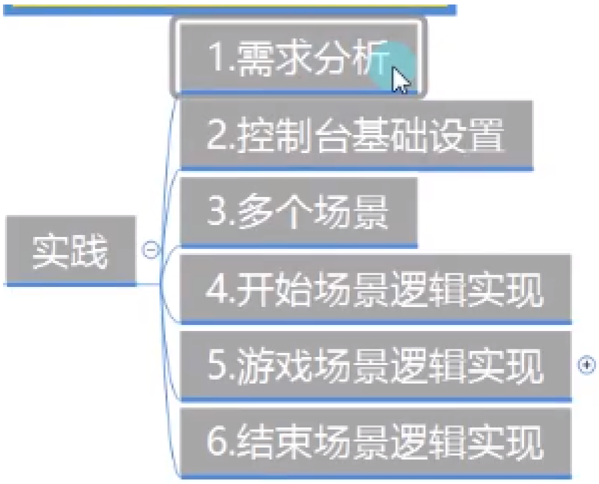
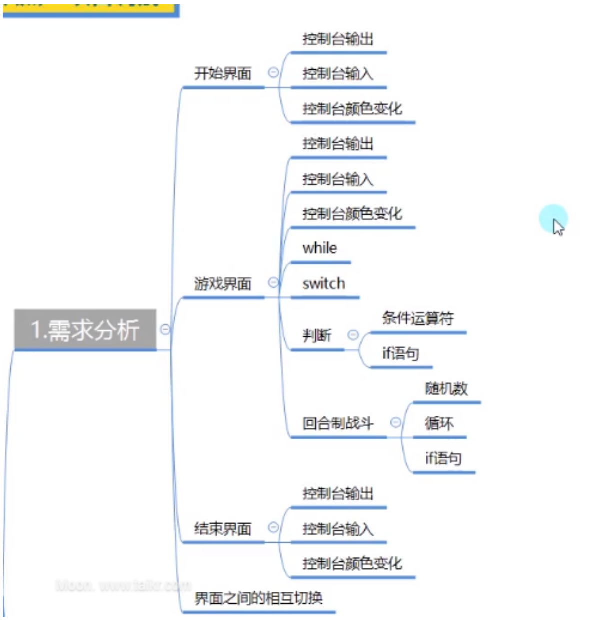
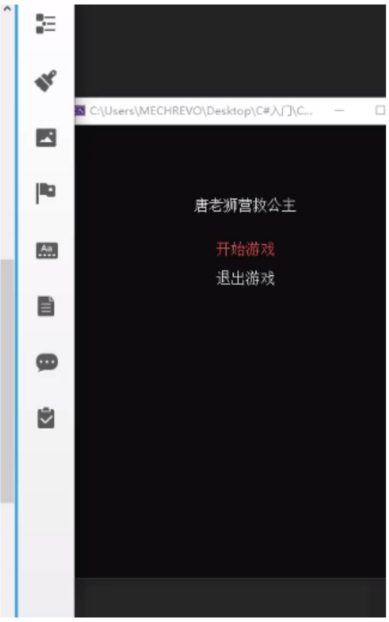
项目制作 营救公主



- 实践
  - 1.需求分析
  - 2.控制台基础设置
  - 3.多个场景
  - 4.开始场景逻辑实现
  - 5.游戏场景逻辑实现
  - 6.结束场景逻辑实现



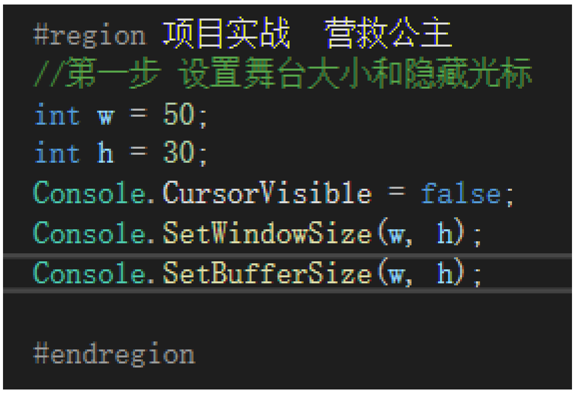
- 1.需求分析
  - 开始界面
    - 控制台输出
    - 控制台输入
    - 控制台颜色变化
  - 游戏界面
    - 控制台输出
    - 控制台输入
    - 控制台颜色变化
    - while
    - switch
    - 判断
      - 条件运算符
      - if语句
    - 回合制战斗
      - 随机数
      - 循环
      - if语句
  - 结束界面
    - 控制台输出
    - 控制台输入
    - 控制台颜色变化
  - 界面之间的相互切换



唐老狮营救公主

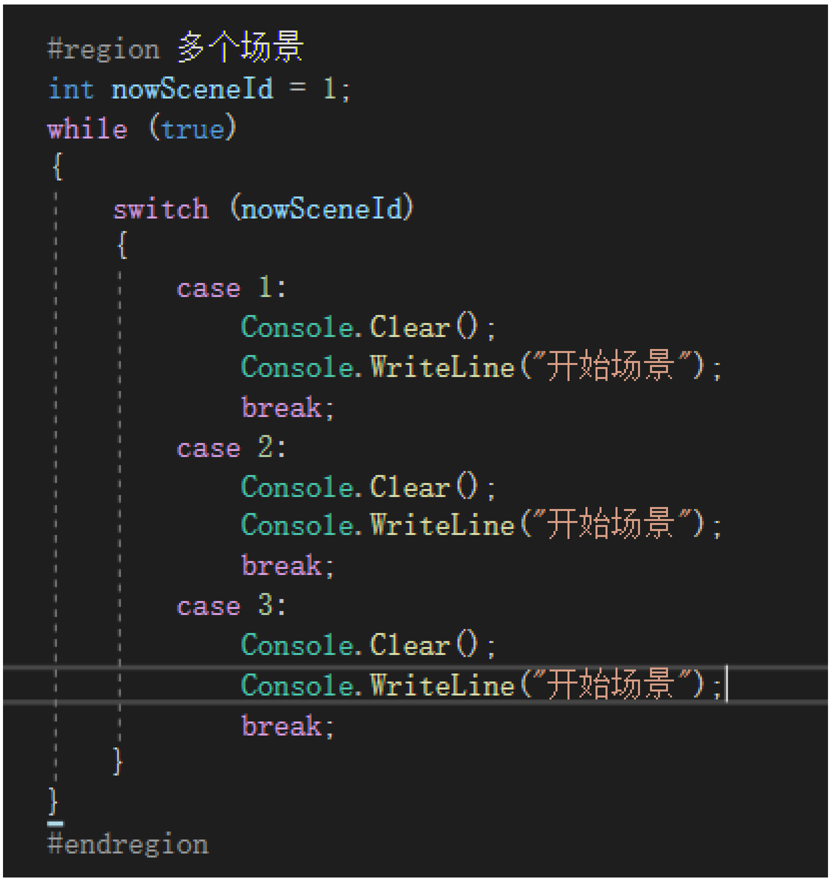
开始游戏

退出游戏



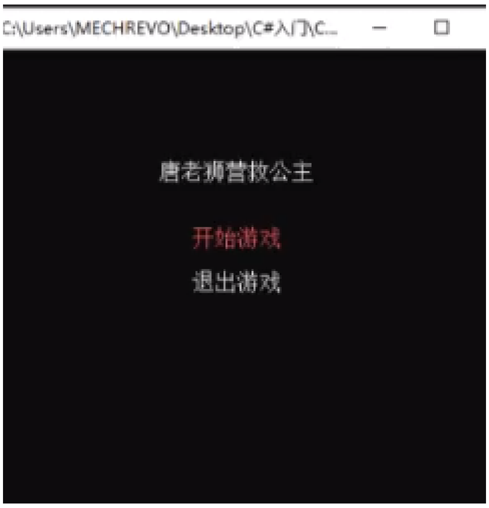
```
#region 项目实战  营救公主
//第一步 设置舞台大小和隐藏光标
int w = 50;
int h = 30;
Console.CursorVisible = false;
Console.SetWindowSize(w, h);
Console.SetBufferSize(w, h);

#endregion
```

多个场景： 就一个死循环里面 加一个switch判断ID切换场景即可。



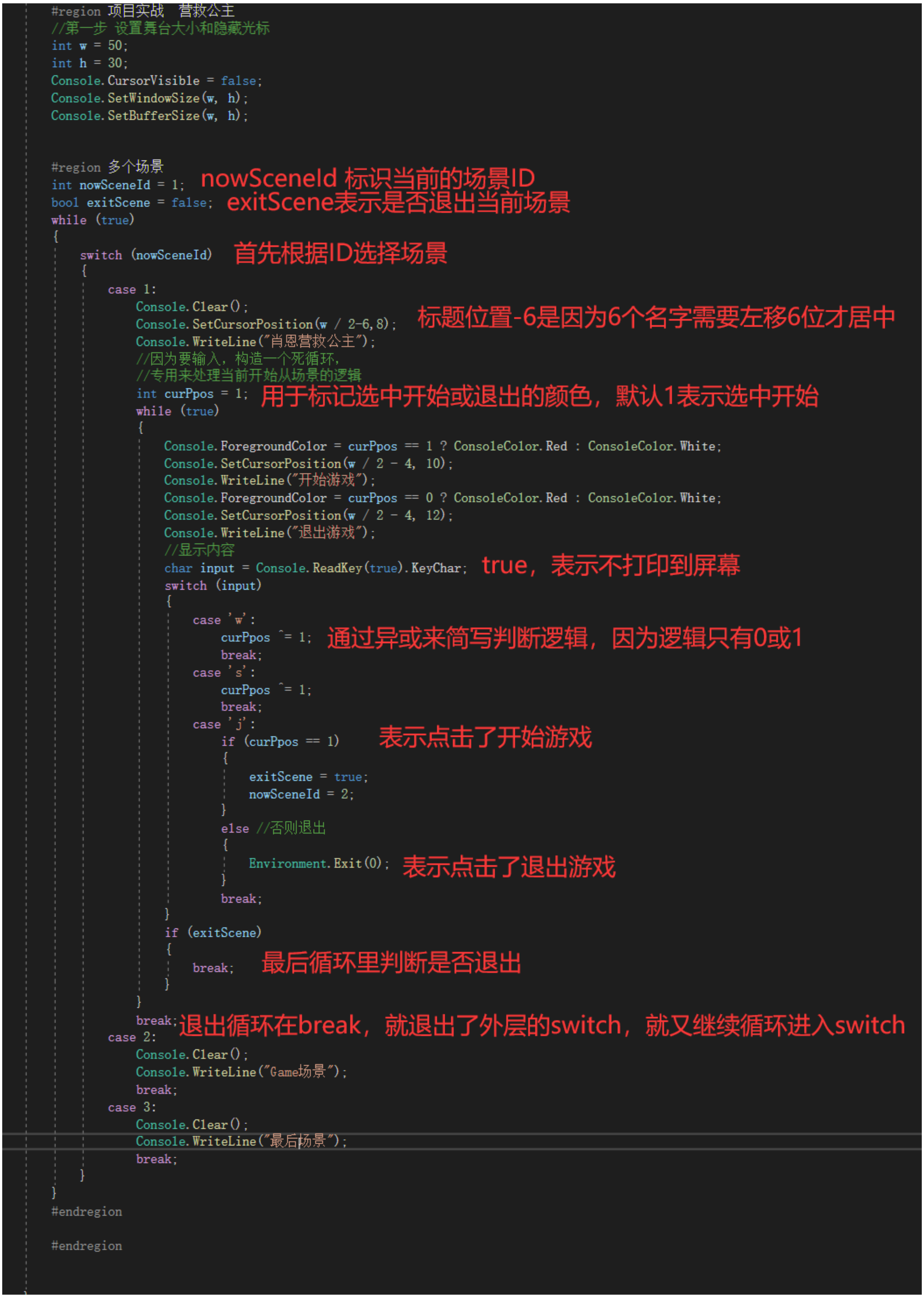
```
#region 多个场景
int nowSceneId = 1;
while (true)
{
    switch (nowSceneId)
    {
        case 1:
            Console.Clear();
            Console.WriteLine("开始场景");
            break;
        case 2:
            Console.Clear();
            Console.WriteLine("开始场景");
            break;
        case 3:
            Console.Clear();
            Console.WriteLine("开始场景");
            break;
    }
}
#endregion
```

开始场景：



```
唐老狮营救公主

开始游戏
退出游戏
```

Code:



```
#region 项目实战  营救公主
//第一步 设置舞台大小和隐藏光标
int w = 50;
int h = 30;
Console.CursorVisible = false;
Console.SetWindowSize(w, h);
Console.SetBufferSize(w, h);


#region 多个场景
int nowSceneId = 1;
bool exitScene = false;
while (true)
{
    switch (nowSceneId)
    {
        case 1:
            Console.Clear();
            Console.SetCursorPosition(w / 2-6,8);
            Console.WriteLine("肖恩营救公主");
            //因为要输入，构造一个死循环，
            //专用来处理当前开始从场景的逻辑
            int curPpos = 1;
            while (true)
            {
                Console.ForegroundColor = curPpos == 1 ? ConsoleColor.Red : ConsoleColor.White;
                Console.SetCursorPosition(w / 2 - 4, 10);
                Console.WriteLine("开始游戏");
                Console.ForegroundColor = curPpos == 0 ? ConsoleColor.Red : ConsoleColor.White;
                Console.SetCursorPosition(w / 2 - 4, 12);
                Console.WriteLine("退出游戏");
                //显示内容
                char input = Console.ReadKey(true).KeyChar;
                switch (input)
                {
                    case 'w':
                        curPpos ^= 1;
                        break;
                    case 's':
                        curPpos ^= 1;
                        break;
                    case 'j':
                        if (curPpos == 1)
                        {
                            exitScene = true;
                            nowSceneId = 2;
                        }
                        else //否则退出
                        {
                            Environment.Exit(0);
                        }
                        break;
                }
                if (exitScene)
                {
                    break;
                }
            }
            break;
        case 2:
            Console.Clear();
            Console.WriteLine("Game场景");
            break;
        case 3:
            Console.Clear();
            Console.WriteLine("最后场景");
            break;
    }
}
#endregion

#endregion
```

nowSceneId 标识当前的场景ID

exitScene表示是否退出当前场景

首先根据ID选择场景

标题位置-6是因为6个名字需要左移6位才居中

用于标记选中开始或退出的颜色，默认1表示选中开始

true，表示不打印到屏幕

通过异或来简写判断逻辑，因为逻辑只有0或1

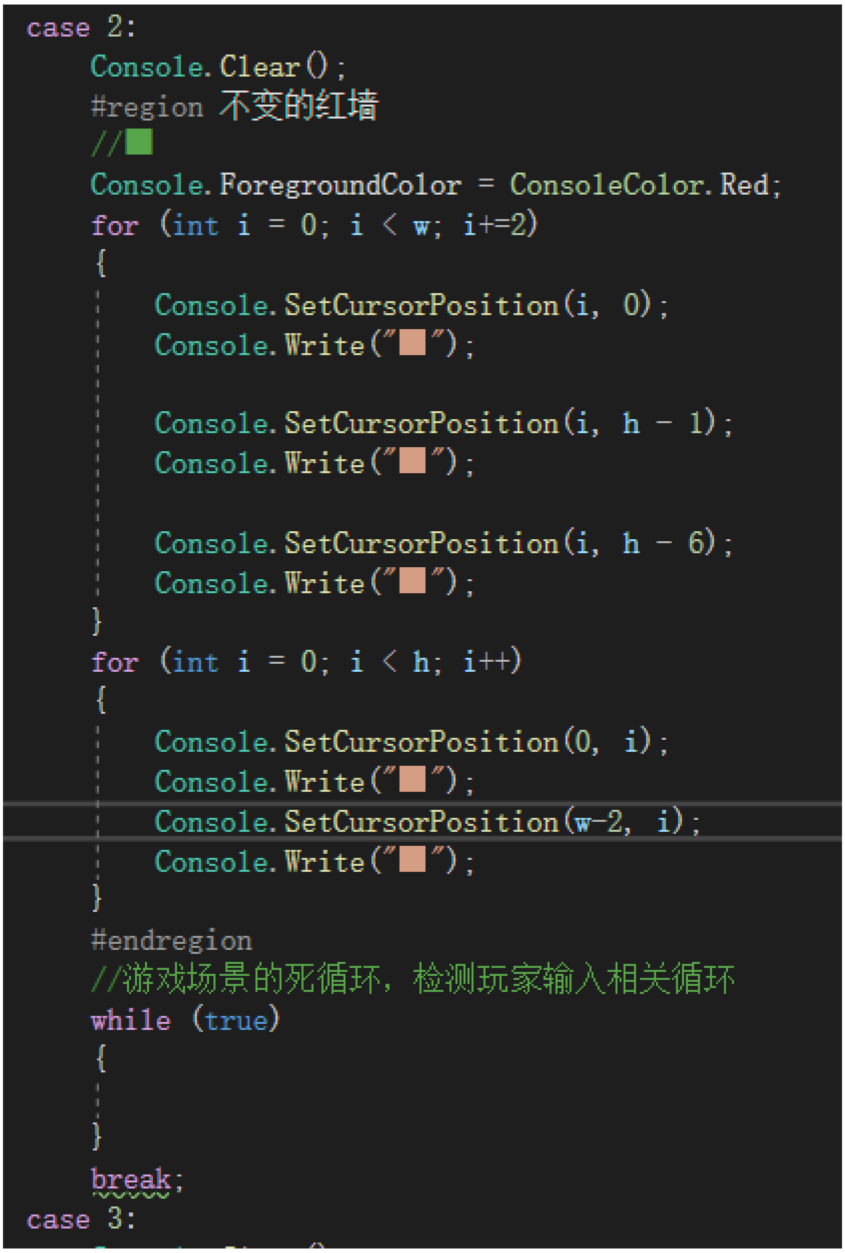
表示点击了开始游戏

表示点击了退出游戏

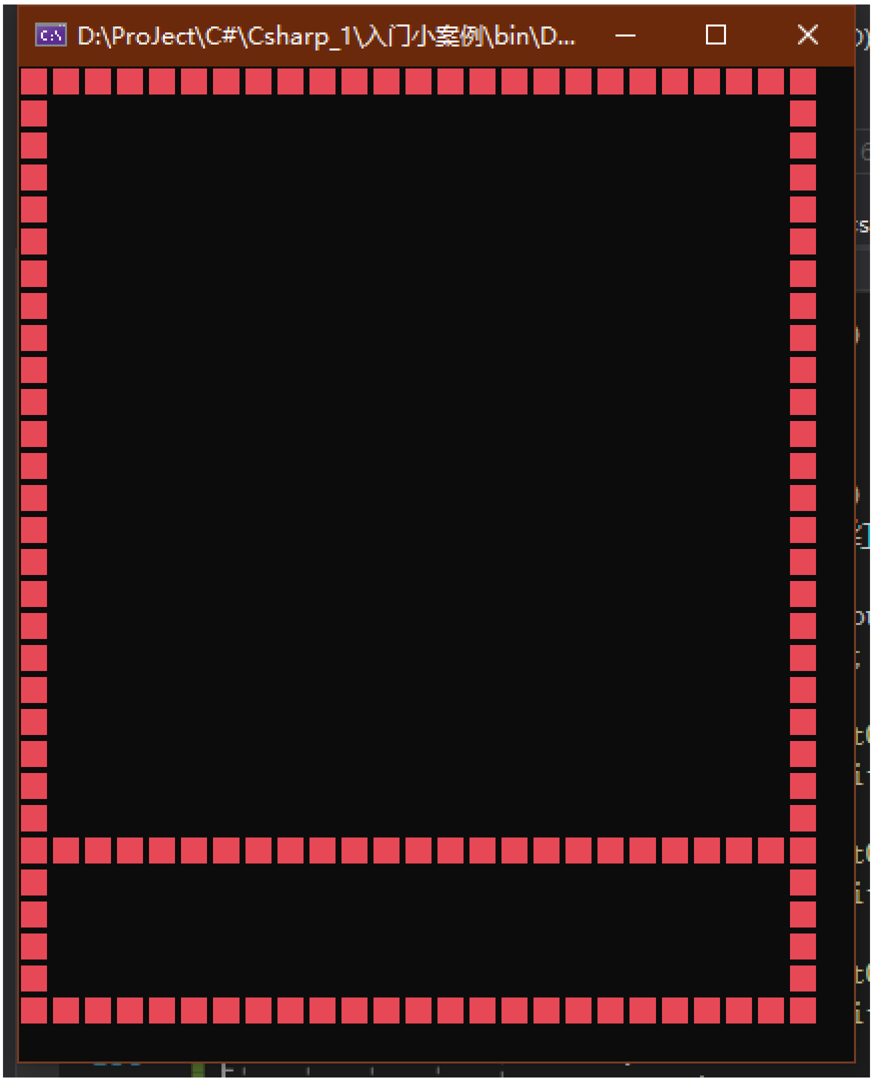
最后循环里判断是否退出

退出循环在break，就退出了外层的switch，就又继续循环进入switch

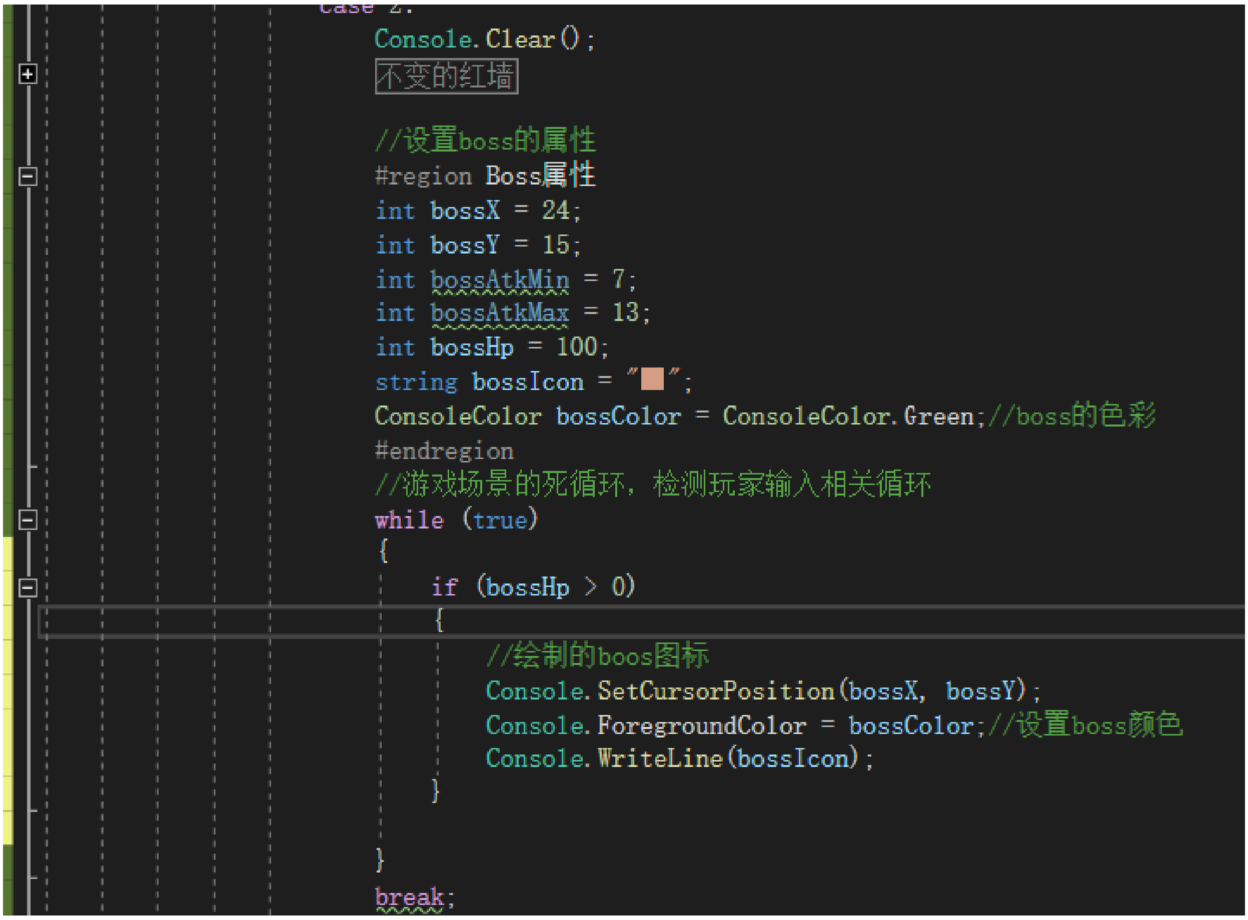
红墙：



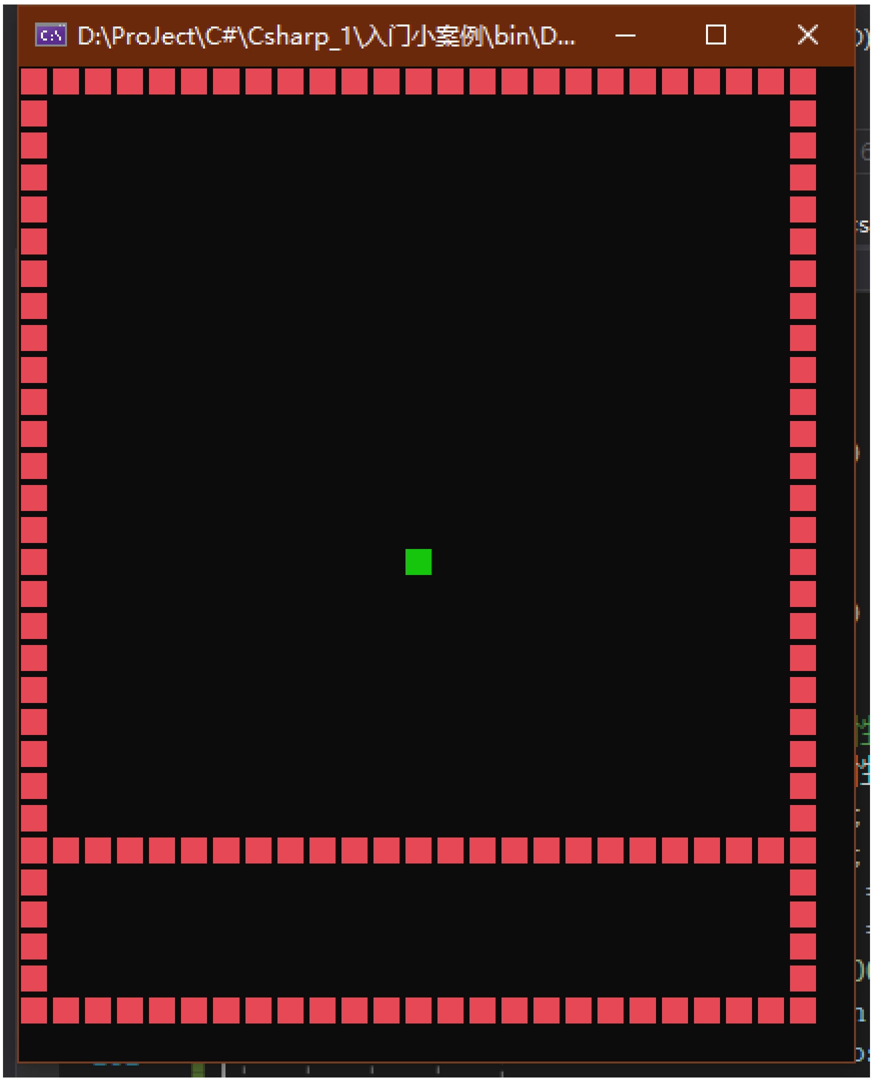
```
case 2:
    Console.Clear();
    #region 不变的红墙
    //■
    Console.ForegroundColor = ConsoleColor.Red;
    for (int i = 0; i < w; i+=2)
    {
        Console.SetCursorPosition(i, 0);
        Console.Write("■");

        Console.SetCursorPosition(i, h - 1);
        Console.Write("■");

        Console.SetCursorPosition(i, h - 6);
        Console.Write("■");
    }
    for (int i = 0; i < h; i++)
    {
        Console.SetCursorPosition(0, i);
        Console.Write("■");
        Console.SetCursorPosition(w-2, i);
        Console.Write("■");
    }
    #endregion
    //游戏场景的死循环，检测玩家输入相关循环
    while (true)
    {

    }
    break;
case 3:
```



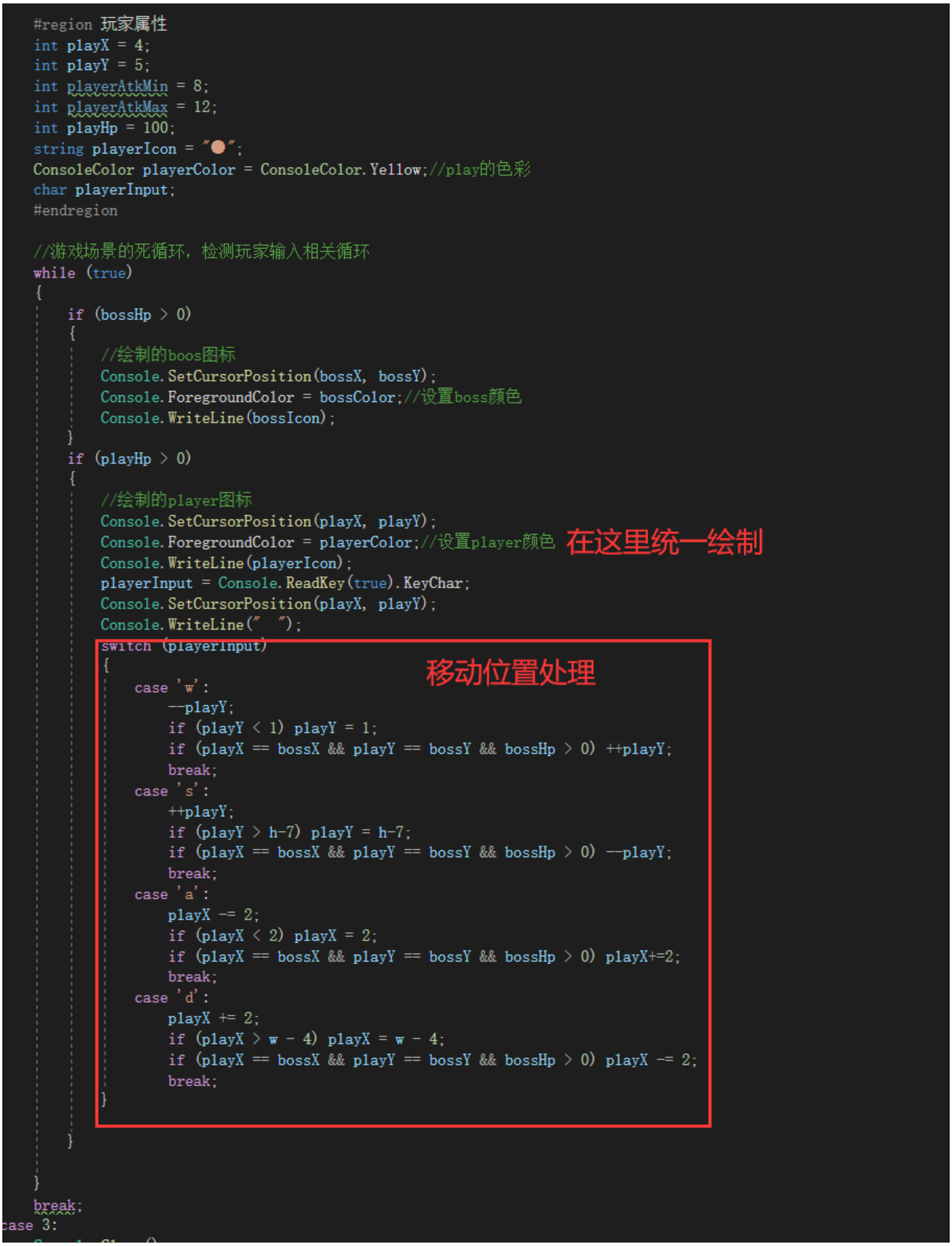
绘制boss：



```
case 2:
    Console.Clear();
    不变的红墙

    //设置boss的属性
    #region Boss属性
    int bossX = 24;
    int bossY = 15;
    int bossAtkMin = 7;
    int bossAtkMax = 13;
    int bossHp = 100;
    string bossIcon = "■";
    ConsoleColor bossColor = ConsoleColor.Green;//boss的色彩
    #endregion
    //游戏场景的死循环，检测玩家输入相关循环
    while (true)
    {
        if (bossHp > 0)
        {
            //绘制的boos图标
            Console.SetCursorPosition(bossX, bossY);
            Console.ForegroundColor = bossColor;//设置boss颜色
            Console.WriteLine(bossIcon);
        }

    }
    break;
```

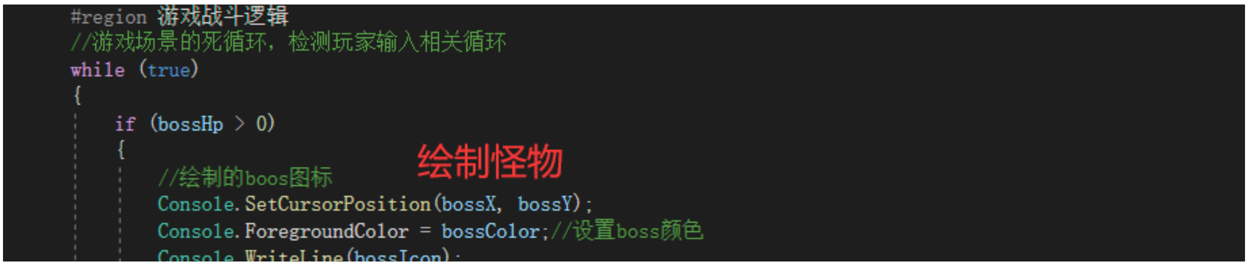


玩家移动相关：

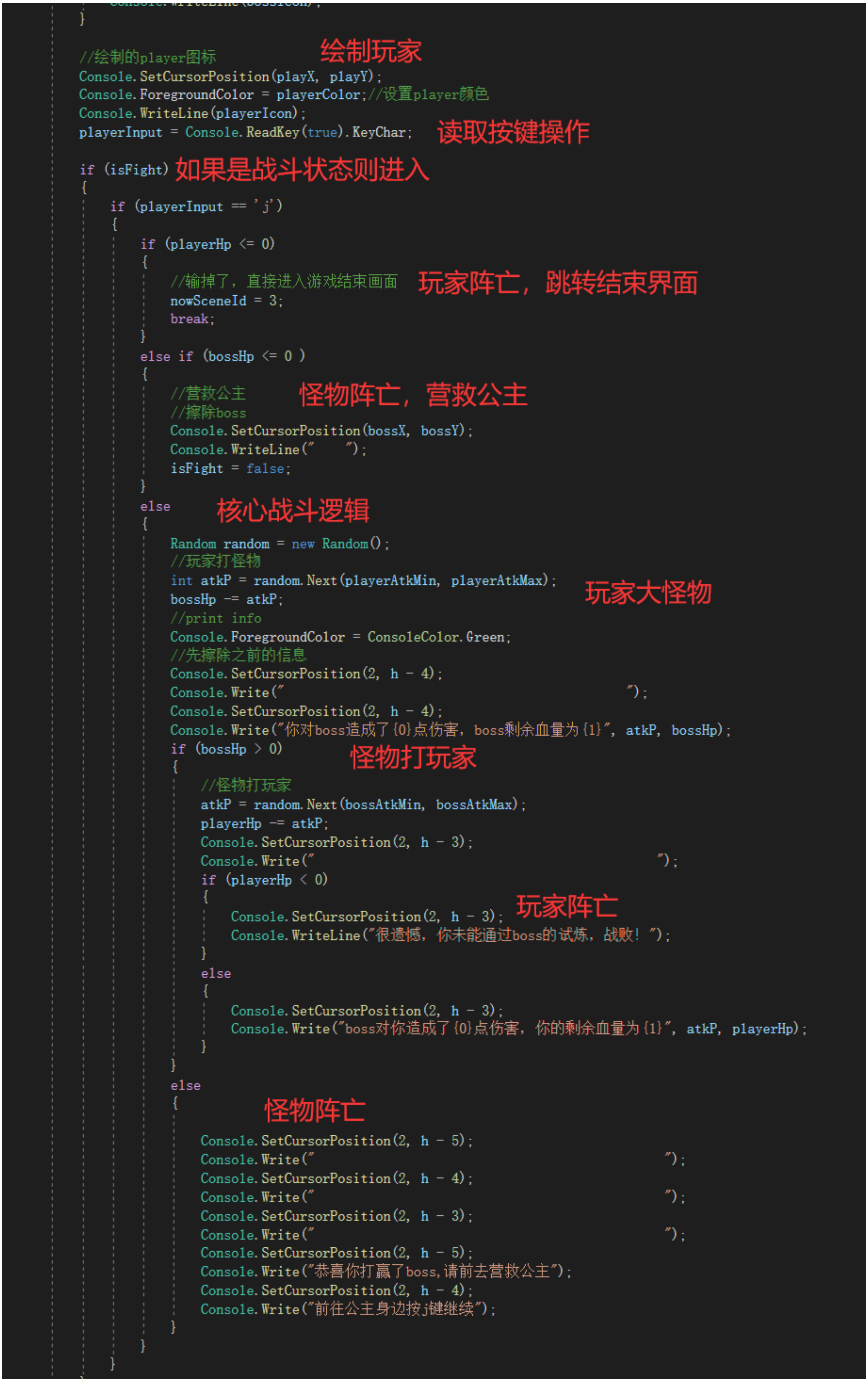


```
    #region 玩家属性
    int playX = 4;
    int playY = 5;
    int playerAtkMin = 8;
    int playerAtkMax = 12;
    int playHp = 100;
    string playerIcon = "●";
    ConsoleColor playerColor = ConsoleColor.Yellow;//play的色彩
    char playerInput;
    #endregion

    //游戏场景的死循环，检测玩家输入相关循环
    while (true)
    {
        if (bossHp > 0)
        {
            //绘制的boos图标
            Console.SetCursorPosition(bossX, bossY);
            Console.ForegroundColor = bossColor;//设置boss颜色
            Console.WriteLine(bossIcon);
        }
        if (playHp > 0)
        {
            //绘制的player图标
            Console.SetCursorPosition(playX, playY);
            Console.ForegroundColor = playerColor;//设置player颜色
            Console.WriteLine(playerIcon);
            playerInput = Console.ReadKey(true).KeyChar;
            Console.SetCursorPosition(playX, playY);
            Console.WriteLine("  ");
            switch (playerInput)
            {
                case 'w':
                    --playY;
                    if (playY < 1) playY = 1;
                    if (playX == bossX && playY == bossY && bossHp > 0) ++playY;
                    break;
                case 's':
                    ++playY;
                    if (playY > h-7) playY = h-7;
                    if (playX == bossX && playY == bossY && bossHp > 0) --playY;
                    break;
                case 'a':
                    playX -= 2;
                    if (playX < 2) playX = 2;
                    if (playX == bossX && playY == bossY && bossHp > 0) playX+=2;
                    break;
                case 'd':
                    playX += 2;
                    if (playX > w - 4) playX = w - 4;
                    if (playX == bossX && playY == bossY && bossHp > 0) playX -= 2;
                    break;
            }

        }

    }
    break;
case 3:
```

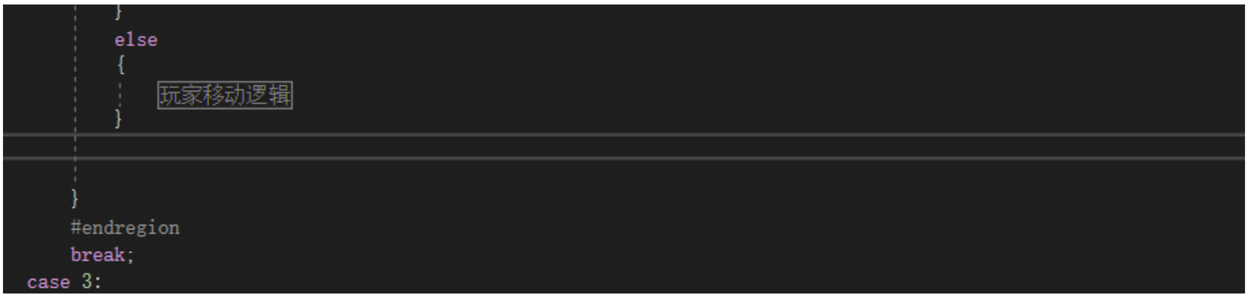
玩家和Boss战斗



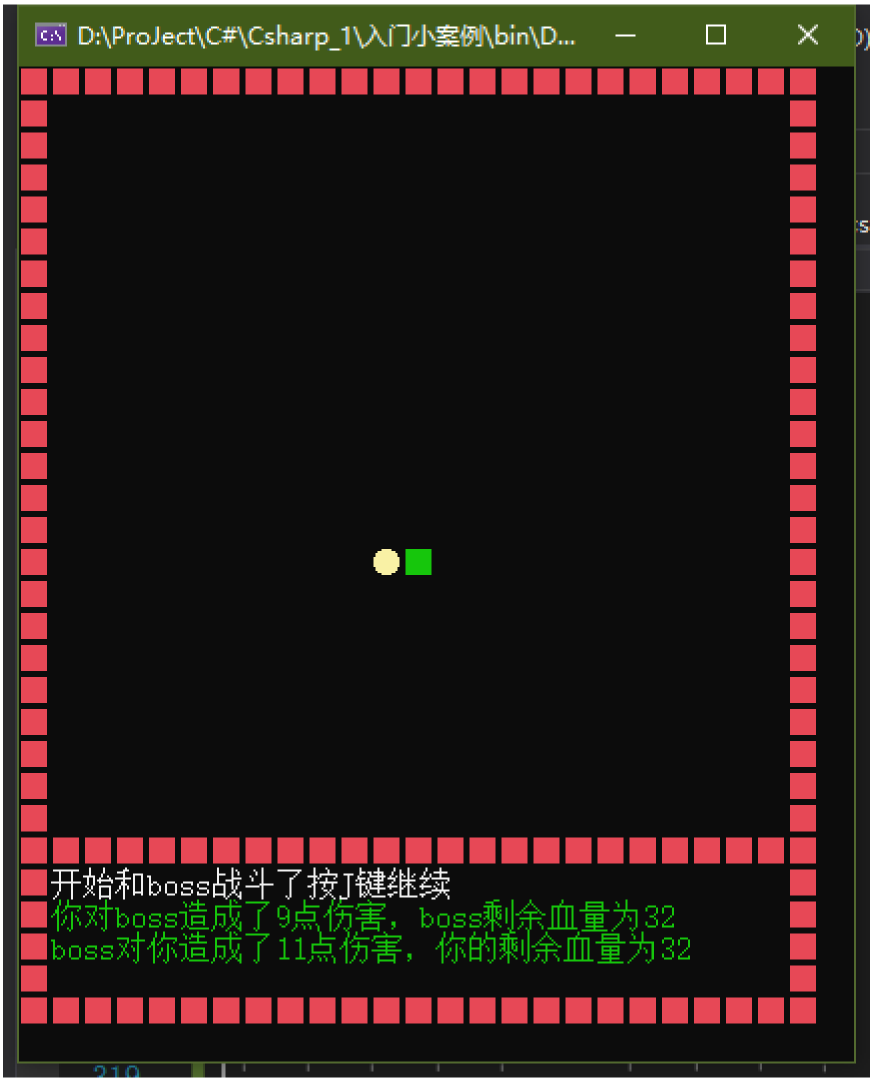
```
#region 游戏战斗逻辑
//游戏场景的死循环，检测玩家输入相关循环
while (true)
{
    if (bossHp > 0)
    {
        //绘制的boos图标
        Console.SetCursorPosition(bossX, bossY);
        Console.ForegroundColor = bossColor;//设置boss颜色
        Console.WriteLine(bossIcon);
```



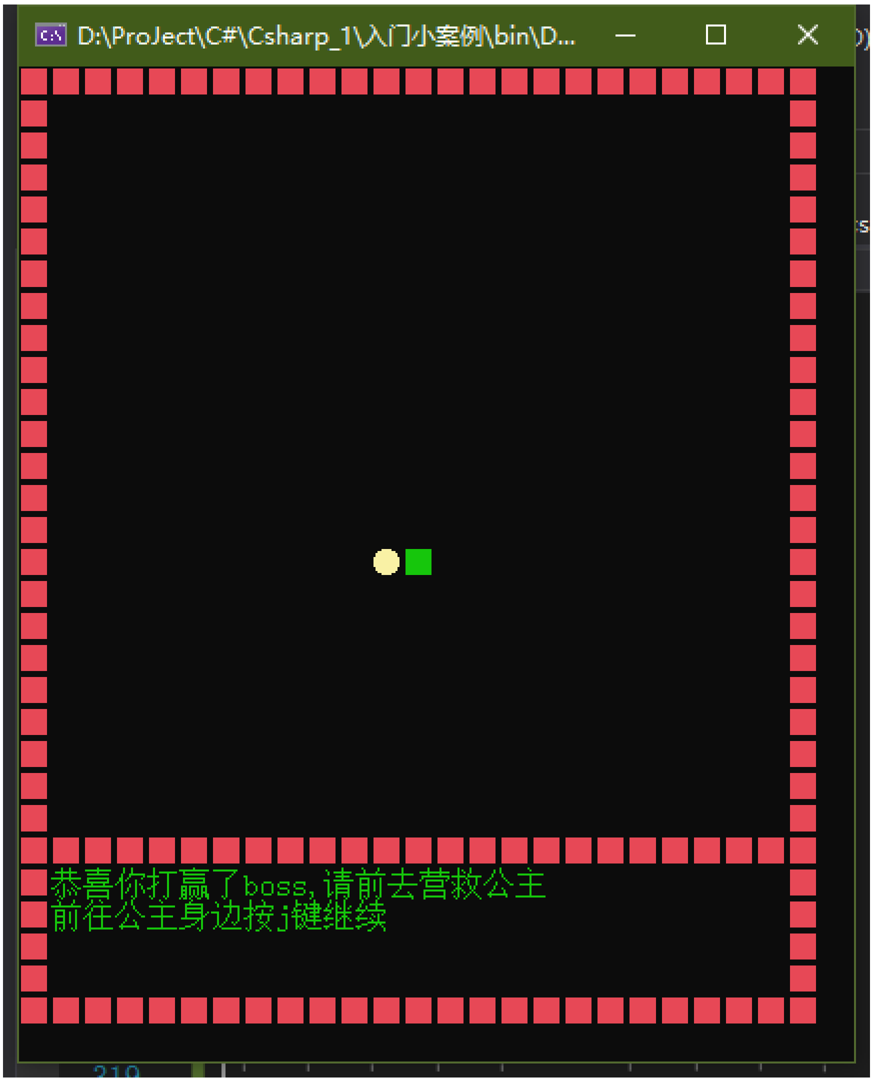
```
    Console.WriteLine(bossIcon);
}

//绘制的player图标                    绘制玩家
Console.SetCursorPosition(playX, playY);
Console.ForegroundColor = playerColor;//设置player颜色
Console.WriteLine(playerIcon);
playerInput = Console.ReadKey(true).KeyChar;   读取按键操作

if (isFight) 如果是战斗状态则进入
{
    if (playerInput == 'j')
    {
        if (playerHp <= 0)
        {
            //输掉了，直接进入游戏结束画面   玩家阵亡，跳转结束界面
            nowSceneId = 3;
            break;
        }
        else if (bossHp <= 0 )
        {
            //营救公主          怪物阵亡，营救公主
            //擦除boss
            Console.SetCursorPosition(bossX, bossY);
            Console.WriteLine("    ");
            isFight = false;
        }
        else       核心战斗逻辑
        {
            Random random = new Random();
            //玩家打怪物
            int atkP = random.Next(playerAtkMin, playerAtkMax);
            bossHp -= atkP;                                    玩家大怪物
            //print info
            Console.ForegroundColor = ConsoleColor.Green;
            //先擦除之前的信息
            Console.SetCursorPosition(2, h - 4);
            Console.Write("                                          ");
            Console.SetCursorPosition(2, h - 4);
            Console.Write("你对boss造成了{0}点伤害，boss剩余血量为{1}", atkP, bossHp);
            if (bossHp > 0)         怪物打玩家
            {
                //怪物打玩家
                atkP = random.Next(bossAtkMin, bossAtkMax);
                playerHp -= atkP;
                Console.SetCursorPosition(2, h - 3);
                Console.Write("                                          ");
                if (playerHp < 0)
                {
                    Console.SetCursorPosition(2, h - 3); 玩家阵亡
                    Console.WriteLine("很遗憾，你未能通过boss的试炼，战败！");
                }
                else
                {
                    Console.SetCursorPosition(2, h - 3);
                    Console.Write("boss对你造成了{0}点伤害，你的剩余血量为{1}", atkP, playerHp);
                }
            }
            else
            {       怪物阵亡

                Console.SetCursorPosition(2, h - 5);
                Console.Write("                                          ");
                Console.SetCursorPosition(2, h - 4);
                Console.Write("                                          ");
                Console.SetCursorPosition(2, h - 3);
                Console.Write("                                          ");
                Console.SetCursorPosition(2, h - 5);
                Console.Write("恭喜你打赢了boss,请前去营救公主");
                Console.SetCursorPosition(2, h - 4);
                Console.Write("前往公主身边按j键继续");
            }
        }
    }
```



```
        }
        else
        {
            玩家移动逻辑
        }

    }
    #endregion
    break;
case 3:
```

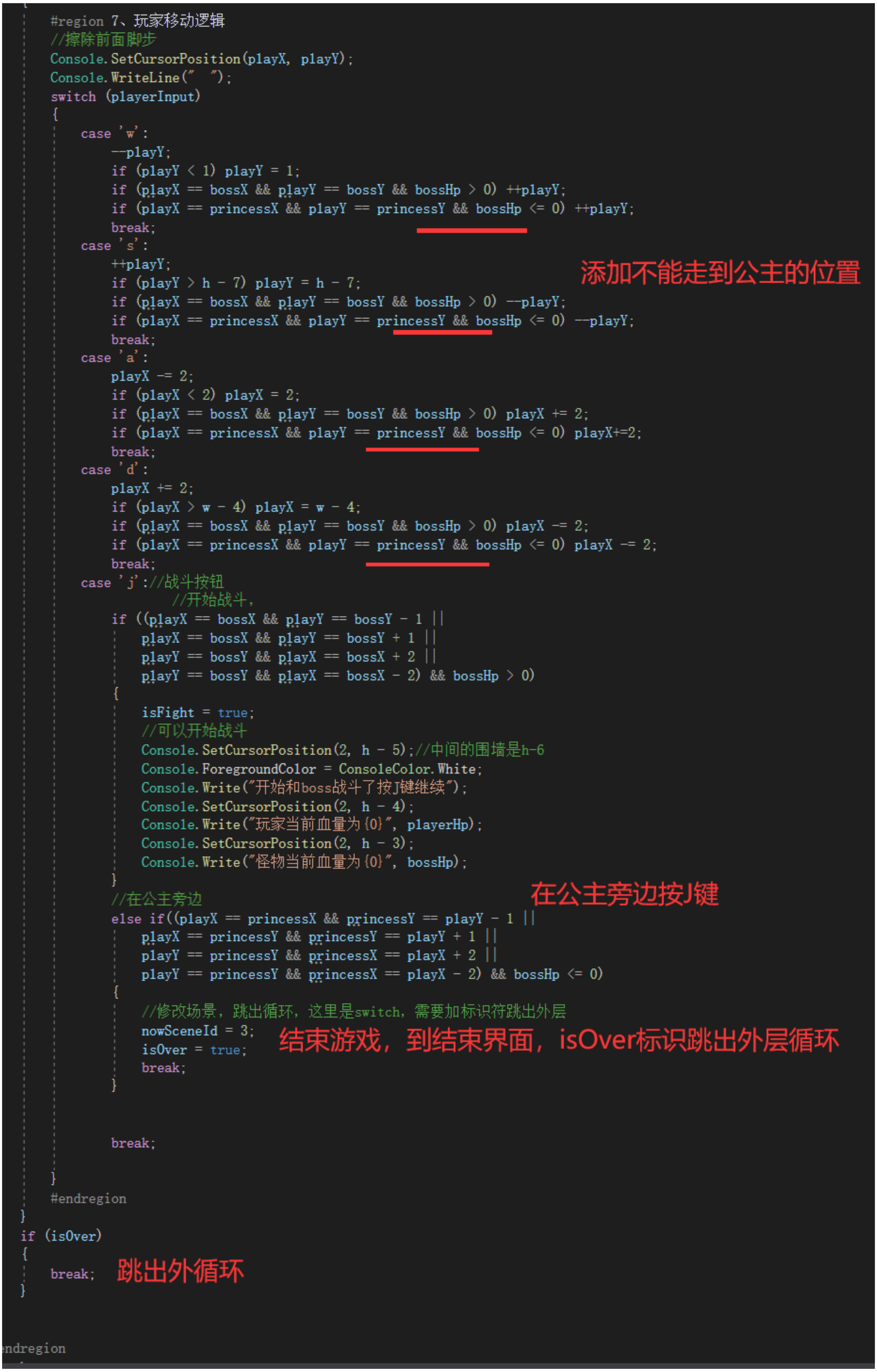


```
开始和boss战斗了按J键继续
你对boss造成了9点伤害，boss剩余血量为32
boss对你造成了11点伤害，你的剩余血量为32
```



```
恭喜你打赢了boss,请前去营救公主
前往公主身边按j键继续
```

营救公主逻辑:



```
        #region 7、玩家移动逻辑
        //擦除前面脚步
        Console.SetCursorPosition(playX, playY);
        Console.WriteLine("  ");
        switch (playerInput)
        {
            case 'w':
                --playY;
                if (playY < 1) playY = 1;
                if (playX == bossX && playY == bossY && bossHp > 0) ++playY;
                if (playX == princessX && playY == princessY && bossHp <= 0) ++playY;
                break;
            case 's':
                ++playY;
                if (playY > h - 7) playY = h - 7;
                if (playX == bossX && playY == bossY && bossHp > 0) --playY;
                if (playX == princessX && playY == princessY && bossHp <= 0) --playY;
                break;
            case 'a':
                playX -= 2;
                if (playX < 2) playX = 2;
                if (playX == bossX && playY == bossY && bossHp > 0) playX += 2;
                if (playX == princessX && playY == princessY && bossHp <= 0) playX+=2;
                break;
            case 'd':
                playX += 2;
                if (playX > w - 4) playX = w - 4;
                if (playX == bossX && playY == bossY && bossHp > 0) playX -= 2;
                if (playX == princessX && playY == princessY && bossHp <= 0) playX -= 2;
                break;
            case 'j'://战斗按钮
                        //开始战斗，
                if ((playX == bossX && playY == bossY - 1 ||
                    playX == bossX && playY == bossY + 1 ||
                    playY == bossY && playX == bossX + 2 ||
                    playY == bossY && playX == bossX - 2) && bossHp > 0)
                {
                    isFight = true;
                    //可以开始战斗
                    Console.SetCursorPosition(2, h - 5);//中间的围墙是h-6
                    Console.ForegroundColor = ConsoleColor.White;
                    Console.Write("开始和boss战斗了按J键继续");
                    Console.SetCursorPosition(2, h - 4);
                    Console.Write("玩家当前血量为{0}", playerHp);
                    Console.SetCursorPosition(2, h - 3);
                    Console.Write("怪物当前血量为{0}", bossHp);
                }
                //在公主旁边
                else if((playX == princessX && princessY == playY - 1 ||
                    playX == princessY && princessY == playY + 1 ||
                    playY == princessY && princessX == playX + 2 ||
                    playY == princessY && princessX == playX - 2) && bossHp <= 0)
                {
                    //修改场景，跳出循环，这里是switch，需要加标识符跳出外层
                    nowSceneId = 3;
                    isOver = true;
                    break;
                }


                break;

        }
        #endregion
    }
    if (isOver)
    {
        break;
    }


ndregion
```

添加不能走到公主的位置

在公主旁边按J键

结束游戏，到结束界面，isOver标识跳出外层循环

跳出外循环

最终修改代码见.cs文件