Weighted Randomization

A weighted randomization would be when you have several values that you want to present with different odds between them. For example, consider values A, B, and C. If you were to decide that you want one of these 3 values, but you want A to occur 20% of the time, B 40%, and C 60%, that would be a weighted randomization. The odds of each value are potentially different, and adding to 100%. These classes will provide you the tools to define and implement your own weighted randomization. I used this myself for creating templates for enemy types in an RPG, and allocating points to stats based on the weights defined by the template.

To use simply initialize a WeightedRandomizer instance with the value type as your generic argument. Call AddWeight with the value you want and the odds that you want that value to occur. For each value that you add, the odds can be whatever you want between 0 and 1, but before you try and get a value, the sum of all the weights provided must add up to 1, so that there is a value guaranteed to come back. Once all weights are added, use the GetNext method to get the next value.

```
public void RandomFloat()
{
    WeightedRandomizer<float> randFloat = new WeightedRandomizer<float>();
    randFloat.AddOrUpdateWeight(1, .25f);
    randFloat.AddOrUpdateWeight(2, .25f);
    randFloat.AddOrUpdateWeight(3, .50f);

    RunBattery(randFloat, 200);
}

private void RunBattery(WeightedRandomizer<float> rand, int runs)
{
    float val;
    int oneCount = 0;
    int twoCount = 0;
    int threeCount = 0;
    for (int i = 0; i < runs; i++)
    {
        val = rand.GetNext();
        if (val == 1)
            oneCount++;
        if (val == 2)
            twoCount++;
        if (val == 3)
            threeCount++;
    }

    Debug.Log(string.Format("one: {0}, two: {1}, three: {2}", oneCount, twoCount, threeCount));
}
```