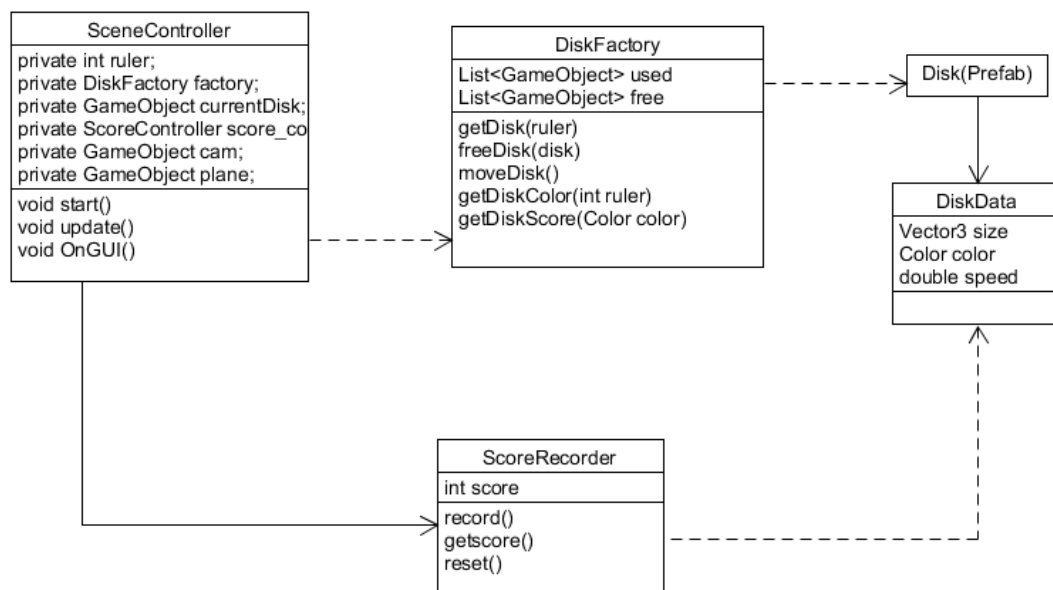


UML 图



游戏规则：打中红色 5 分，打中蓝色 10 分，打中黑色 20 分

在一定时间内达到目标分数可以进入下一轮。（待实现）

设计思路：

一开始打算想师兄一样做子弹射击，不过由于周末要实训的关系，捣鼓了一下有点问题，于是换成了简单的点击射击加分。

类的关系如上，这次没有用到 `SSDirector`。然后场记 `SceneController` 充当每轮的管理员 `RoundController` 的角色，负责接收从 `DiskFactory` 运送过来的 `disk` 以及动作。游戏的动作（点击）也放在了这里面。`ScoreController` 负责计分规则，判断颜色进行 `score` 的更改。

`DiskData` 作为飞碟的数据参数，脚本挂载在了 `Prefabs` 的 `Disk` 里面，把每个飞碟的参数分离出来集中处理。

这一次作业的核心是工厂模式的实现，第一问的作业相当于飞碟游戏的铺垫，如何从工厂类 DiskFactory 生成实例，回收对象。在这个框架的基础上，对游戏效果进行管理。应该说第二问和第一问的区别是，第一问是点击生成 Cylinder。第二问是点击飞碟修改分数，飞碟还有动作。其核心的管理框架都是工厂模式生产，由 Factory 批量化生产实例，并通过 used 和 free 两个 List 进行管理。

核心代码的实现：

```
5 public class Singleton<T> : MonoBehaviour where T : MonoBehaviour
6 {
7
8     protected static T instance;
9
10    public static T Instance {
11        get {
12            if (instance == null) {
13                instance = (T)FindObjectOfType(typeof(T));
14                if (instance == null) {
15                    Debug.LogError ("An instance of " + typeof(T) +
16                        " is needed in the scene, but there is none.");
17                }
18            }
19            return instance;
20        }
21    }
22 }
```

每个类都是单实例的实现，不过因为之前没看懂这个模板类，所以是每个类都写了一个 getInstance()...

DiskFactory.cs

```
public void FreeDisk(GameObject disk){
    for(int i = 0; i < used.Count;i++){
        if(used[i].transform.position == disk.transform.position){
            used.RemoveAt(i);
            disk.transform.position = new Vector3 (-19, 7,
0);
```

```

        free.Add (disk);
        break;
    }
}
disk.transform.position = new Vector3(-19,0,0);
used.Remove (disk);
free.Add (disk);
}

```

```

public GameObject GetDisk(int ruler){
    if (free.Count == 0) {
        GameObject a_disk = Instantiate<GameObject> (Resources.Load<GameObject> ("Prefabs/RedDisk"));
        Diskdata dd = a_disk.GetComponent<Diskdata> ();
        Renderer render = a_disk.GetComponent<Renderer> ();
        render.material.color = getDiskColor (ruler);
        dd.color = render.material.color;
        dd.score = getDiskScore (dd.color);
        dd.transform.position = new Vector3 (-19, 1, 0);
        used.Add (a_disk);
        //要不要修改Adisk 的屬性 diskdata?
        return a_disk;
        // Created a new disk
    }
    else {
        GameObject a_disk = free [0];
        free.RemoveAt (0);
        used.Add (a_disk);
        //要不要修改Adisk 的屬性 diskdata?
        return a_disk;
    }
}

```

SceneController.cs

```

void OnGUI(){
    GUI.Box(new Rect(0, 0, 80, 40), "Round " + ruler);
    GUI.Box(new Rect(0,40,80,40),"Score: " + score_controller.getscore());
    if (Input.GetButtonDown ("Fire1")) {

```

```

        Vector3 mp = Input.mousePosition;
        Camera ca = cam.GetComponent<Camera> ();
        Ray ray = ca.ScreenPointToRay (Input.mousePosition
    );

    RaycastHit hit;
    if (Physics.Raycast (ray, out hit)) {
        GameObject get_hit = hit.collider.gameObject;
        if (get_hit.tag.Contains ("Finish")) {
            score_controller.record (get_hit.GetComponent
ent<Diskdata> ());
            factory.FreeDisk (get_hit);
        }
    }
}

```