

任务63：蛇转向 知识点

蛇转向就需要监听按键，监听按键就会导致阻塞，所以我们需要一种方法来防止阻塞：

1、使用Console.KeyAvailable:

```
public GameScene()
{
    map = new Map();
    snake = new Lesson6.Snake(40, 10);
}
2 个引用
public void upData()
{
    if (updateTime == 4000)
    {
        map.Draw();
        snake.Move();
        snake.Draw();
        updateTime = 0;
    }
    updateTime++;
    //new 判断有没有键盘输入，如果有才会进入if
    if (Console.KeyAvailable) //这里会导致循环变慢，所以上面的计数次数要减少
    {
        //检测输入输出，不能在间隔帧里面去处理，应该每次都检测，这样才准确
        switch (Console.ReadKey(true).Key)
        {
            case ConsoleKey.W:
                snake.changeDir(E_MoveDir.Up);
                break;
            case ConsoleKey.S:
                snake.changeDir(E_MoveDir.Down);
                break;
            case ConsoleKey.A:
                snake.changeDir(E_MoveDir.Left);
                break;
            case ConsoleKey.D:
                snake.changeDir(E_MoveDir.Right);
                break;
        }
    }
}
```

当有按键激活时，返回true，进入if语句，修改蛇方向，循环一直在跑，这时候就会实现转向，注意这个属性会导致主循环速度变慢，因为设计到按键监听激活，一部分的时间给了按键监听激活，所以主循环时间会变慢

方法二：使用线程，在线程使用while循环

```
1 个引用
public GameScene()
{
    map = new Map();
    snake = new Lesson6.Snake(40, 10);
    new Thread(new ThreadStart(m)).Start();
}

1 个引用
public void m()
{
    while (true)
    {
        switch (Console.ReadKey(true).Key)
        {
            case ConsoleKey.W:
                snake.changeDir(E_MoveDir.Up);
                break;
            case ConsoleKey.S:
                snake.changeDir(E_MoveDir.Down);
                break;
            case ConsoleKey.A:
                snake.changeDir(E_MoveDir.Left);
                break;
            case ConsoleKey.D:
                snake.changeDir(E_MoveDir.Right);
                break;
        }
    }
}
```

线程的方法监听按键

这里使用第一种方法，线程还没学习

```
Snake.cs Food.cs Wall.cs Game.cs EndScene.cs GameScene.cs ISceneUpdate.cs GameObject.cs Position.cs
Snake
65
66
67 #endregion
68
69 #region Lesson8 改变方向
70 4 个引用
71 public void changeDir(E_MoveDir dir)
72 {
73     //1、只有头部,那么可以直接上下左右转向
74     if(nowNum == 1) 只有头部, 随便转向
75     {
76         this.dir = dir;
77     }
78     else //2、有身体,就不能和当前移动方向相反的方向直接转向, (往右移动中按左没反应)
79     {
80         switch (this.dir) 有身体后, 转向需要限制
81         {
82             case E_MoveDir.Up:
83             case E_MoveDir.Down:
84                 if (dir == E_MoveDir.Down || dir == E_MoveDir.Up)
85                 {
86                     break;
87                 }
88                 this.dir = dir;
89                 break;
90             case E_MoveDir.Left:
91             case E_MoveDir.Right:
92                 if (dir == E_MoveDir.Left || dir == E_MoveDir.Right)
93                 {
94                     break;
95                 }
96                 this.dir = dir;
97                 break;
98         }
99     }
100
101
102
```