

玩家结构体和枚举

#region 8、玩家枚举和结构体

6 个引用

```
enum E_PlayType
{
    /// <summary>
    /// 玩家
    /// </summary>
    Player,
    /// <summary>
    /// 电脑
    /// </summary>
    Computer,
}
```

7 个引用

```
struct Player
{
    public E_PlayType type;
    // 这里我们不知道xy的位置，但是我们知道格子对象是带有xy的
    //所以我们只需要记录格子的下标索引 作为玩家的位置就很好
    public int index;

    2 个引用
    public Player(int index,E_PlayType type)
    {
        this.index = index;
        this.type = type;
    }

    //绘制方法
    2 个引用
    public void Draw(Map map)
    {
        //从传入的map中找到index下标的格子，就是我们玩家当前的位置
        Grid grid = map.grids[index];
        //找到位置，并修改为玩家的颜色，画出玩家即可，
        Console.SetCursorPosition(grid.pos.x, grid.pos.y);
        switch (this.type)//判断是玩家还是电脑，他们都是用的这个结构体
        {
            case E_PlayType.Player:
                Console.ForegroundColor = ConsoleColor.Cyan;
                Console.Write("☆");
                break;
            case E_PlayType.Computer:
                Console.ForegroundColor = ConsoleColor.DarkMagenta;
                Console.Write("▲");
                break;
            default:
                break;
        }
    }
}
```

#endregion

思想：就是设置一个枚举和结构体，枚举就是玩家和电脑，结构体是公用的。属性一致，添加了draw自己的方法，跟格子类一样。这里我们会根据类型判断取画自己的图形。把重合的逻辑单独写一个函数来封装这个draw函数：

```
#region 9、绘制玩家(存在重合)
1 个引用
static void DrawPlayer(Player player, Player computer, Map map)
{
    //重合时:
    if (player.index == computer.index)
    {
        Console.SetCursorPosition(map.grids[player.index].pos.x, map.grids[player.index].pos.y);
        Console.ForegroundColor = ConsoleColor.Green;
        Console.Write("◎");
    }
    else //不重合，就画自己的
    {
        player.Draw(map);
        computer.Draw(map);
    }
}
#endregion
```

output:

不重合

```
#region 4、游戏场景逻辑
1 个引用
static void GameScene(int w, int h, E_SceneType nowSceneType)
{
    //绘制墙体
    DrawWall(w, h);
    //绘制地图, 初始化一张地图
    Map map = new Map(14, 3, 90);
    map.Draw();

    //绘制玩家
    Player player = new Player(0, E_PlayType.Player);
    Player computer = new Player(1, E_PlayType.Computer);
    DrawPlayer(player, computer, map);

    while (true)
    {

```

展示:



展示重合:

```
map.Draw();

//绘制玩家
Player player = new Player(0, E_PlayType.Player);
Player computer = new Player(0, E_PlayType.Computer);
DrawPlayer(player, computer, map);

while (true)
{

```

下标一致即重合



