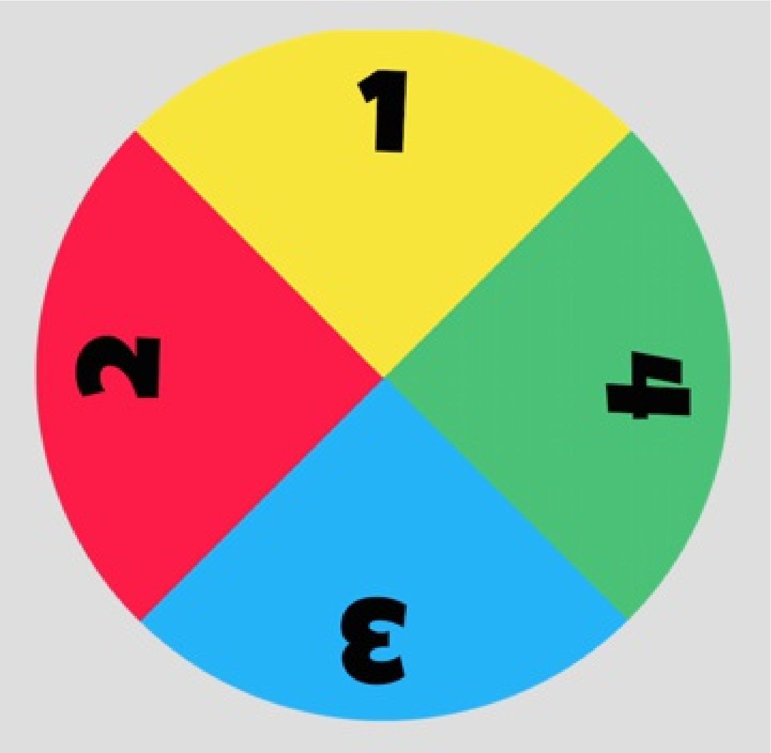
Wheel Spinner Manual

Do you want to make a game like Pirates King ? or a prize spin game? or just make a tool that choose a random result from many choices ?
WheelSpinner provide you the way to implement that you want.
Features

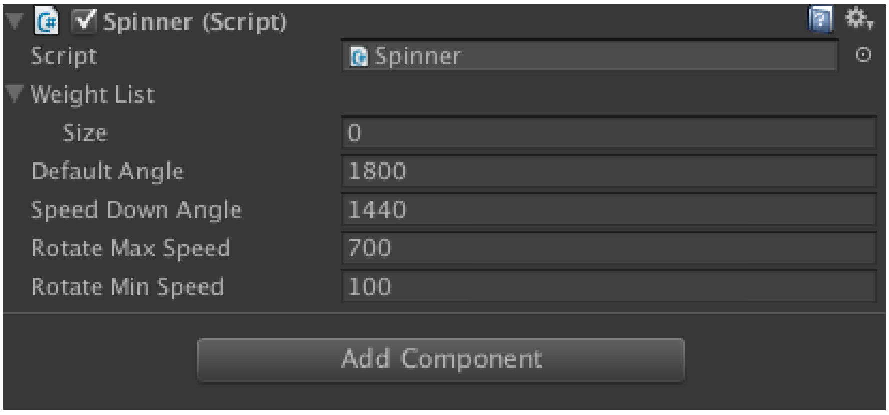
- Random spin with **weight**. You can decide what result is more important to make its probability higher than the other.
- Support multi-platform iOS, Android, Web Player
- Use latest Unity UI system. Easy to use and easy to modify.
- Easy to config and extends. In sample, we created 8 spin mode from 3 part mode to 10 part mode. Of course, you can easily make the further mode with resources you want.
- Alternative mode with DOTween

Create a Spinner

To create a spinner. First you must to prepare a spinner resources. For example, here we create a four mode spinner with the texture as bellow



At first you create a sprite with that texture and attach the script `Spinner` to it.

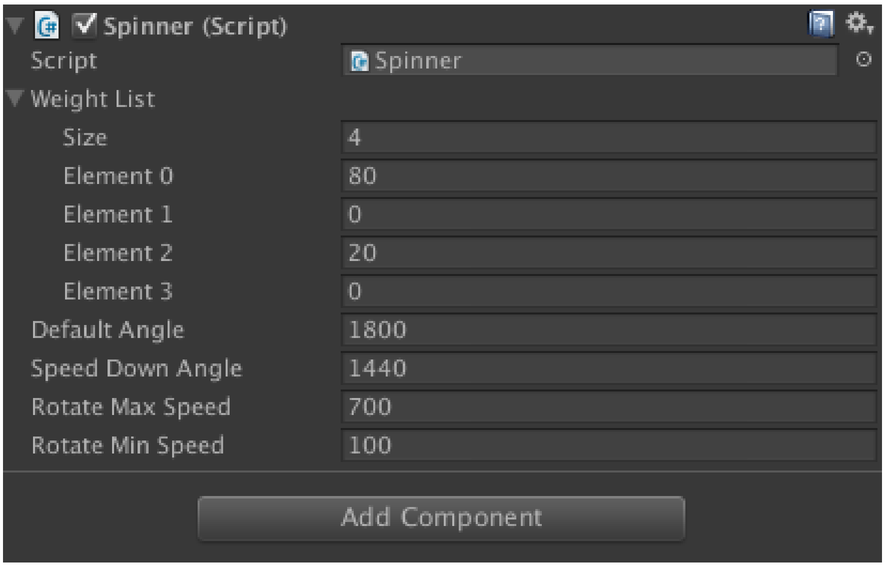


☑ Spinner (Script)

Script	Spinner
Weight List	
Size	0
Default Angle	1800
Speed Down Angle	1440
Rotate Max Speed	700
Rotate Min Speed	100

The only thing you need to config here is fill the `Weight List`. You must notice that if you how part mode you want to make, the `Weight List` must also have the same the number of elements.

You choose size = 4. And here is the most exiting of this package, choose weight for every elements. For example here we set `Element 0 = 80`, `Element 3 = 20`. Its means, the probability of 1 part (yellow part) is 80% and the probability of 3 part (blue part) is 20%. Of course, you don't need always choose the probability that sum of that equal to 100. It can be whatever you want, and the probability of a result will be calculated by `its probability/ sum of probability.` Make sure you change it only before first spin.



☑ Spinner (Script)

Script	Spinner
Weight List	
Size	4
Element 0	80
Element 1	0
Element 2	20
Element 3	0
Default Angle	1800
Speed Down Angle	1440
Rotate Max Speed	700
Rotate Min Speed	100

Other parameters

1. `Default Angle`: The angle that spinner rotate before slow down and stop. Is must be multiple of 360.
2. `Speed Down Angle`: The angle that spinner start to slow down speed.
3. `Rotate Max Speed`: Max rotate speed.
4. `Rotate Min Speed`: Speed right before stop.

Deeper Inside Source Code

```
int BiasRandom(List<float> weightList)
    {
        float total = weightList.Sum();
        if (total == 0)
        {
            return 0;
        }
        float rand = Random.Range(0, total);
        for (int i = 0; i < weightList.Count; i++)
        {
            rand -= weightList[i];
            if (rand <= 0)
            {
                return i;
            }
        }
        return 0;
    }
```

Here you create a function to choose a random index from weight list to get your result.
The idea behind the spinner is that you will rotate the spinner a `Default Angle` with a specific speed. And after that, you slow down it to a angle that point to the result that you want.
Every results's angle are calculated already in `angleList`.

```
angleUnit = 360f/weightList.Count;
angleList = new List<float>();
for (int i = 0; i < weightList.Count; i++)
{
    float angle = defaultAngle + i * angleUnit;
    angleList.Add(angle);
}
```

To slow down speed to the speed you want, you must calculate the spinner acceleration.
Do you remember the formula `maxspeed^2 - minspeed^2 = 2 * distance * acceleration` ? Here is also same.

```
rotateAcceleration = (rotateMaxSpeed * rotateMaxSpeed
- rotateMinSpeed * rotateMinSpeed) / (2 * rotateAngle);
```

After each spin turn, you must update the index of each weight for next spin turn.

```
void ShiftIndex(int shift)
{
    List<float> newWeightList = new List<float>();
    for (int i = 0; i < weightList.Count; i++)
    {
        newWeightList.Add(weightList[(i + shift) % weightList.Count]);
    }
    weightList = newWeightList;
}
```

and last, spin the spinner

```
private IEnumerator SpinProcess(float angle)
{
    curEuler = transform.eulerAngles;
    Vector3 newAngle = curEuler - angle * Vector3.forward;
    while (curEuler.z > newAngle.z)
    {
        if (curEuler.z < speedDownZ)
        {
            rotateSpeed = Mathf.MoveTowards(rotateSpeed,
            rotateMinSpeed, rotateAcceleration*Time.deltaTime);
        }
        curEuler.z = Mathf.MoveTowards(curEuler.z, newAngle.z,
        rotateSpeed * Time.deltaTime);

        transform.eulerAngles = curEuler;
        yield return null;
    }
    isRotating = false;
}
```

Alternative Spinner with DOTween

DOTween is really strong plugin to make object moving or rotating. It is an update version of HOTween, if you have any experience with tweening, it's so great. In this package, we provide you an alternative way to create the spinner. It's very simple and run with high performance.
First, you must import DOTween to your project. It's totally FREE.
https://www.assetstore.unity3d.com/en/#!/content/27676
You only need to set the `defaultAngle`. And then run the `DGMode` scene and enjoy!

Let's enjoy and make your fantastic games!

Do Trung Kien