

格子结构体与枚举类型以及draw

格子里面的道具类型

```
enum E_Grid_Type
{
    /// <summary>
    /// 普通格子
    /// </summary>
    Normal,
    /// <summary>
    /// 炸弹
    /// </summary>
    Boom,
    /// <summary>
    /// 暂停
    /// </summary>
    Pause,
    /// <summary>
    /// 时空隧道：随机 倒退、暂停、换位置
    /// </summary>
    Tunnel,
}
```

格子的位置，用结构体标识，方便后续的3d学习，他本质也是结构体

```
struct Vector2
{
    public int x;//x轴
    public int y;//y轴
    //构造函数
    0 个引用
    public Vector2(int x,int y)
    {
        this.x = x;
        this.y = y;
    }
}
```

格子结构体，以及draw方法。

```

struct Grid
{
    //格子的类型
    public E_Grid_Type type;
    //格子的位置，用个类型代替，
    public Vector2 pos;

    //构造函数
    1 个引用
    public Grid(int x,int y,E_Grid_Type type)
    {
        pos.x = x;
        pos.y = y;
        this.type = type;
    }

    //既然是格子类，且后面还要画到游戏界面，在这里定义画的方法就很方便了，
    //因为位置已经有了
    1 个引用
    public void Draw()
    {
        Console.SetCursorPosition(pos.x, pos.y); //这里就很方便直接拿到位置就画
        switch (type)
        {
            case E_Grid_Type.Normal:
                //画普通格子
                Console.ForegroundColor = ConsoleColor.White;
                Console.Write("□");
                break;
            case E_Grid_Type.Boom:
                //画炸弹
                Console.ForegroundColor = ConsoleColor.Red;
                Console.Write("●");
                break;
            case E_Grid_Type.Pause:
                //画暂停
                Console.ForegroundColor = ConsoleColor.Blue;
                Console.Write("||");
                break;
            case E_Grid_Type.Tunnel:
                //画时空隧道
                Console.ForegroundColor = ConsoleColor.Yellow;
                Console.Write("∞");
                break;
        }
    }
}

```

使用方法：很方便，只需要定义一个格子对象，调用draw即可，因为对象是保存了自己的位置的，所以需要就直接画。

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
    Grid grid = new Grid(5, 5, E_Grid_Type.Boom);
    grid.Draw();
    //绘制玩家

    while (true)
    {
        ...
    }
}
```