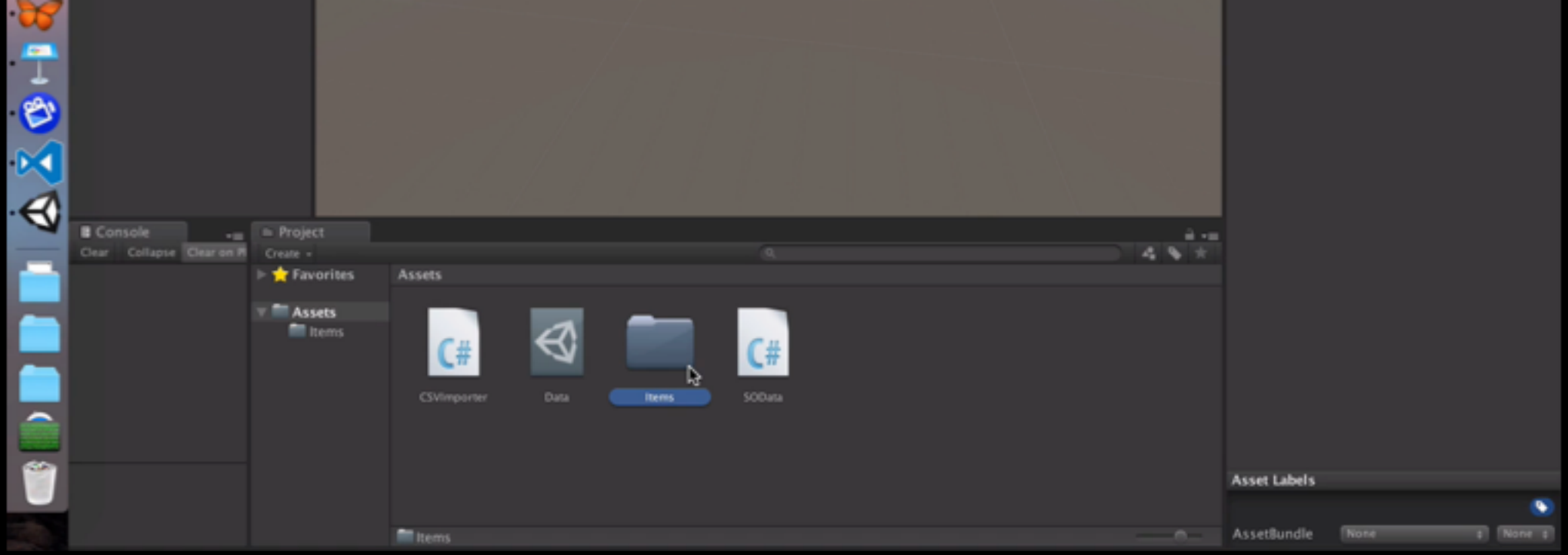



Unity
TempProject
TestProject
ピクチャ
アプリケー…
書類
Unitys
Monosnap
AirDrop
Data.item
サイズ
種類
Data.item 33 バイト 書類
SO5.mp4 500 KB MPEG-…ービー
SO4.mp4 188 KB MPEG-…ービー
SO3.mp4 839 KB MPEG-…ービー
SO2.mp4 1.8 MB MPEG-…ービー
so1.mp4 157 KB MPEG-…ービー
exp2.mp4 4.6 MB MPEG-…ービー
Errors
Tasks
Items
AssetBundle
None
None
ata.item
{"index":100,"name":"layer test"}


Unity File Edit Assets GameObject Component Window Help
100% U.S. 8:58
Unity 2017.2.0b11 (64bit) - Untitled - SOTest - PC, Mac & Linux Standalone <OpenGL 4.1>
Pivot Global
Collab Account Layers Layout
Hierarchy
Scene Game Asset Store Timeline
Shaded 2D Gizmos
Inspector
Items
Launchpad
Directional Light
Persp
Errors Tasks


Console
Clear
Collapse
Project
Create
Favorites
Assets
Items
CSVImporter
Items
SOData
Asset Labels
AssetBundle
None
None

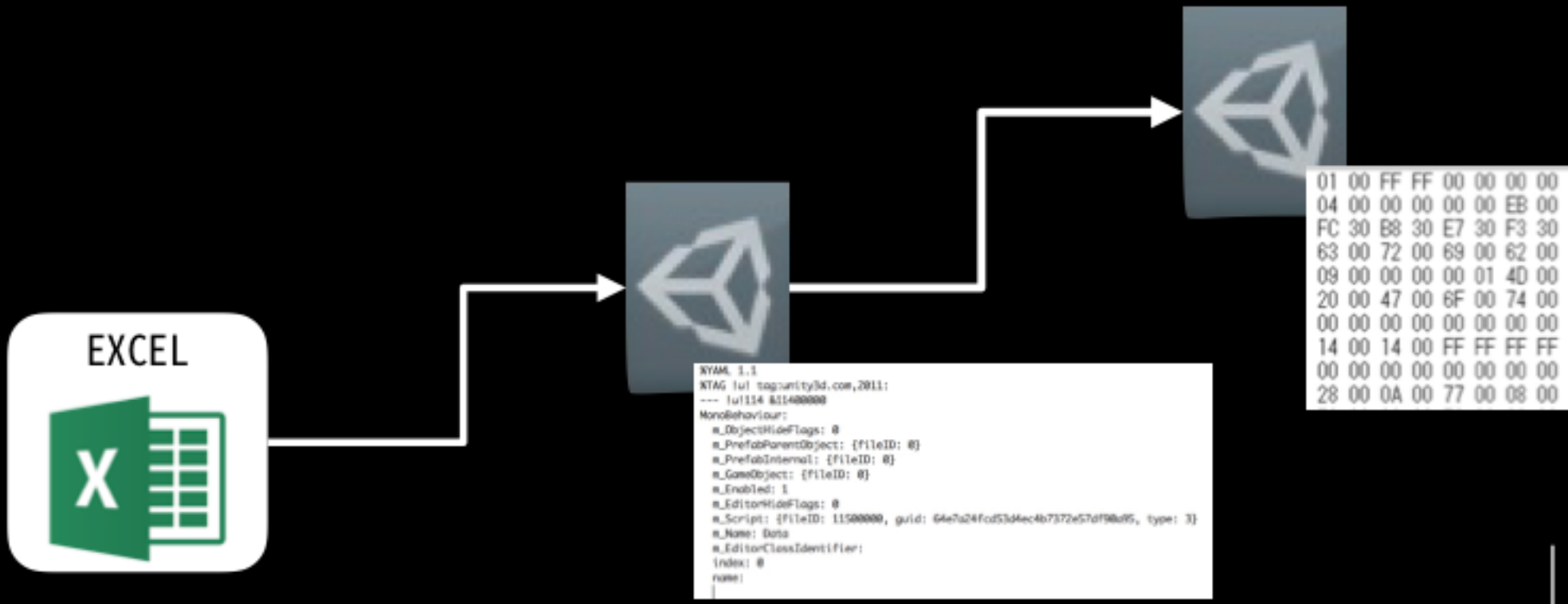

Unity
File
Edit
Assets
GameObject
Component
Window
Help
Unity 2017.2.0b11 (64bit) - Untitled - SOTest - PC, Mac & Linux Standalone <OpenGL 4.1>
Pivot
Global
Collab
Account
Layers
Layout
Hierarchy
Create
Untitled
Main Camera
Directional Light
Scene
Game
Asset Store
Timeline
Shaded
2D
Gizmos
Persp
Inspector
Items
Open
Console
Project

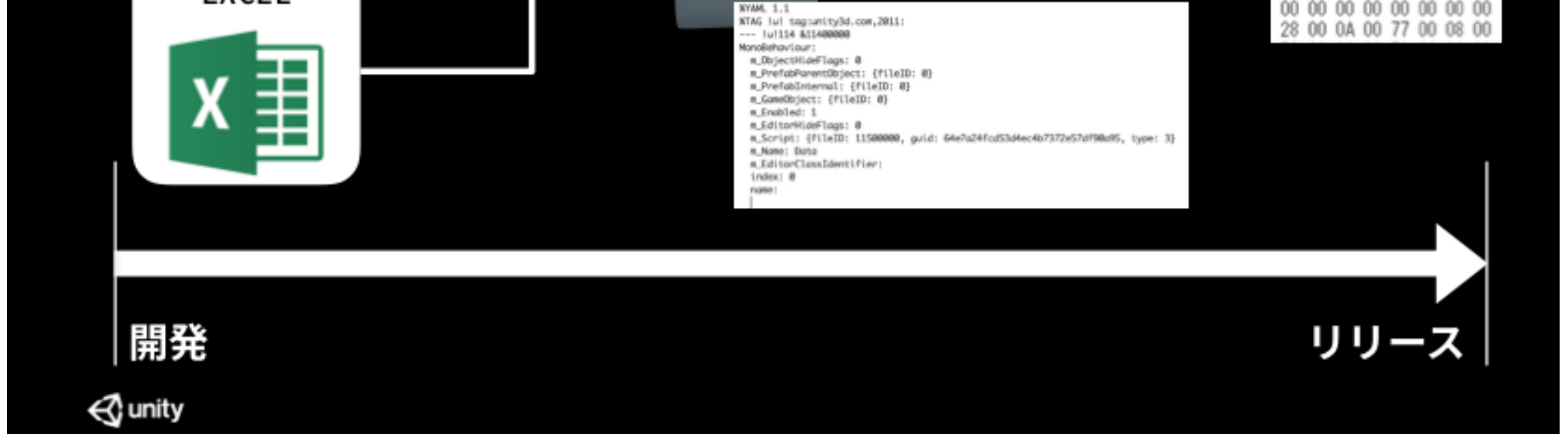



```
%YAML 1.1
%TAG !u! tag:unity3d.com,2011:
--- !u!114 &11400000
MonoBehaviour:
  m_ObjectHideFlags: 0
  m_PrefabParentObject: {fileID: 0}
  m_PrefabInternal: {fileID: 0}
  m_GameObject: {fileID: 0}
  m_Enabled: 1
  m_EditorHideFlags: 0
  m_Script: {fileID: 11500000, guid: 64e7a24fcd53d4ec4b7372e57df90a95, type: 3}
  m_Name: Data
  m_EditorClassIdentifier:
  index: 0
```

```
m_GameObject: {fileID: 0}
m_Enabled: 1
m_EditorHideFlags: 0
m_Script: {fileID: 11500000, guid: 64e7a24fcd53d4ec4b7372e57df90a95, type: 3}
m_Name: Data
m_EditorClassIdentifier:
index: 0
name:
```

●ScriptableObjectはエディターではYAMLフォーマット





ScriptableObjectの中身を独自にデシリアライズ

```
[CreateAssetMenu]
public class SampleDictionary : ScriptableObject, ISerializationCallbackReceiver
{
   const string path = "item.json";

   public Item item = new Item ();

   public void OnBeforeSerialize ()
   {
      var json = JsonUtility.ToJson (item);
      File.WriteAllText (path, json);
   }

   public void OnAfterDeserialize ()
   {
```

コールバックを受ける
インターフェース

シリアライズ前

デシリアライズ後

シリアライズ前

デシリアライズ後

```
public void OnBeforeSerialize ()
{
    var json = JsonUtility.ToJson (item);
    File.WriteAllText (path, json);
}

public void OnAfterDeserialize ()
{
    if (File.Exists (path)) {
        var json = File.ReadAllText (path);
        JsonUtility.FromJsonOverwrite (json, item);
    }
}
```

再生前に巻き戻る

●Scriptable Objectのリセット

アイディア

●ゲーム開始時にScriptable Objectをリセットする処理を追加

●シリアライズするデータとゲーム中に触るデータは分ける

●ゲーム再生終了時にScriptable Objectをアンロード

●ゲーム再生終了時にScriptable Objectをアンロード

ゲーム開始時にScriptable Objectをリセット

```
public abstract class ResettableScriptableObject : ScriptableObject
{
    public abstract void Reset ();
}
```

```
[CreateAssetMenu]
public class SaveData : ResettableScriptableObject
{
  public int count = 0;

    public override void Reset ()
    {
```

```
public class SaveData : ResettableScriptableObject
{
  public int count = 0;

    public override void Reset ()
    {
        //初期化コード
        count = 0;
   }
}
```

ゲーム開始時にScriptable Objectをリセット

```
public class DataResetter : MonoBehaviour
{
    public ResettableScriptableObject[] resettableScriptableObjects;

  private void Awake ()
    {
      for (int i = 0; i < resettableScriptableObjects.Length; i++)
      {
          resettableScriptableObjects[i].Reset ();
      }
```

```
    {
      for (int i = 0; i < resettableScriptableObjects.Length; i++)
      {
          resettableScriptableObjects[i].Reset ();
      }
  }
}
```

データを分ける

```
public abstract class ResettableScriptableObject : ScriptableObject
{
  [SerializeField] int _count = 0;
  public int count{get; set;}

  void OnEnable()
  {
    count = _count;
  }
}
```

```
    count = _count;
  }
}
```

アンロードする

```
public class ResettableScriptableObject : ScriptableObject
{
  protected virtual void OnEnable()
  {
#if UNITY_EDITOR
    if( EditorApplication.isPlayingOrWillChangePlaymode == true ){
      UnityEditor.EditorApplication.playModeStateChanged += (state) => {
        if( EditorApplication.isPlayingOrWillChangePlaymode == false) {
          Resources.UnloadAsset(this);
        }
      } ;
    }
```

```
        if( EditorApplication.isPlayingOrWillChangePlaymode == false) {
          Resources.UnloadAsset(this);
        }
      } ;
    }
#endif
  }
}
```

アンロードする

```
[CreateAssetMenu]
public class Config : ResettableScriptableObject
{
  public float _count;
}
```

Reset

- Resetメソッド呼び出しにより初期化
- 起動するシーン内にリセットを呼び出すコンポーネントが必要
- 実行中のの再初期化は比較的容易
- ゲーム再生時でも直接値を操作できる

データ分け

- アクセス時に初期化
- 実行中の再初期化は容易
- 実行中に値を編集したい場合は、特別なコードが必要になる

Unload

- 再生終了時に初期化
- 実行中の再初期化はシステムのリソース管理の理解が必要
- 再生前にプロジェクトのセーブが必要

比較的容易

●ゲーム再生時でも直接
値を操作できる

生存戦略

Scriptable Objectの
寿命とアンロード

ScriptableObjectの寿命とアンロード

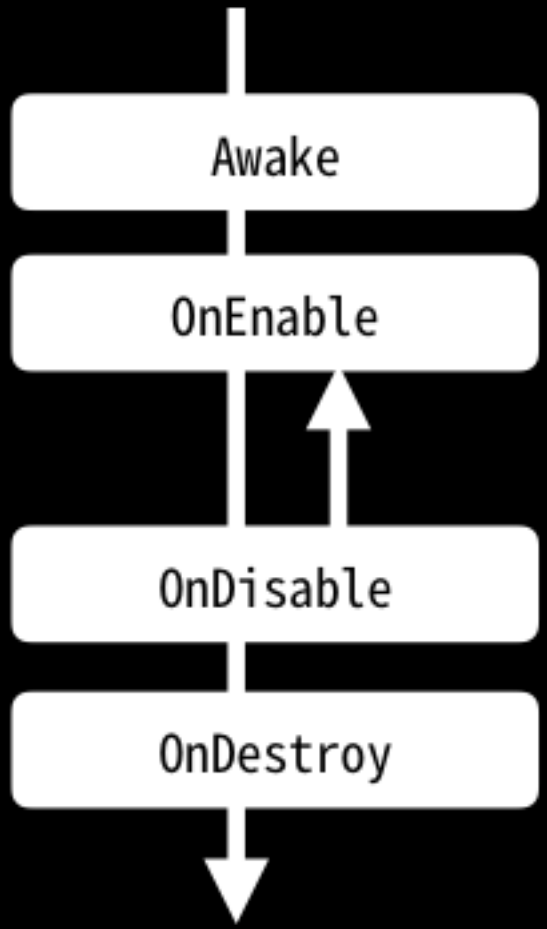
Unity

- ScriptableObjectをデシリアライズ
- インスタンスIDを付与
  （参照関係の解決に使用する）

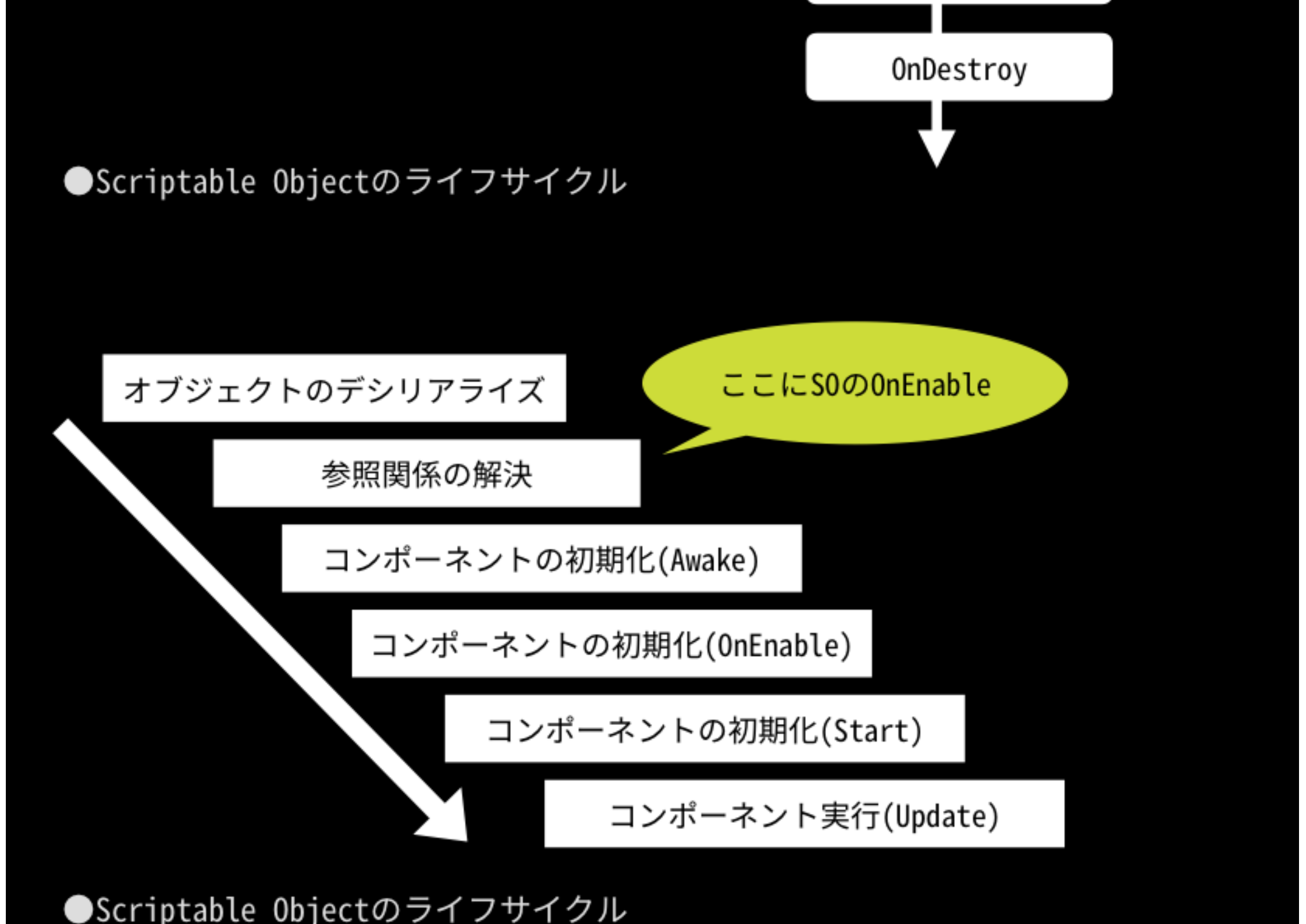
C#

- デシリアライズしたデータを
  スクリプトに流し込む
- コールバックの実体を実行
- 動作を実装

●UnityエンジンはC++、スクリプトはC#



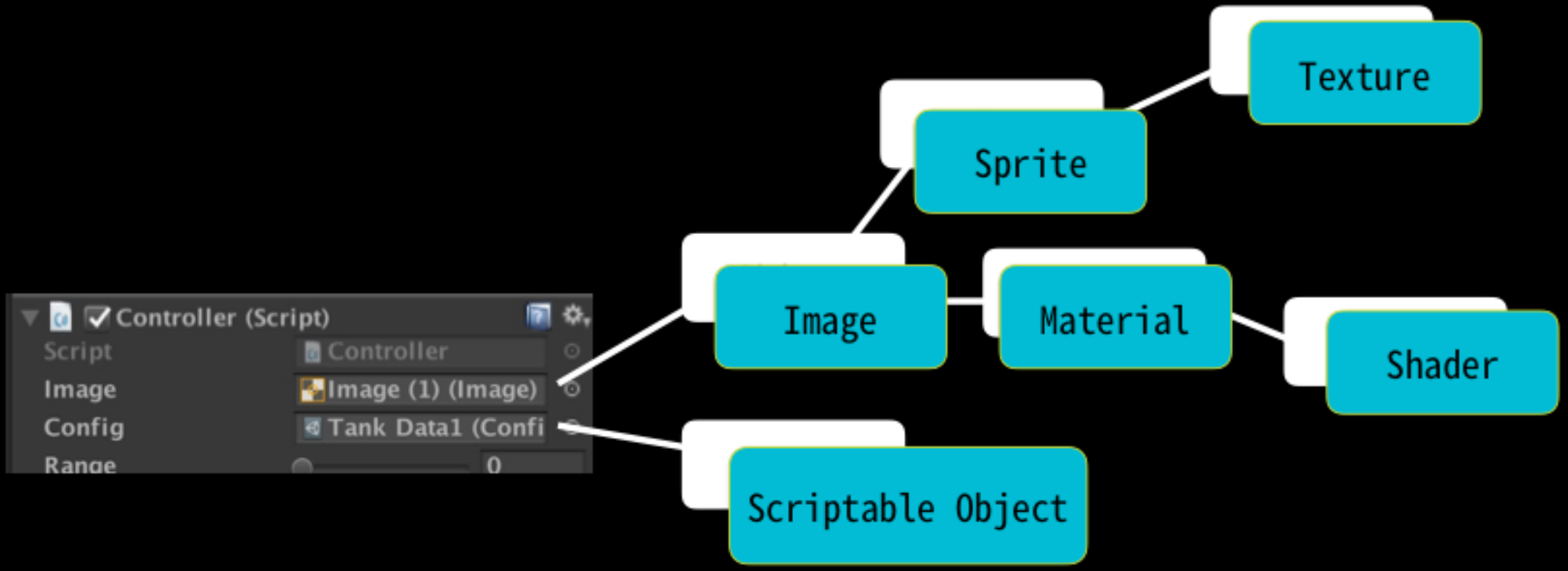
●ScriptableObjectのライフサイクル



●Scriptable Objectのライフサイクル

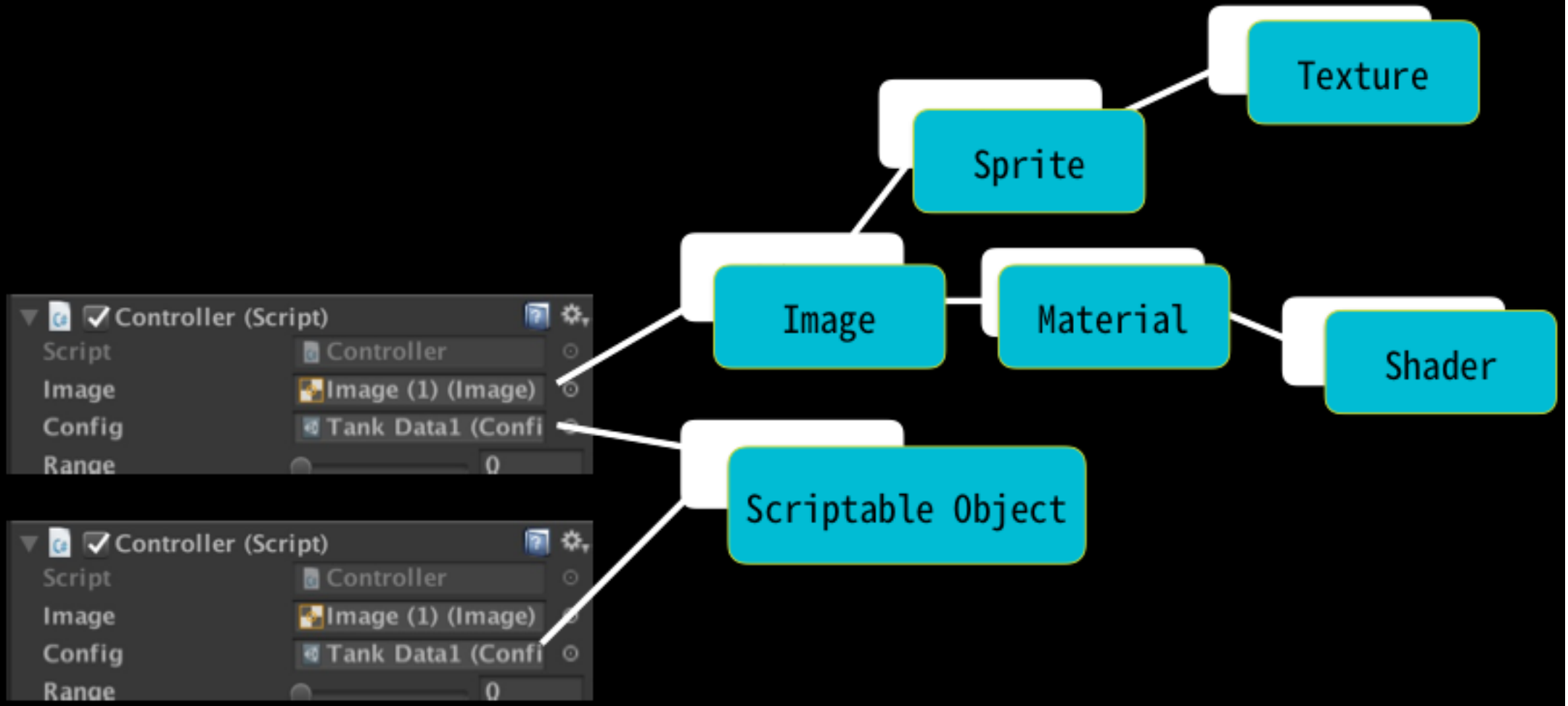
コンポーネント実行(Update)

●Scriptable Objectのライフサイクル

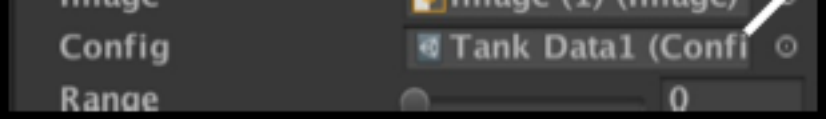


●参照されると関係するオブジェクトをロード

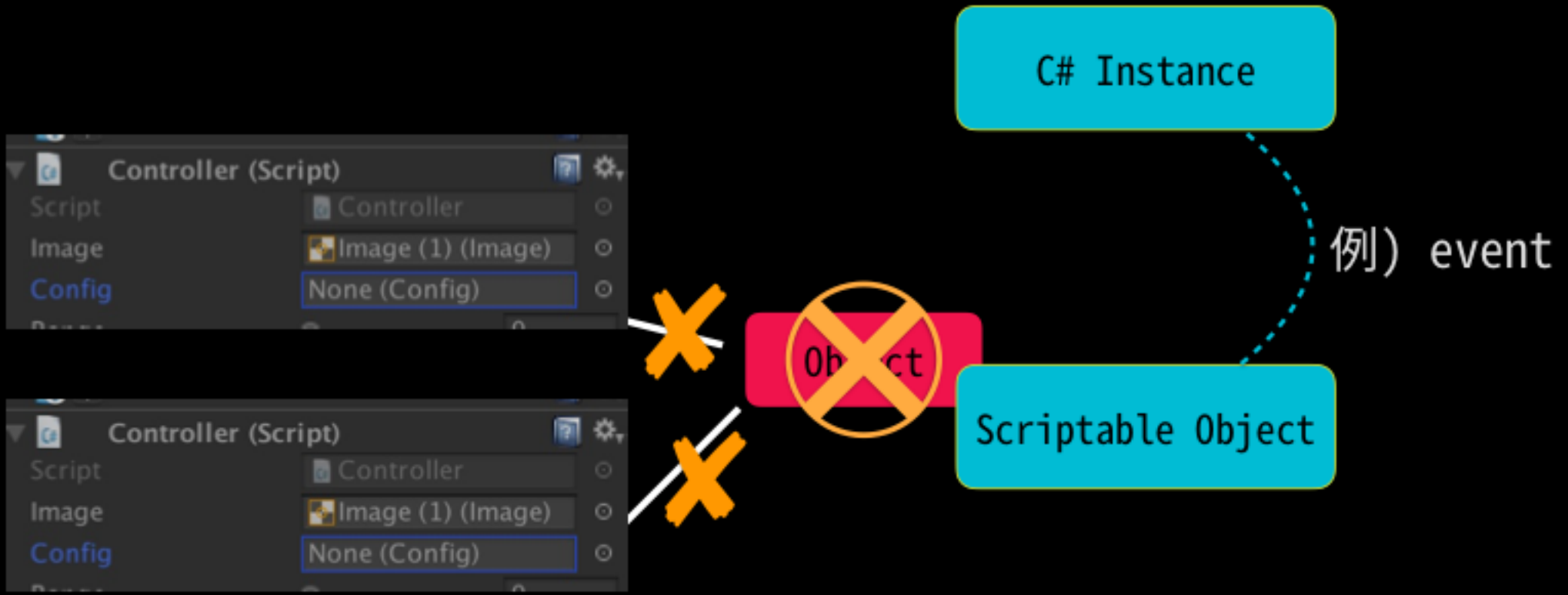
●参照されると関係するオブジェクトをロード

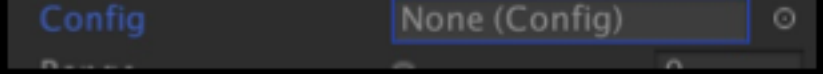


●テーブルが生きてるならば、同一のインスタンスにアクセスする



●テーブルが生きてるならば、同一のインスタンスにアクセスする



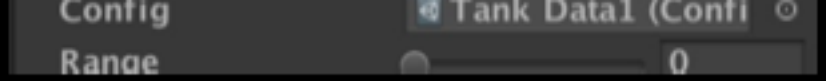


C# Instance

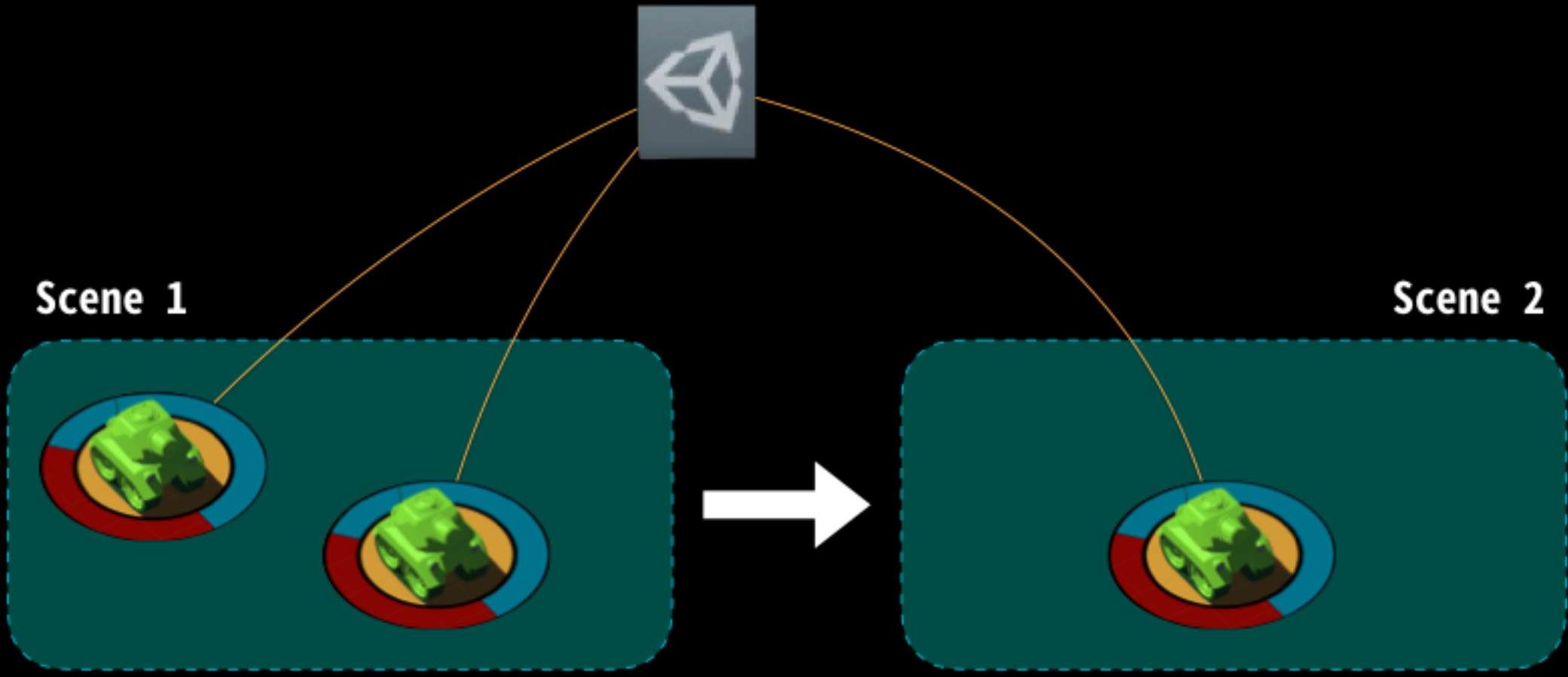
Scriptable Object

Scriptable Object

●再度Objectを作ると、新しく参照関係を構築する

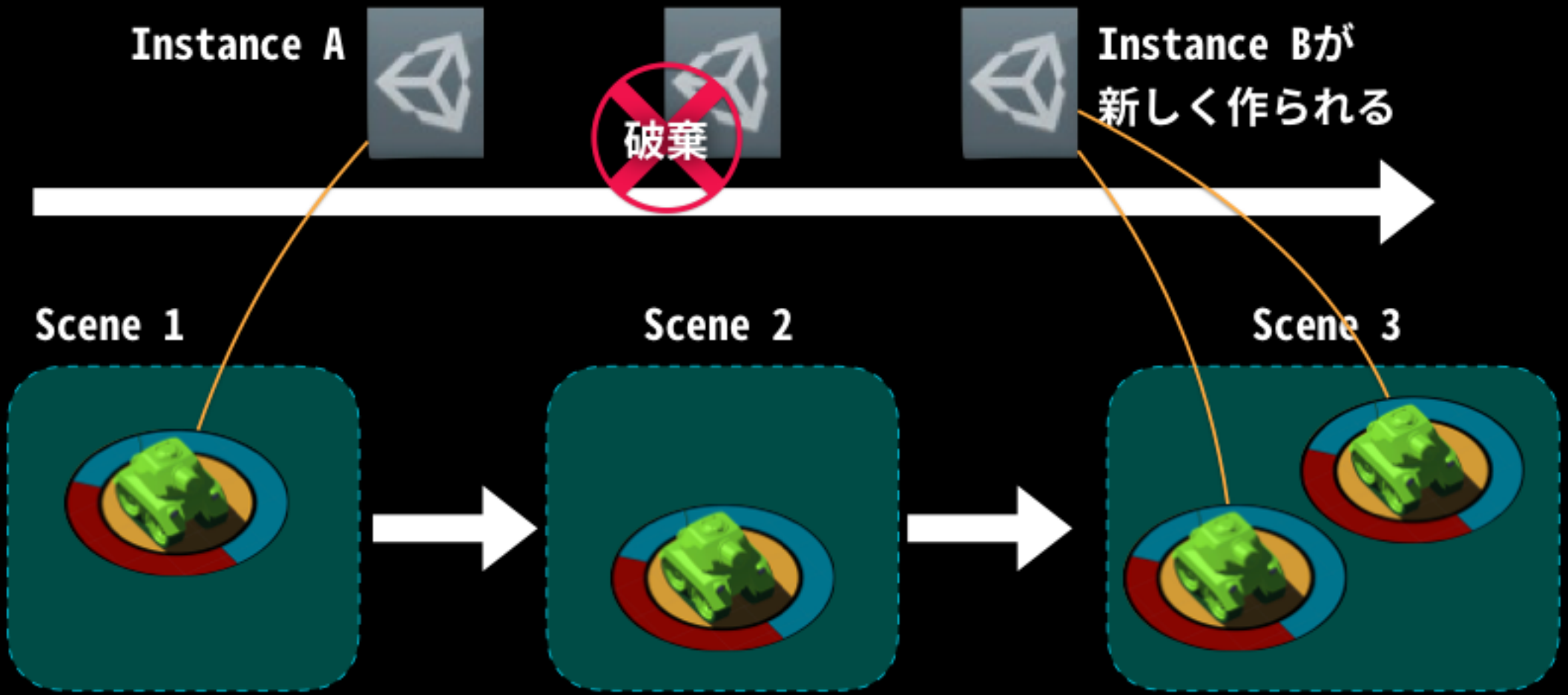


●再度Objectを作ると、新しく参照関係を構築する



●移動先のシーンでSOが参照されている場合、破棄されない

●移動先のシーンでSOが参照されている場合、破棄されない



●参照されていないシーンに移動すると、アセットは破棄され、
参照されると別インスタンスIDを持って再びロードされる

●参照されていないシーンに移動すると、アセットは破棄され、
参照されると別インスタンスIDを持って再びロードされる

明確に破棄したい場合

●Destroy

CreateInstanceで生成したインスタンスを破棄

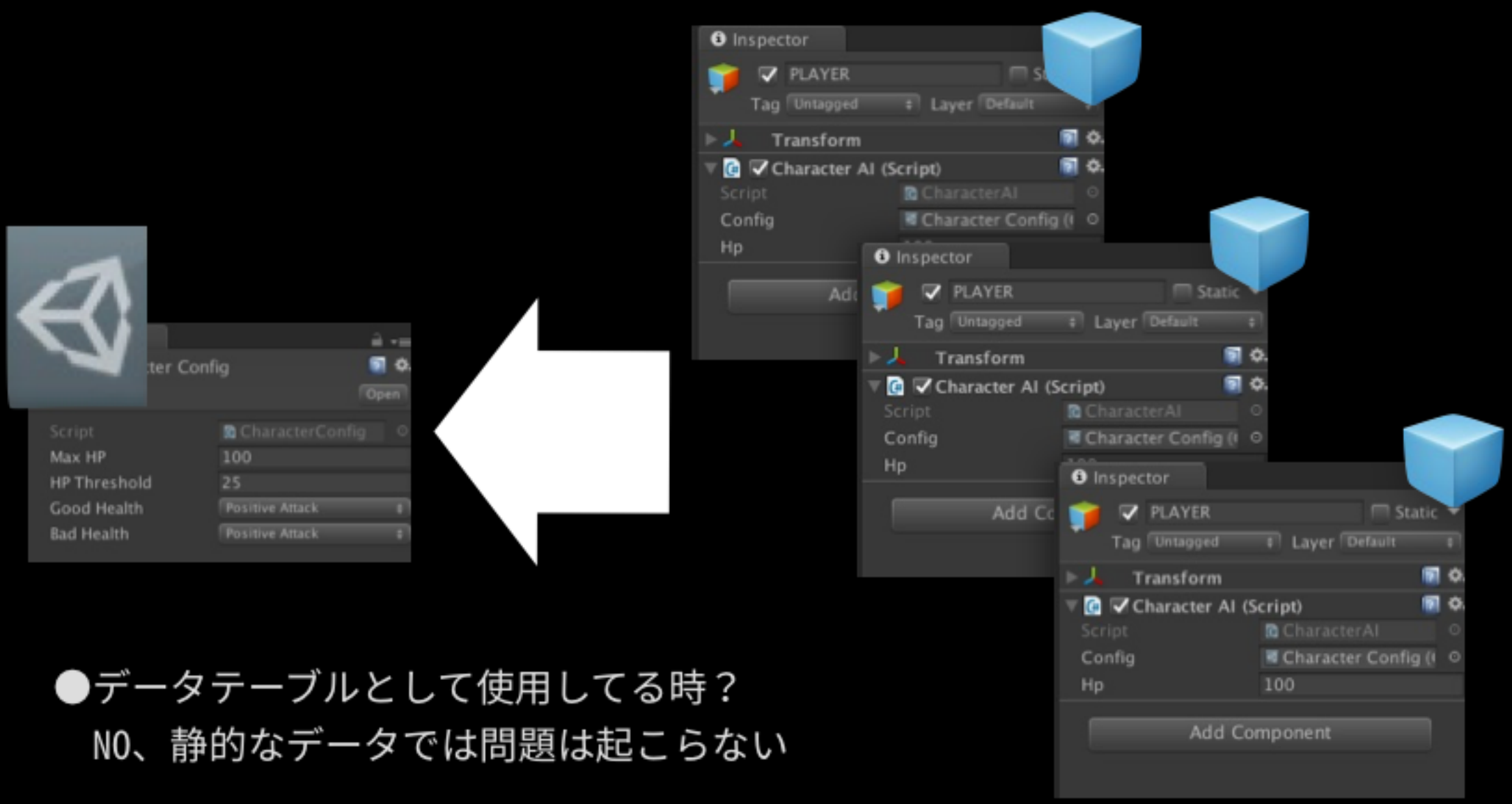
●UnloadUnusedAsset

被参照のアセットを開放

●Unload

指定したアセットを強制開放

どんな時に注意すべき？

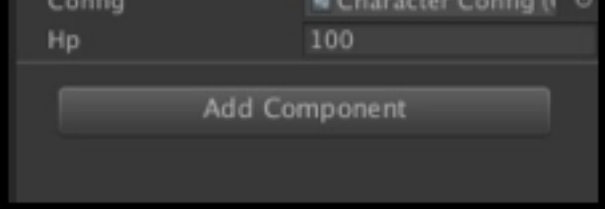


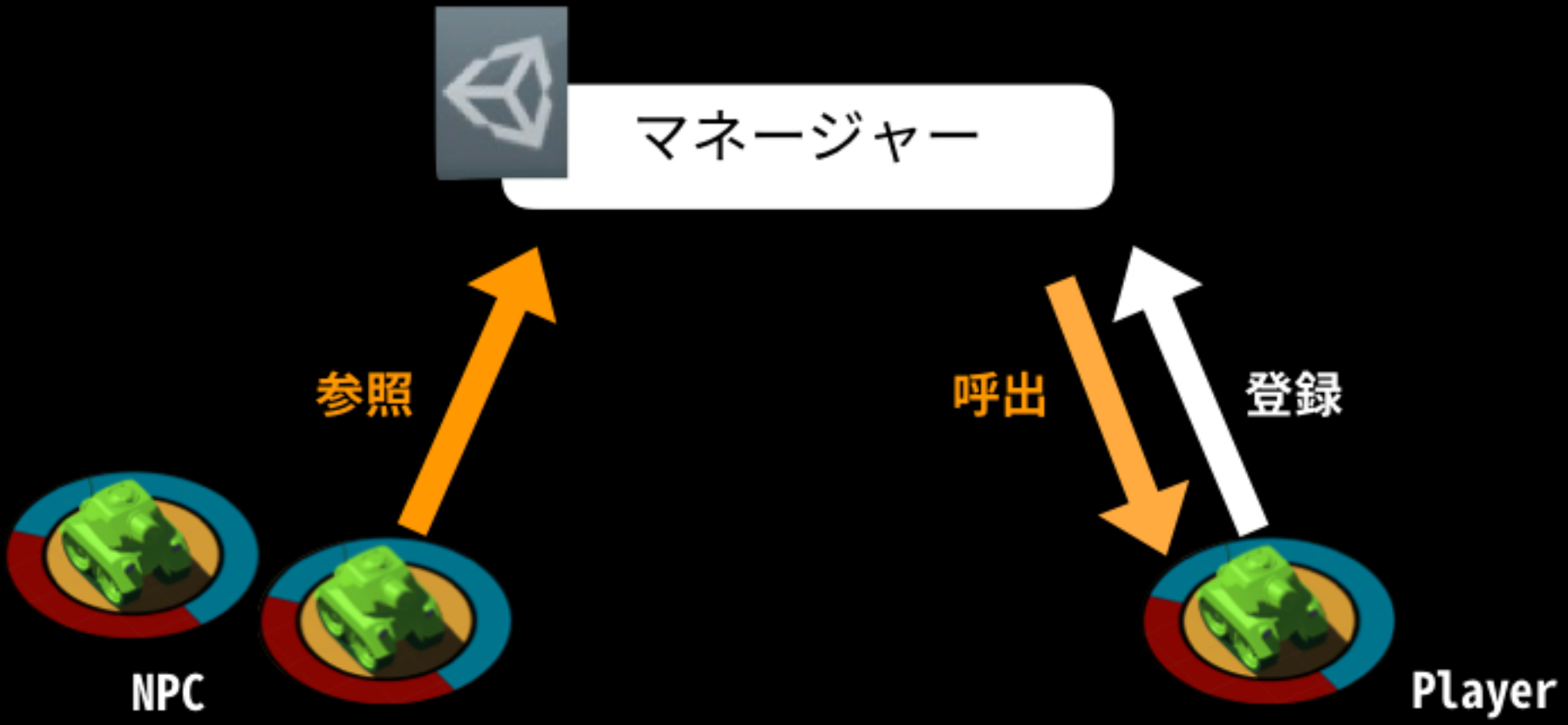
●データテーブルとして使用してる時？

NO、静的なデータでは問題は起こらない

●データテーブルとして使用してる時？

NO、静的なデータでは問題は起こらない



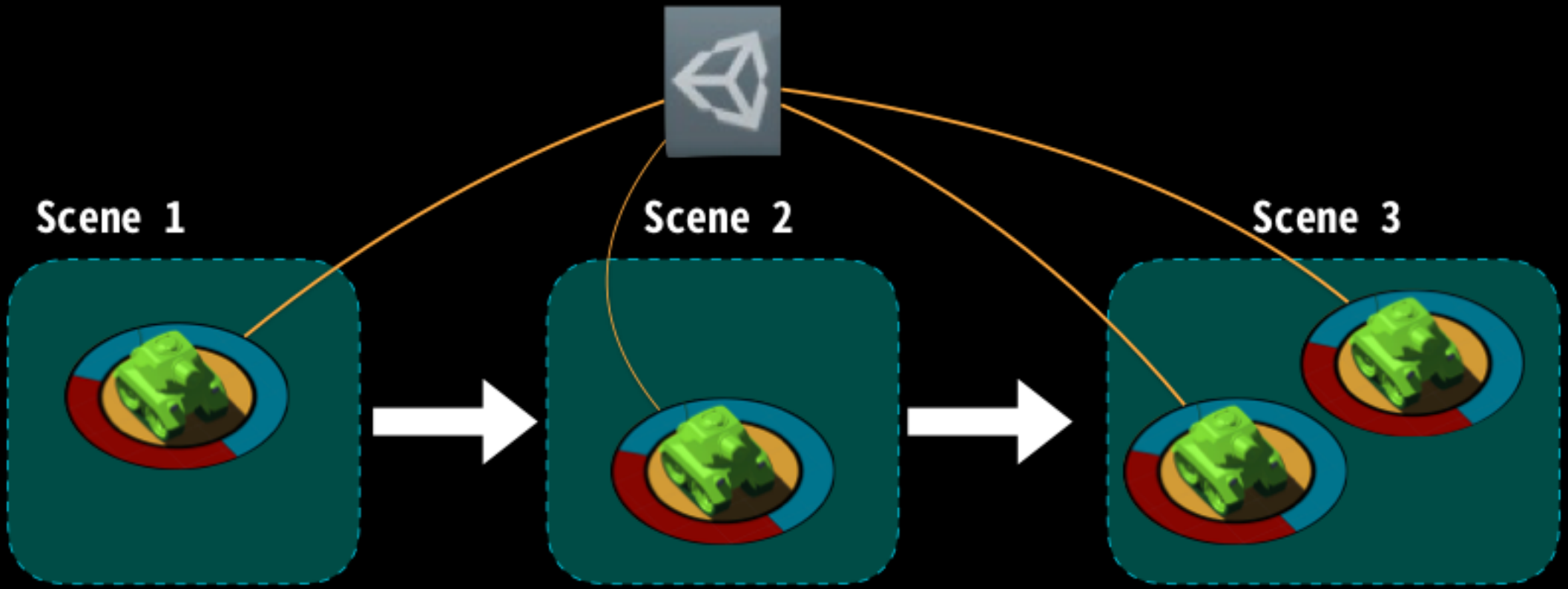


●オブジェクトのバインド？

Unloadで開放する場合は少し注意が必要。UnloadUnusedAssetsを使う場合はOK

●オブジェクトのバインド？

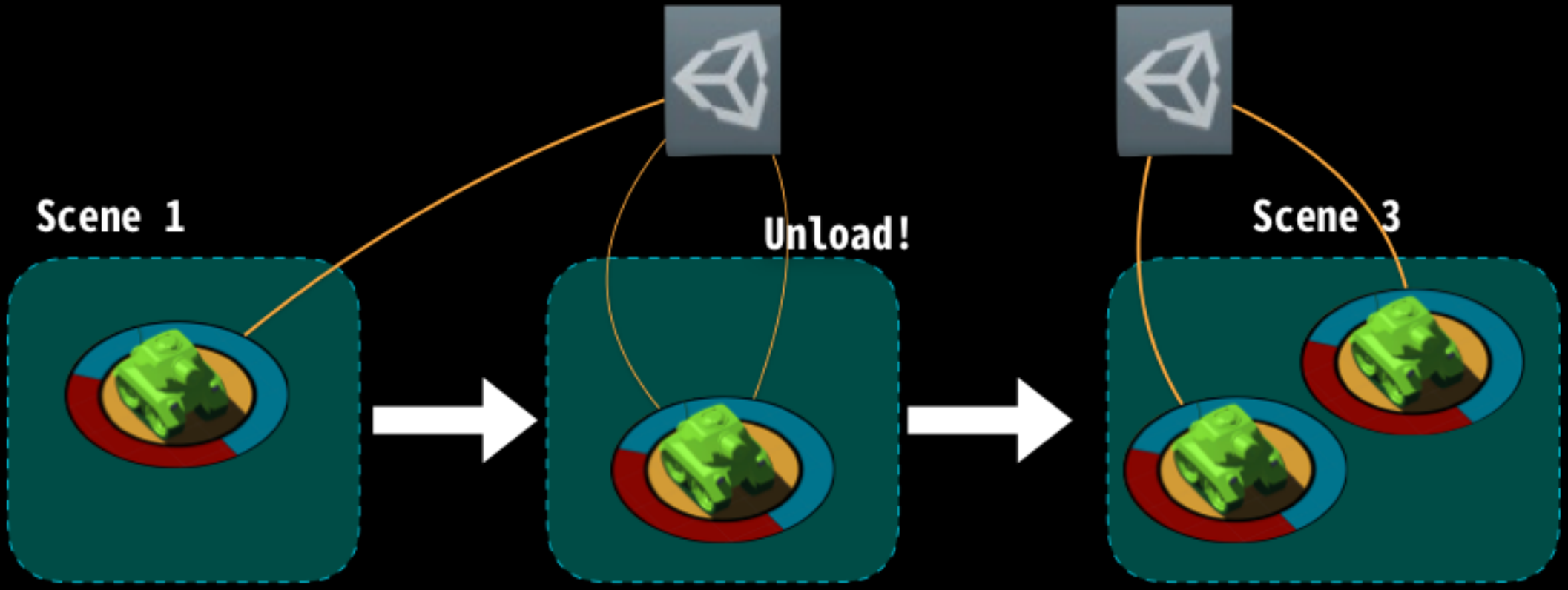
Unloadで開放する場合は少し注意が必要。UnloadUnusedAssetsを使う場合はOK



●ゲーム進行等を保持する場合

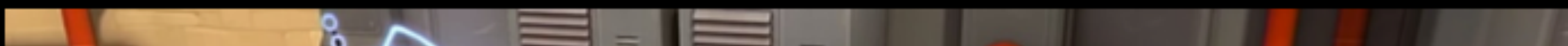
非参照の状況でインスタンスが破棄される可能性があるので注意

●ゲーム進行等を保持する場合
非参照の状況でインスタンスが破棄される可能性があるので注意



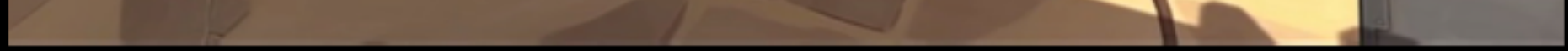
●逆に開放したい場合、シーン移行直前でSOをアンロードすると、
次のシーンでは新鮮なインスタンスにアクセスする

次のシーンでは新鮮なインスタンスにアクセスする



unity

unity



unity

Thank you