

10: 输入线程

```
namespace 俄罗斯方块
{
    2 个引用
    class GameScene : ISceneUpdate
    {
        Map map;
        BlockWorker woker;

        //检测输入线程
        Thread inputThread;

        1 个引用
        public GameScene()
        {
            map = new Map();
            woker = new BlockWorker();

            inputThread = new Thread(CheckInputThread);
            //设置后台线程, 声明周期随主程序决定
            inputThread.IsBackground = true;
            //开启线程
            inputThread.Start();
        }

        1 个引用
        public void CheckInputThread()
        {
            while (true)
            {
                if (Console.KeyAvailable)
                {
                    lock (woker) 添加锁, 否则 屏幕上的放行会乱跳现象, 多线程占用资源导致
                    {
                        switch (Console.ReadKey(true).Key)
                        {
                            case ConsoleKey.A:
                                //判断是否可变形
                                if (woker.CanChange(E_ChangeType.Left, map))
                                {
                                    woker.Change(E_ChangeType.Left);
                                }
                                break;
                            case ConsoleKey.D:
                                if (woker.CanChange(E_ChangeType.Right, map))
                                {
                                    woker.Change(E_ChangeType.Right);
                                }
                                break;
                            case ConsoleKey.RightArrow:// ->
                                if (woker.CanMoveLR(E_ChangeType.Right, map))
                                {
                                    woker.MoveLR(E_ChangeType.Right);
                                }
                                break;
                            case ConsoleKey.LeftArrow://<-
                                if (woker.CanMoveLR(E_ChangeType.Left, map))
                                {
                                    woker.MoveLR(E_ChangeType.Left);
                                }
                                break;
                            case ConsoleKey.DownArrow://向下移动
                                if (woker.CanMove(map))
                                {
                                    woker.AutoMove();
                                }
                                break;
                        }
                    }
                }
            }
        }

        public void upData() //游戏界面(主要的逻辑处理更新)
    }
}
```

```
lock (woker)
{
    //地图更新绘制//每一帧都绘制一次
    map.Draw();
    //绘制砖块
    woker.Draw();
    if (woker.CanMove(map))
        woker.AutoMove();
}
```

```
Thread.Sleep(200);
```

不能将延迟放锁中，否则会导致，等待线程也会等休眠的事件

修正前面的绘制方法在向下移动没有绘制，导致控制台是没有绘制向下移动的方形

```
2 个引用
public void AutoMove()
{
    //只要有移动，就需要先擦除

    ClearDraw();

    //首要得到移动的距离
    Position pos = new Position(0, 1);
    //得到所有方块，让其向下移动
    for (int i = 0; i < blocks.Count; i++)
    {
        blocks[i].pos += pos;
        //也可以直接y轴+1:
        //blocks[i].pos.y += 1;
    }
    Draw();
}
```