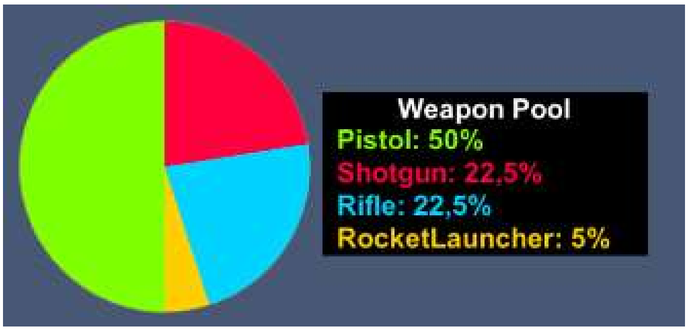
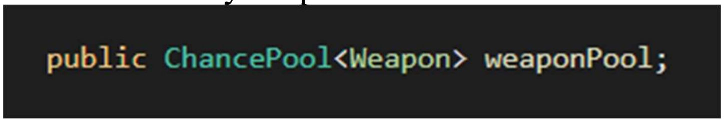
Martysh’s

# Chance Selection Pool



## Quick start.

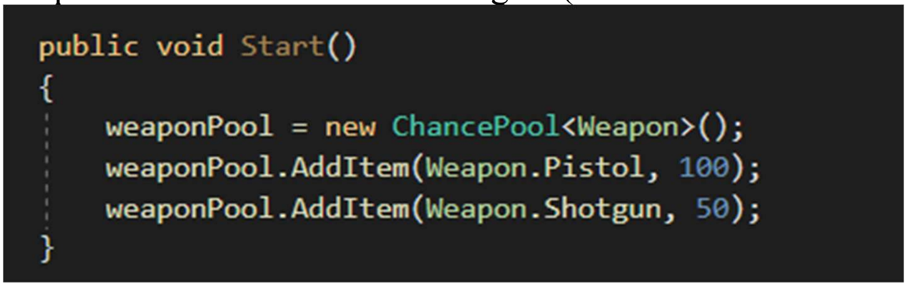
1. First of all, define a variable for your pool:



```
public ChancePool<Weapon> weaponPool;
```

Instead of ***<Weapon>***, there can be any ***gameObject, color, material, vector3, class, struct, enum*** etc – everything you can select from. For my example in documentation, I just chose a ***weapon*** as an entity. which is often found in computer games.

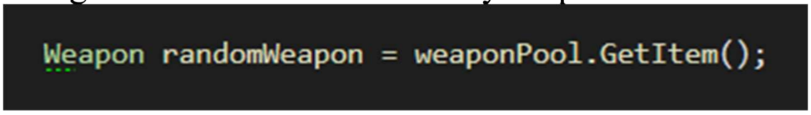
2. Then fill your pool with items with their weights (chance to be selected from pool).



```
public void Start()
{
    weaponPool = new ChancePool<Weapon>();
    weaponPool.AddItem(Weapon.Pistol, 100);
    weaponPool.AddItem(Weapon.Shotgun, 50);
}
```

The more weight you put on things, the more likely it will be selected from the pool. In the example above, a pistol has twice more chance to be selected versus a shotgun.

3. When you need to get some random item from your pool use this method:

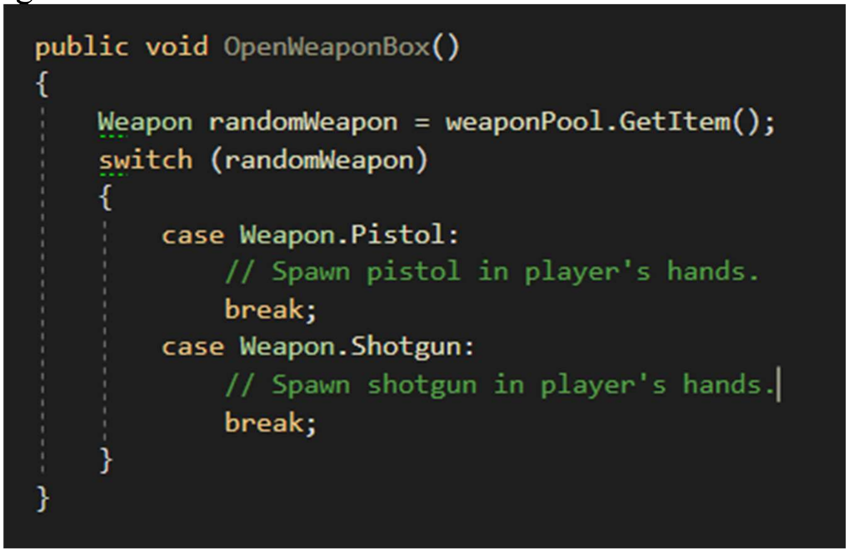


```
Weapon randomWeapon = weaponPool.GetItem();
```

And then use the resulting value as you need it. For example, spawn an object.

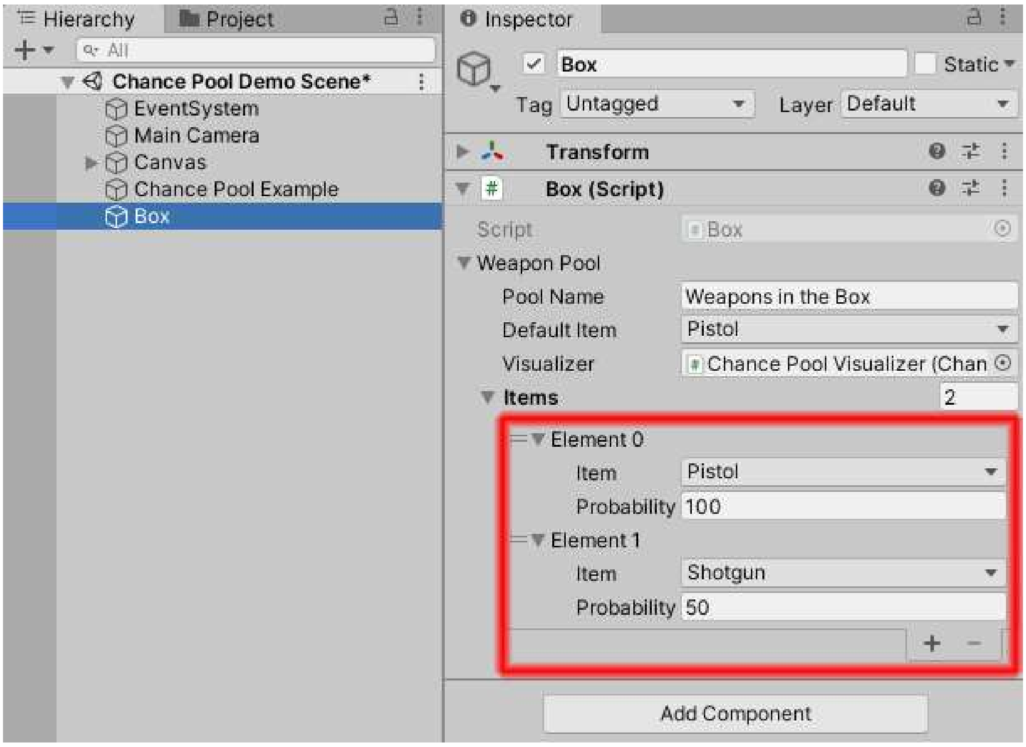
# <u>Managing your pool.</u>

Example of using result value:



```
public void OpenWeaponBox()
{
    Weapon randomWeapon = weaponPool.GetItem();
    switch (randomWeapon)
    {
        case Weapon.Pistol:
            // Spawn pistol in player's hands.
            break;
        case Weapon.Shotgun:
            // Spawn shotgun in player's hands.
            break;
    }
}
```

Also, you can modify your pool straight in inspector: add or remove items, change their weights etc:



☑ Box □ Static

Tag Untagged Layer Default

Script Box

Weapon Pool

Pool Name Weapons in the Box

Default Item Pistol

Visualizer Chance Pool Visualizer (Chan

Items 2

Element 0

Item Pistol

Probability 100

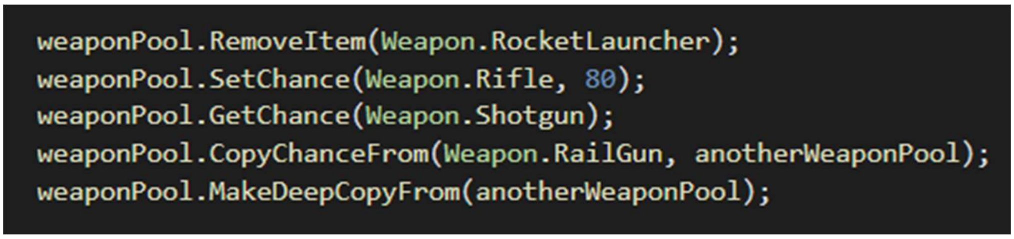
Element 1

Item Shotgun

Probability 50

* Warning: in Unity 2019 (and older) you can’t see ***chance pool*** in inspector, because only Unity 2020 (and newer) allow see generic lists in inspector.

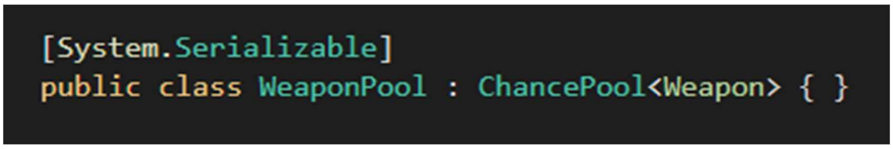
You can manage the pool using these and other methods. Each method is well documented.



```
weaponPool.RemoveItem(Weapon.RocketLauncher);
weaponPool.SetChance(Weapon.Rifle, 80);
weaponPool.GetChance(Weapon.Shotgun);
weaponPool.CopyChanceFrom(Weapon.RailGun, anotherWeaponPool);
weaponPool.MakeDeepCopyFrom(anotherWeaponPool);
```

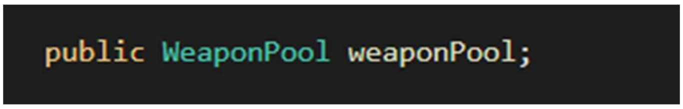
## Show pool in inspector in Unity 2019 and older.

If you use Unity 2019 (or older) and want to see ***chance pool*** in inspector, you need to make one additional intermediate step before using the pool. Create a new class that inherits from the ***ChancePool<T>*** with an explicit data type. Add ***[System.Serializable]*** attribute to it:



```
[System.Serializable]
public class WeaponPool : ChancePool<Weapon> { }
```

And then make a variable for new class:



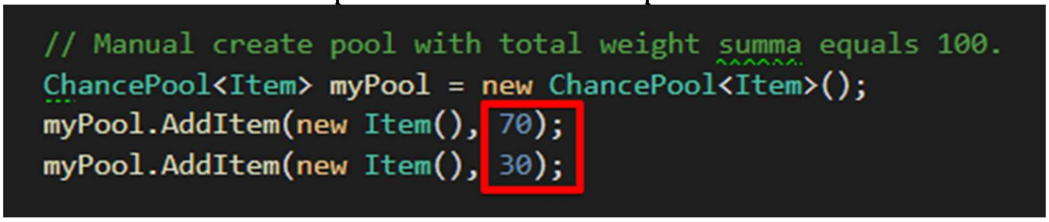
```
public WeaponPool weaponPool;
```

Then, use you pool as usual. You can watch it now in inspector in Unity 2019 (or older).

## Best practices.

Set the chance of the most common item in your pool = 100. Then, for items that should appear more often, set the higher weight. For example, if you want some item to appear 2 times more often, set its weight to 200. If you want it to appear 2 times less often, set its weight to 50.

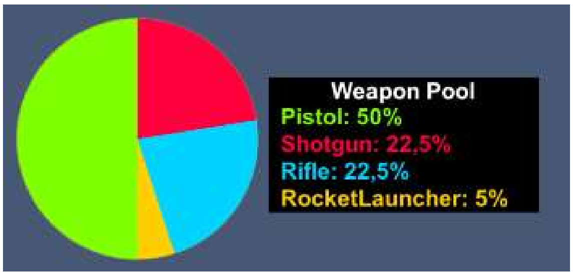
If at some point you want to know what percentage of the probability each item in the pool is out of 100%, use the method ***NormalizePool().*** But do not try to keep the total weight of 100 after adding things or changing their probability. You will not succeed. As practice shows, this quickly gets out of control. If you definitely want the summa of all item weights in the pool equals 100, create a new pool and manually enter all the values, making sure that their sum is equal to 100. For example:



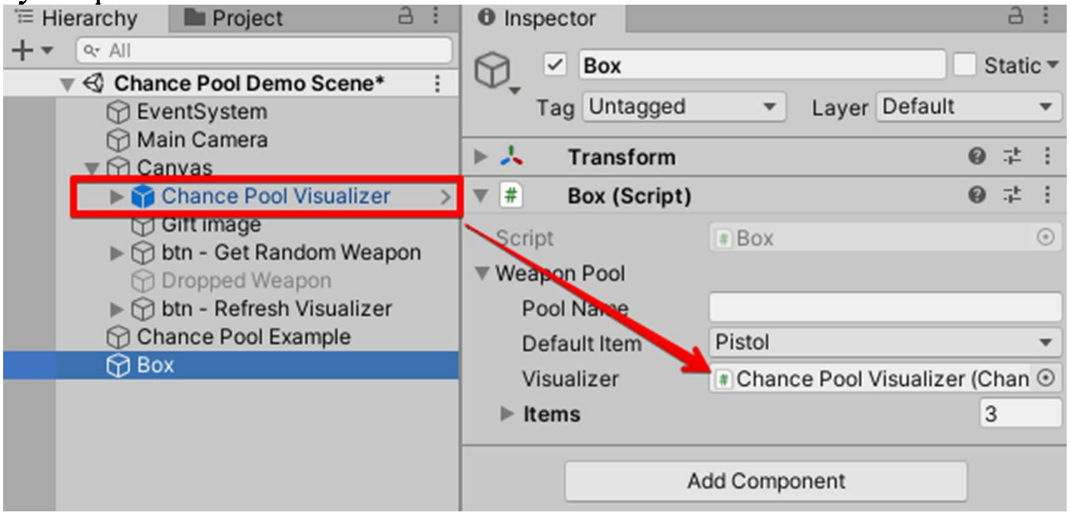
```
// Manual create pool with total weight summa equals 100.
ChancePool<Item> myPool = new ChancePool<Item>();
myPool.AddItem(new Item(), 70);
myPool.AddItem(new Item(), 30);
```

## Pool visualizer.

If you want to monitor the pool state in real time, use ***Chance Pool Visualizer***. It is just some little UI element. You can find its prefab in prefab folder. Just grab it and drag to any canvas in your scene.



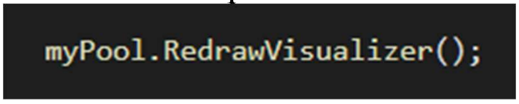
Then link it with your pool by dragging it from Hierarchy to special field in script that contains your pool.



Box ☑ □ Static
Tag Untagged Layer Default

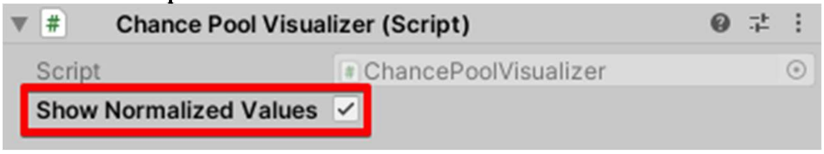
Script Box
Weapon Pool
Pool Name
Default Item Pistol
Visualizer Chance Pool Visualizer (Chan
Items 3

Then, in code, when you make some changes to your pool, use method ***RedrawVisualizer()*** to refresh its visual representation in UI:



```
myPool.RedrawVisualizer();
```

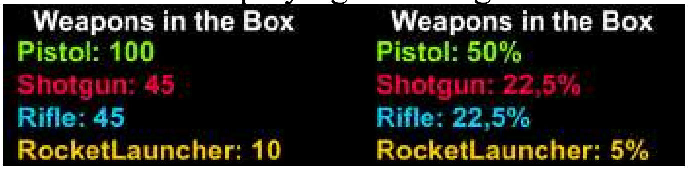
You can change the way of display the information by checking the "***Show normalized values***" checkbox in the inspector:



Script ChancePoolVisualizer

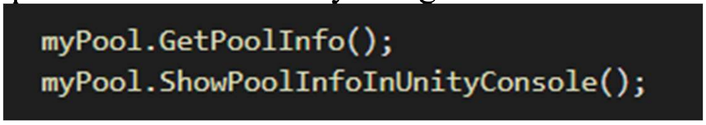
Show Normalized Values ☑

This way you can switch between displaying raw weights and normalized values:


Weapons in the Box
Pistol: 100
Shotgun: 45
Rifle: 45
RocketLauncher: 10

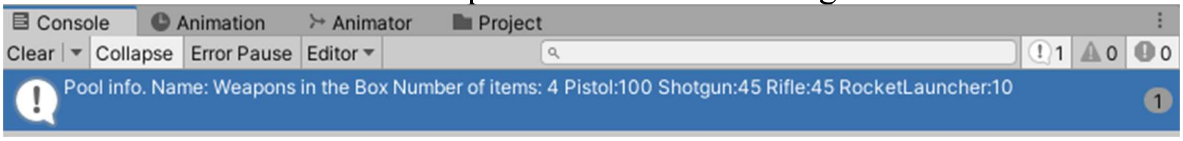
Weapons in the Box
Pistol: 50%
Shotgun: 22,5%
Rifle: 22,5%
RocketLauncher: 5%

Also, you can watch pool data in console by using such methods:



```
myPool.GetPoolInfo();
myPool.ShowPoolInfoInUnityConsole();
```

You will see full information about pool items and their weights.



```
Pool info. Name: Weapons in the Box Number of items: 4 Pistol:100 Shotgun:45 Rifle:45 RocketLauncher:10
```

## Contact information.

If you have any questions, feel free to contact me at: **martysh@yahoo.com**