

Classes

Monday, January 4, 2021 1:50 PM

A key part of object-oriented programming, breaking functionality into modular segments of code. MonoBehaviours allow attaching scripts to gameObject. MonoBehaviours are for behaviors!

Custom Classes

Use MonoBehaviours when creating game logic features and behaviors.

Can be thought of as a blueprint.

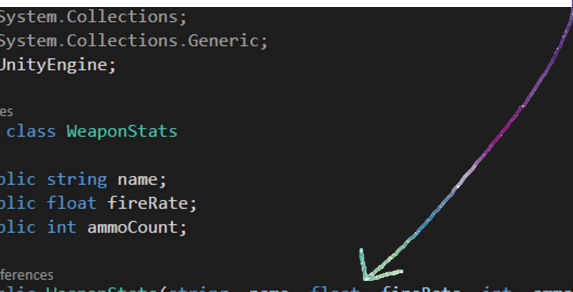
Can create classes within classes.

```
55 1 reference
56 public void CalculateMovement()
57 {
58     float hInput = Input.GetAxis("Horizontal");
59     float vInput = Input.GetAxis("Vertical");
60
61     transform.Translate(new Vector3(hInput, vInput, 0) * speed * Time.deltaTime);
62
63     if (transform.position.x > 8.5f)
64     {
65         transform.position = new Vector3(8.5f, transform.position.y, 0);
66     }
67     else if (transform.position.x < -8.5f)
68     {
69         transform.position = new Vector3(-8.5f, transform.position.y, 0);
70     }
71 }
72
73 1 reference
74 public void Shoot()
75 {
76     if (Input.GetKeyDown(KeyCode.Space))
77     {
78         Instantiate(laserPrefab, firePos.position, Quaternion.identity);
79     }
80 }
```

Constructors

- Used for initialization.
- Named after the class it's found in.

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 3 references
6 public class WeaponStats
7 {
8     public string name;
9     public float fireRate;
10    public int ammoCount;
11
12    2 references
13    public WeaponStats(string _name, float _fireRate, int _ammoCount)
14    {
15        // name = "Blasters";
16        // fireRate = 0.25f;
17        // ammoCount = 50;
18        name = _name;
19        fireRate = _fireRate;
20        ammoCount = _ammoCount;
21    }
22 }
```



```

22 public class ClassesPlayer : MonoBehaviour
23 {
24     private WeaponStats blasters;
25     private WeaponStats rockets;
26
27     private void Start()
28     {
29         blasters = new WeaponStats("Blasters", 0.25f, 50);
30         rockets = new WeaponStats("Rockets", 5.0f, 1);
31
32         Debug.Log("Current Weapon Name: " + blasters.name);
33         // blasters = new WeaponStats();
34         // blasters.name = "Blasters";
35         // blasters.fireRate = 0.25f;
36         // blasters.ammoCount = 50;
37
38         // rockets = new WeaponStats();
39         // rockets.name = "Rockets";
40         // rockets.fireRate = 5.0f;
41         // rockets.ammoCount = 1;
42     }
43 }
44
45
46

```

Custom Class Example, RPG SPELL SYSTEM

```

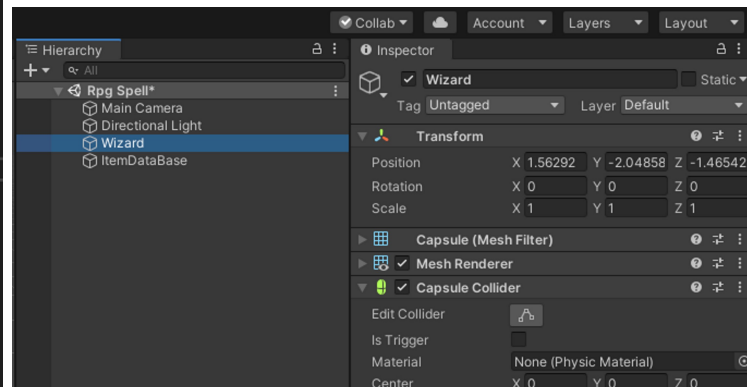
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 /// <summary> Define the traits of a spell.
6 [System.Serializable]
7
8 public class Spell
9 {
10     public string spellName;
11     public int levelRequired;
12     public int itemIdRequired;
13     public int expGained;
14
15     // Constructor to initialize values.
16     public Spell(string _spellname, int _levelRequired, int _itemIdRequired, int _expGained)
17     {
18         spellName = _spellname;
19         levelRequired = _levelRequired;
20         itemIdRequired = _itemIdRequired;
21         expGained = _expGained;
22     }
23
24     public void Cast()
25     {
26         Debug.Log("Casting: " + spellName);
27     }
28 }
29
30

```

```

1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class Wizard : MonoBehaviour
6 {
7     // public Spell fireBlast;
8     public Spell[] spells;
9     public int level = 1;
10    public int exp;
11
12    void Start()
13    {
14        // fireBlast = new Spell("Fire Blast", 1, 27, 35);
15    }
16 }

```



```
13  Unity Message | 0 references
14  void Start()
15  {
16      // fireBlast = new Spell("Fire Blast", 1, 27, 35);
17  }
18  Unity Message | 0 references
19  void Update()
20  {
21      if (Input.GetKeyDown(KeyCode.Space))
22      {
23          // fireBlast.Cast();
24          // exp += fireBlast.expGained;
25
26          foreach (var spell in spells)
27          {
28              if (spell.levelRequired == level)
29              {
30                  spell.Cast();
31                  exp += spell.expGained;
32              }
33          }
34      }
35  }
36  }
```

Project

Assets > 9. Classes > Scenes

BankExample

Classes

Pet

RefValue

Rpg Spell

Wizard (Script)

Script

Wizard

Spells

Size

2

Fire Blast

Spell Name

Fire Blast

Level Required

1

Item Id Required

27

Exp Gained

35

Ice Blast

Spell Name

Ice Blast

Level Required

3

Item Id Required

29

Exp Gained

55

Level

1

Exp

0

Default-Material

Shader

Standard

Add Component

Console

Clear

Collapse

Clear on Play

Clear on Build

Error Pause

Edit