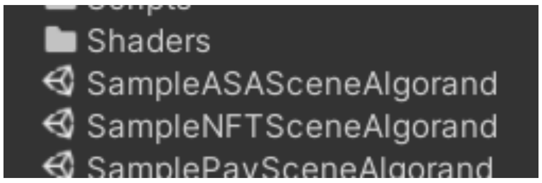
Introduction

In this tutorial we will talk about examples of using Assets in Unity.

Two samples scene:



Using an Algorand NFT and load from IPFS in Unity

Sample scene: *SampleNFTSceneAlgorand*

At this link IPFS URI there is very simple mesh in GLFT2.0 format (khronos.org/gltf/).

It is a Not Fungible Token in Algorand BlockChain.

See: https://goalseeker.purestake.io/algorand/testnet/asset/15942738

In this tutorial we use: GLTFUtility to load GLB/GLTF 2.0 mesh and render it in Unity.

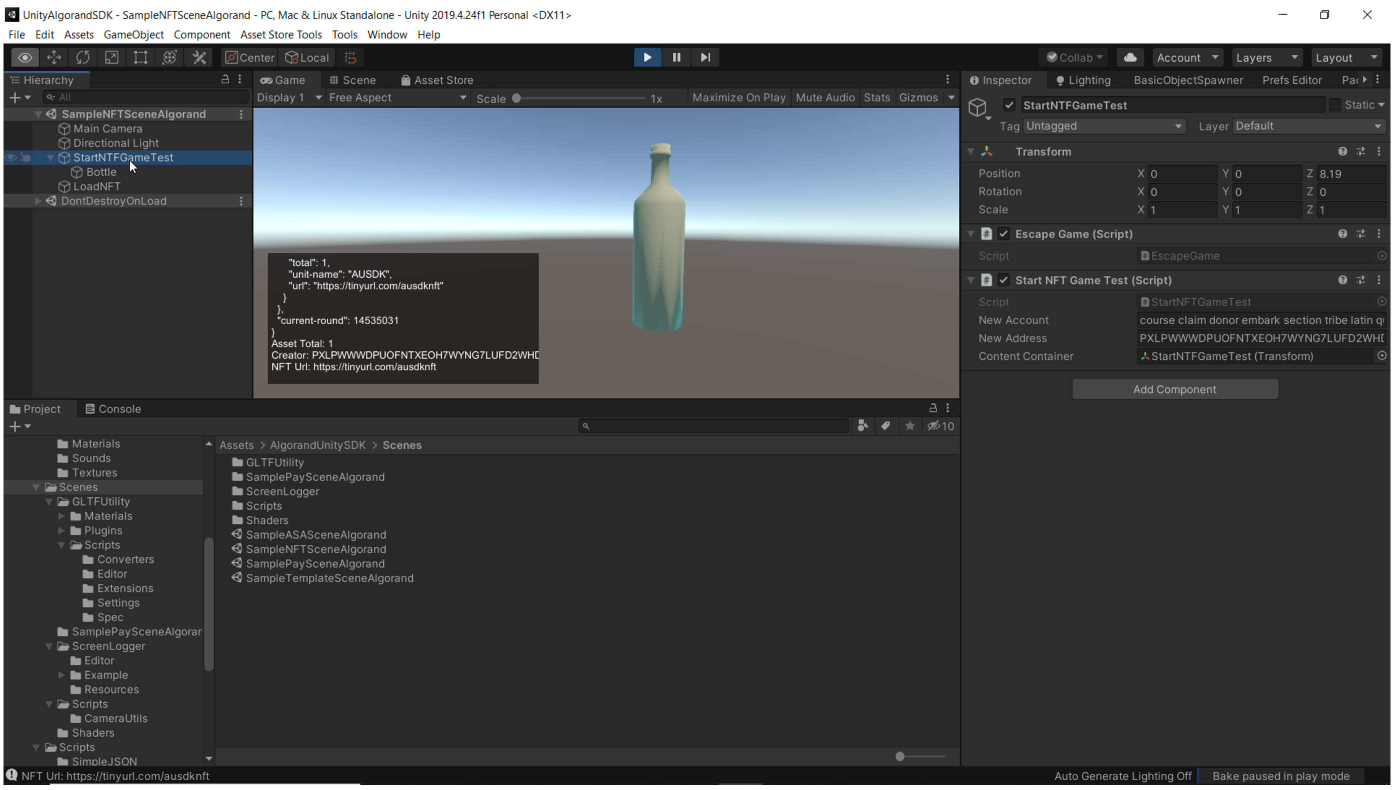
```
//Connect to Node
AlgorandManager.Instance.ConnectToNode(AlgorandManager.Instance.ALGOD_URL_ENDPOINT,
AlgorandManager.Instance.ALGOD_TOKEN);
//Get info About NFT Asset
var jsonResult =
AlgorandManager.Instance.GetAsset(AlgorandManager.Instance.ALGOD_URL_ENDPOINT_INDEXER,
    AlgorandManager.Instance.ALGOD_TOKEN,
    15942738);
//Debug.Log(jsonResult); //DEBUG
//https://wiki.unity3d.com/index.php/SimpleJSON
var N = JSON.Parse(jsonResult);
//Show Asset Total Example
Debug.Log("Asset Total: " + N["asset"]["params"]["total"]);
//Show Creator Example
Debug.Log("Creator: " + N["asset"]["params"]["creator"]);
//Get Asset Algorand NFT
var UrlToNFT = N["asset"]["params"]["url"];
```

To load using *UrlToNFT* variable we use Unity Coroutine:

```
IEnumerator LoadGLTFRoutine(string uri)
{
    UnityWebRequest webRequest = new UnityWebRequest(uri);
    webRequest.downloadHandler = new DownloadHandlerBuffer();
    yield return webRequest.SendWebRequest();
    if (webRequest.isNetworkError || webRequest.isHttpError)
    {
        Debug.Log(webRequest.error);
    }
    else
    {
        // Or retrieve results as binary data
        byte[] results = webRequest.downloadHandler.data;
        AnimationClip[] clips;
        GameObject result = Importer.LoadFromBytes(results, new ImportSettings() {
useLegacyClips = true }, out clips);
        result.transform.SetParent(contentContainer.transform, false);
    }
}
```

You can use normal asset trasfer method if you want sell it to specific Algorand Address.

The final result:


UnityAlgorandSDK - SampleNFTSceneAlgorand - PC, Mac & Linux Standalone - Unity 2019.4.24f1 Personal <DX11>

```
    "total": 1,
    "unit-name": "AUSDK",
    "url": "https://tinyurl.com/ausdknft"
   }
 },
 "current-round": 14535031
}
Asset Total: 1
Creator: PXLPWWWDPUOFNTXEOH7WYNG7LUFD2WHD
NFT Url: https://tinyurl.com/ausdknft
```

StartNTFGameTest — Tag: Untagged — Layer: Default

Transform — Position X 0 Y 0 Z 8.19; Rotation X 0 Y 0 Z 0; Scale X 1 Y 1 Z 1

Escape Game (Script) — Script: EscapeGame

Start NFT Game Test (Script)
- Script: StartNFTGameTest
- New Account: course claim donor embark section tribe latin q
- New Address: PXLPWWWDPUOFNTXEOH7WYNG7LUFD2WHD
- Content Container: StartNTFGameTest (Transform)

NFT Url: https://tinyurl.com/ausdknft

Transfert Algorand Assets in Unity

Sample scene: *SampleASASceneAlgorand*

Code analysis

Tha main code example is the script: *StartGameASATest.cs*

First it is checked if there is a "AlgorandAccountSDK" key and therefore it has been saved.

```
if (!PlayerPrefs.HasKey("AlgorandAccountSDK"))
```

If check is true, it will load and show in Debug.Log the account Address used and its mnemonic key in BIP 39 format.

```
//Load Algorand Account from encrypted PlayerPrefs
NewAddress = AlgorandManager.Instance.LoadAccountFromPlayerPrefs();
//Show Algorand Account Address
Debug.Log(NewAddress);
//Get Mnemonic Algorand Account Passphrase
Debug.Log(AlgorandManager.Instance.GetMnemonicPassphrase());
NewAccount = AlgorandManager.Instance.GetMnemonicPassphrase();
//Get Algorand Account Address from AlgorandManager Instances
Debug.Log(AlgorandManager.Instance.GetAddressAccount());
//Verify Algorand Account Address passed
Debug.Log("Valid Algorand Address: " +
AlgorandManager.Instance.AddressIsValid(NewAddress));
//Show URL ENDPOINT ALGOD
Debug.Log("URL ENDPOINT: " + AlgorandManager.Instance.ALGOD_URL_ENDPOINT);
//Show URL ENDPOINT INDEXER
Debug.Log("URL ENPOINT INDEXER: " +
AlgorandManager.Instance.ALGOD_URL_ENDPOINT_INDEXER);
//Show Token Used
Debug.Log("Token Used: " + AlgorandManager.Instance.ALGOD_TOKEN);
```

Next we will start the background query via UnityThreadQueue of Algorand addresses so as not to interrupt the execution of Unity's MainThread by modifying the Text Pro at runtime.

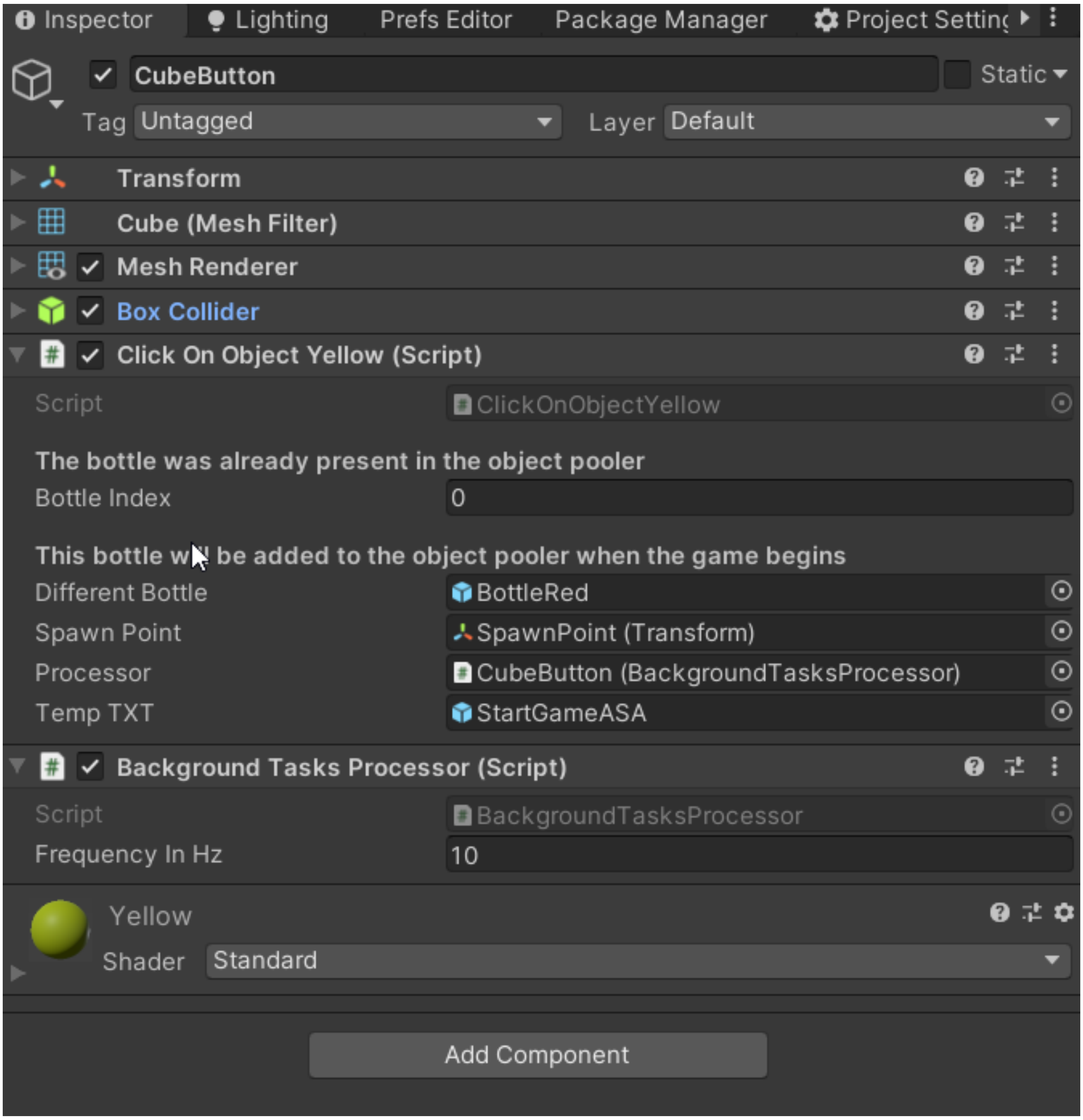
```
UnityThreadQueue.Instance.Enqueue(() =>
{
    //Green
    var jsonResult =
AlgorandManager.Instance.GetAccount(AlgorandManager.Instance.ALGOD_URL_ENDPOINT_INDEXER
,
    AlgorandManager.Instance.ALGOD_TOKEN,
"PVT67ZSBADU5ATXRIYBRIDBWSOIJOJJR73FJPCUFSKPHXI4M7PIRS5SRRI");
    //Debug.Log(jsonResult);
    var N = JSON.Parse(jsonResult);
    //Show Asset Total Example
    Debug.Log("Total Amount for user Green: "+N["account"]["assets"][6]["amount"]);
    AmountGreen = N["account"]["assets"][6]["amount"];
    //PAUSE for PureStake limitation
    Thread.Sleep(2000);
    //Yellow
    jsonResult =
AlgorandManager.Instance.GetAccount(AlgorandManager.Instance.ALGOD_URL_ENDPOINT_INDEXER
,
    AlgorandManager.Instance.ALGOD_TOKEN,
"F52PF5E2GNMUZN2JYPXS4ANMXUY23F6RVE6VEJH4ZZYHMDUZPYUFKWYX6Q");
    N = JSON.Parse(jsonResult);
    //Show Asset Total Example
    Debug.Log("Total Amount for user Yellow: "+N["account"]["assets"][3]["amount"]);
    AmountYellow = N["account"]["assets"][3]["amount"];
});
```

We have two cube: one yellow and one green.

If you click with right mouse button on one of cubes a new "assets" red bottle wil be created and 1 ASA Algorand will be trasfert to Yellow Address User or Green Address User.

See ASA on TESTNET: https://goalseeker.purestake.io/algorand/testnet/asset/16038175

This is the configuration on Inspector using PoolObject like: https://github.com/Rfrixy/Generic-Unity-Object-Pooler



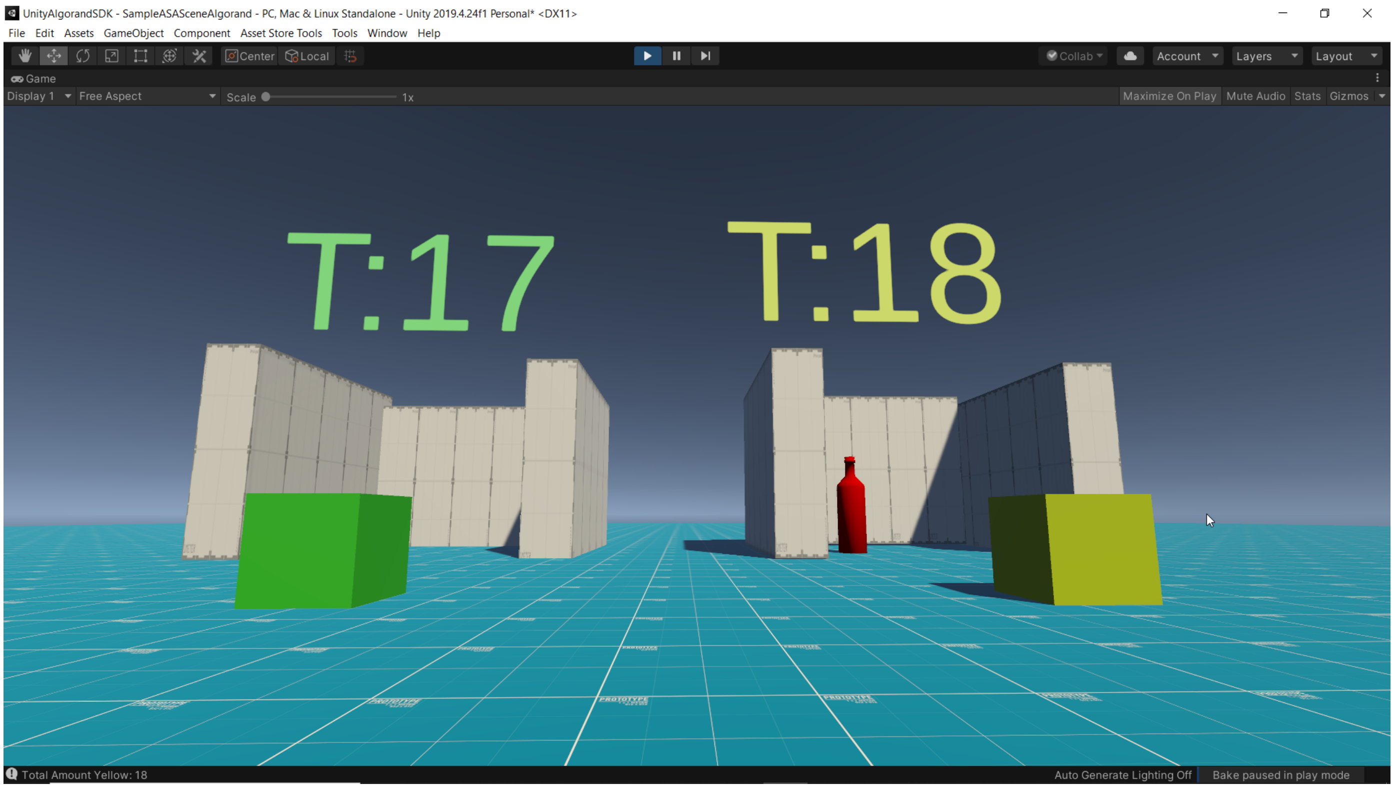
CubeButton
Tag Untagged Layer Default
Transform
Cube (Mesh Filter)
Mesh Renderer
Box Collider
Click On Object Yellow (Script)
Script ClickOnObjectYellow
The bottle was already present in the object pooler
Bottle Index 0
This bottle will be added to the object pooler when the game begins
Different Bottle BottleRed
Spawn Point SpawnPoint (Transform)
Processor CubeButton (BackgroundTasksProcessor)
Temp TXT StartGameASA
Background Tasks Processor (Script)
Script BackgroundTasksProcessor
Frequency In Hz 10
Yellow
Shader Standard

It is very important to understand that you need a system that starts the calls to the various Algorand services in the background and in completely asynchronous mode so as not to interrupt and create problems in the execution of the MainThread of Unity. For this purpose use *BackGroundTasksProcessor.cs* script.

```
void OnMouseDown()
{
    Debug.Log("Transfer 1 ASA to Yellow User");
    GameObject bottle = OP.GetPooledObject(bottleIndex);
    bottle.transform.rotation = SpawnPoint.transform.rotation;
    //float xPos = Random.Range(-5f, 5f);
    //float zPos = Random.Range(-5f, 5f);
    bottle.transform.position = SpawnPoint.transform.position;
```

```
     bottle.SetActive(true);
     //Transfert ASA
     Processor.Process(
             () =>
             {
                 //Do here Algorand ASA Transaction
                 TxIDPayment = AlgorandManager.Instance.AssetTransfer(
                 AlgorandManager.Instance.ALGOD_URL_ENDPOINT,
                 AlgorandManager.Instance.ALGOD_TOKEN,
                 "F52PF5E2GNMUZN2JYPXS4ANMXUY23F6RVE6VEJH4ZZYHMDUZPYUFKWYX6Q",
                 1,
                 16038175,
                 "Test Tx for Asset Transfert: " + System.DateTime.Now.ToString());
                 Debug.Log("TxID: " + TxIDPayment);
                 //PAUSE for PureStake limitation
                 Thread.Sleep(4000);
                 //Update Balance
                 var jsonResult =
AlgorandManager.Instance.GetAccount(AlgorandManager.Instance.ALGOD_URL_ENDPOINT_INDEXER
,
                 AlgorandManager.Instance.ALGOD_TOKEN,
"F52PF5E2GNMUZN2JYPXS4ANMXUY23F6RVE6VEJH4ZZYHMDUZPYUFKWYX6Q");
                 var N = JSON.Parse(jsonResult);
                 //Show Asset Total Example
                 Debug.Log("Total Amount for user Yellow: " + N["account"]["assets"][3]
["amount"]);
                 Amount = N["account"]["assets"][3]["amount"];
                 return Amount;
             },
             (result) =>
             {
                 Debug.Log("Total Amount Yellow: " + result);
                 //Update Blue Balance
                 tempTXT.GetComponent<StartGameASATest>().AmountYellow = Amount;
             }
         );
}
```

The final result:


UnityAlgorandSDK - SampleASASceneAlgorand - PC, Mac & Linux Standalone - Unity 2019.4.24f1 Personal* <DX11>
File Edit Assets GameObject Component Asset Store Tools Tools Window Help
Center Local
Collab Account Layers Layout
Game
Display 1 Free Aspect Scale 1x
Maximize On Play Mute Audio Stats Gizmos
T:17
T:18
Auto Generate Lighting Off
Bake paused in play mode

Total Amount Yellow: 18