Lists

Lists are like arrays but on steroids. They are containers but can be any variable type. You can even have an array of lists. Lists are dynamic. In other words they can grow and shrink.

To declare a list you do the following:

List <int> numbers;

To initialize a list you do the following

List<int> numbers = new List<int>();

You can then use . commands to manipulate the contents of the list.

Example: numbers.add (2); will add the number 2 to the list.

Numbers.Clear(); would remove all the contents of the list

There are a few other .commands but I will move on.

With a list you can sort through them like an array. A good way to do this is a for each loop.

Below is an example from my game that calculates the closest enemy out of the list of enemy’s.

//cycle through targets to find the closest one

foreach (var enemy in targets)

{

// use Vector3.Distance to calculate wolf's distance from each target

distance = Vector3.Distance(enemy.transform.position,transform.position);

// if there is a target within alert distance set as target

if (distance < lookDist)

{

lookDist = distance;

myTarget = enemy;

}

// log current target \*\*for testing only\*\*

if (myTarget == enemy)

{

Debug.Log("my closest target is " + myTarget.transform.name);

}

}

You can also have a list of lists or arrays and this can be useful for all kinds of stuff, inventories, pickups. Ect. As you can imagine they are super powerful.