第 5 章

井字棋游戏

5.1 井字棋游戏介绍

井字棋游戏在九宫方格内进行,如果一方首先沿某方向(横、竖、斜)连成3子,则获 取胜利。本游戏有人人对战和人机对战两种模式。游戏开始时从图5-1(a)中选择对战模式,如 果是人人对战模式,两个玩家轮流下棋。如果是人机对战模式,游戏开始玩家(X方)先走, 计算机(O方)智能对弈下棋。游戏运行界面如图5-1(b)所示。



图 5-1 井字棋游戏运行效果

5.2 程序设计的思路

5.2.1 计算机智能下棋

在游戏中, pos 数组存储玩家、计算机的落子信息, 未落子处存储 0。X 方落子存储 1,

O方落子存储2。

由于人机对战,需要实现计算机的智能性,下面是为计算机设计的简单策略。

(1)如果有一步棋可以让计算机在本轮获胜,就选那一步。

(2)否则,如果有一步棋可以让玩家在本轮获胜,就选那一步。

(3)否则,计算机应该选择最佳空位置来走。最优位置就是中间那个,次优位置是四个 角,剩下的就都算第三优。

假设游戏中方格位置代号形式如图 5-2 所示。

0	1	2
3	4	5
6	7	8

图 5-2 方格位置

在程序中定义一个数组 BEST_MOVES 存储最佳方格位置,代码如下。

#按优劣顺序排序的下棋位置

var BEST_MOVES = [4, 0, 2, 6, 8, 1, 3, 5, 7];

按上述规则设计程序,就可以实现计算机的智能性。

5.2.2 井字棋输赢判断

井字棋输赢判断比较简单,这里横斜竖赢(即三颗同色的棋子排成一条直线)的情况只 有 8 种。通过遍历,就可以判断哪一方是否获胜。

```
//输赢判断
```

```
Iswin: function() {
 //判定(纵)
 for (var i = 0; i < 3; i++) {
   if (pos[i][0] == pos[i][1] && pos[i][1] == pos[i][2] &&
      pos[i][1] != 0)
    return pos[i][1];
 }
 //判定(横)
 for (var i = 0; i < 3; i++) {
  if (pos[0][i] == pos[1][i] && pos[1][i] == pos[2][i] &&
       pos[1][i] != 0)
    return pos[1][i];
 }
 //判定(斜)
 if (pos[0][0] == pos[1][1] && pos[1][1] == pos[2][2] &&
     pos[1][1] != 0) {
  return pos[1][1];
 if (pos[0][2] == pos[1][1] && pos[1][1] == pos[2][0] &&
```

```
pos[1][1] != 0) {
  return pos[1][1];
  }
  return 0;
},
```

5.3 关键技术

5.3.1 画布 canvas

微信小程序画布 canvas 组件的属性如表 5-1 所示。

属性名	类型	默认值	说明
canvas-id	String		canvas 组件的唯一标识符
disable-scroll	Boolean	false	当在 canvas 中移动时且有绑定手势事件时,禁止屏幕 滚动以及下拉刷新
bindtouchstart	EventHandle		手指触摸动作开始
bindtouchmove	EventHandle		手指触摸后移动
bindtouchend	EventHandle		手指触摸动作结束
bindtouchcancel	EventHandle		手指触摸动作被打断,如来电提醒、弹窗
bindlongtap	EventHandle		手指长按 500ms 之后触发, 触发了长按事件后进行移动不会触发屏幕的滚动
binderror	EventHandle		当发生错误时触发 error 事件, detail={errMsg:'something wrong'}

表 5-1 canvas 的属性

注意: canvas 标签默认宽度 300px、高度 225px。在同一页面中的 canvas-id 不可重复, 如果使用一个已经出现过的 canvas-id, 该 canvas 标签对应的画布将被隐藏并不再正常工作。 示例代码:

```
var context = wx.createCanvasContext('firstCanvas')
 context.setStrokeStyle("#00ff00")
 context.setLineWidth(5)
 context.rect(0,0,200,200)
 context.stroke()
 context.setStrokeStyle ("#ff0000")
 context.setLineWidth(2)
 context.moveTo(160,100)
 context.arc(100,100,60,0,2*Math.PI,true)
 context.moveTo(140,100)
 context.arc(100,100,40,0,Math.PI,false)
 context.moveTo(85,80)
 context.arc(80,80,5,0,2*Math.PI,true)
 context.moveTo(125,80)
 context.arc(120,80,5,0,2*Math.PI,true)
 context.stroke()
 context.draw()
}
```

5.3.2 响应 canvas 组件事件

})

canvas 组件可以响应手指触摸动作。可以在<canvas>中加上一些事件,观测手指的坐标。 【例 5-1】 观测手指触摸的坐标。 WXML代码如下。

```
//index.wxml
<canvas canvas-id="myCanvas" style="margin: 5px; border:1px solid
    #d3d3d3;"
    bindtouchstart="start" bindtouchmove="move" bindtouchend="end"/>
    <view hidden="{{hidden}}">
    Coordinates: ({{x}}, {{y}})
    </view>
```

其中, canvas-id 为当前画布的名称。bindtouchstart 是单击后触发, bindtouchend 是手指触摸动 作结束后触发, bindtouchmove 是手指触摸后移动时触发,并且可以传过来目前移动的参数 坐标。例如:

```
move: function( event ) {
    var xx=event.touches[0].x;
    var yy=event.touches[0].y;
    console.log(xx+", "+yy)
},
```

实现了手指触摸后移动时打印坐标。

Index.js 文件完整代码如下。

Page({
 data: {
 x: 0,y: 0,

```
hidden: true
 },
 start: function(e) {
   this.setData({
    hidden: false,
    x: e.touches[0].x,
    y: e.touches[0].y
   })
 },
 move: function(e) {
   this.setData({
    x: e.touches[0].x,
    y: e.touches[0].y
   })
 },
 end: function(e) {
   this.setData({
    hidden: true
   })
 }
})
```

当把手指放到 canvas 中移动,就会在下边显示出触碰点的坐标,如图 5-3 所示。



Coordinates: (139,28)

图 5-3 显示手指触碰点的坐标

在游戏开发中往往需要根据手指触摸、单击动作下棋、移动物体等,都是利用这些 bindtouchstart、bindtouchmove 和 bindtouchend 事件实现的。

5.4 程序设计的步骤

5.4.1 选择对战模式页面

新建一个微信小程序后,在 app.json 中修改原有的 pages 值,增加两个对战页面路径:

```
"pages/Three/Three" ,
"pages/computerThree/computerThree"
```

同时修改导航条标题文字为"井字棋夏敏捷开发",结果如下。

```
{
  "pages": [
  "pages/index/index",
  "pages/logs/logs",
  "pages/Three/Three",
  "pages/computerThree/computerThree"
], "window": {
  "backgroundTextStyle": "light",
  "navigationBarBackgroundColor": "#fff",
  "navigationBarTitleText": "井字棋夏敏捷开发",
  "navigationBarTextStyle": "black"
},
  "style": "v2",
  "sitemapLocation": "sitemap.json"
```

修改原有 index 页面,在 index.js 中增加事件处理函数。

```
//事件处理函数
drawComputerThree:function()
{
    wx.navigateTo({
        url: '../computerThree/computerThree' //跳转到人机对战游戏页面
    })
},
drawThree: function () {
    wx.navigateTo({
        url: '../Three/Three' //跳转到人人对战游戏页面
    })
}
```

在原有 index.wxml 视图文件中, 增加两个按钮, 同时绑定 tap 单击事件。

```
<button bindtap='drawThree'>人人对战井字棋</button><button bindtap='drawComputerThree'>人机对战井字棋</button>
```

结果如下。

```
<!--index.wxml-->
<view class="container">
<view class="userinfo">
<button wx:if="{{!hasUserInfo && canIUse}}" open-type="getUserInfo"
bindgetuserinfo="getUserInfo">获取头像昵称</button>
<block wx:else>
<image bindtap="bindViewTap" class="userinfo-avatar"
src="{{userInfo.avatarUrl}}" mode="cover"></image>
<text class="userinfo-nickname">{{userInfo.avatarUrl}}" mode="cover"></image>
<text class="userinfo-nickname">{{userInfo.nickName}}</text>
</block>
<button bindtap='drawThree'>人人对战井字棋</button>
<button bindtap='drawComputerThree'>人机对战井字棋游戏</button>
</view>
</view>
```

150 微信小程序游戏开发快速入门到实战

至此,可以实现跳转到不同游戏页面。

5.4.2 人人对战游戏页面

在微信小程序 pages 下新建文件夹 Three,在其下新建 page,命名为 Three,用来实现人人对战游戏页面。同时新建文件夹\images\png,其中存储 O.png 和 X.png 棋子图片。

1. Three.wxml 视图文件

```
<!--pages/Three/Three.wxml-->
<view class='title'>人人对战</view>
<canvas canvas-id='myCanvas' style='border:1rpx solid' bindtouchstart=
"touchStart"></canvas>
<text>{{info}}</text>
```

Three.wxml 文件内部仅添加画布,并设置触屏事件函数。<text>组件显示游戏输赢信息。

2. Three.js 文件

人人对战需要记录哪方走棋,这里使用 role 记录。游戏中使用值1代表 X 方,值2代表 O 方。二维数组 pos 存储落子情况。

游戏开始时,棋盘上没有棋子,所以 pos[i][j]元素值存储 0,代表此处无棋子。同时在画 布上绘制九宫格棋盘。

```
/**
* 生命周期函数—_监听页面加载
*/
onLoad: function(options) {
    //创建画布上下文
    this.ctx = wx.createCanvasContext('myCanvas')
    this.init();
    this.drawQipan();//画棋盘
    this.ctx.draw();
},
init: function() {
    for (var i = 0; i < 3; i++) {
}
</pre>
```

```
pos[i] = new Array();
   for (var j = 0; j < 3; j++) {
    pos[i][j] = 0; //0表示空的
   1
 }
},
//画九宫格棋盘
drawOipan: function() {
 let ctx = this.ctx
 ctx.beginPath();
 for (var i = 0; i <= 3; i++) {
  ctx.moveTo(i * 50, 0);
   ctx.lineTo(i * 50, 150);
   ctx.moveTo(0, i * 50);
   ctx.lineTo(150, i * 50);
 }
 ctx.stroke();
},
```

以下是触屏事件函数 touchStart (e),处理用户下棋落子。触屏事件中首先获取触屏位置 后换算成棋盘坐标(startx,starty),如果触摸位置已有棋子,则修改 info 变量,更新页面显 示"此位置已有棋子!"提示。修改落子位置 pos[startx][starty]元素值后重新绘制棋盘和棋子, 最后调用 Iswin()判断输赢。

```
touchStart: function(e) {
 var startx = Math.floor(e.touches[0].x / 50); //获取触屏位置
 var starty = Math.floor(e.touches[0].y / 50);
 if (isOver) {
  console.log("游戏已经结束! ");
   this.setData({
   info: "游戏已经结束! ",
  })
  return;
 }
 //此位置已有棋子
 if (pos[startx][starty] != 0) {
  console.log("此位置已有棋子!");
   this.setData({
   info: "此位置已有棋子! ",
  })
  return;
 }
 console.log("玩家走" + startx + ";" + starty);
 pos[startx][starty] = role; //修改落子位置元素值
 this.changeRole();
                     //改变角色
 this.drawQipan();
 this.drawQi();
 var info = "未赢";
```

```
if (this.Iswin() == 1) {
  info = "x方赢!!!! ";
  isOver = true;
  console.log("X方赢!!!! ");
 } else if (this.Iswin() == 2) {
  info = "o方赢!!!! ";
   isOver = true;
  console.log("O方赢!!!! ");
 }
 if (this.IsBlank() == false) //是否有空位置
 {
  info = "平局!!!! ";
  isOver = true;
  console.log("平局!!!! ");
 }
 this.setData({
  info: info,
 })
 if (isOver == true) {
  wx.showModal({
    title: '提示',
    content: info,
    success: function(res) {
      if (res.confirm) {
       console.log('用户单击确定')
      } else if (res.cancel) {
       console.log('用户单击取消')
     }
    }
   });
 }
},
```

IsBlank()函数判断是否还有空位置,以便判断和局。

```
IsBlank: function() {
  for (var i = 0; i < 3; i++) {
    for (var j = 0; j < 3; j++) {
        if (pos[i][j] == 0)
            return true;
        }
    }
    return false;
},</pre>
```

changeRole()改变下棋者的角色。

changeRole: function() {
 if (role == 1) { //如果是x方换成O方
 role = 2;

```
} else { //如果是o方换成x方
role = 1;
}
```

drawQi ()按 pos 数组记录的下棋落子情况画棋子。

```
drawQi: function() { //画棋子
  let ctx = this.ctx;
  for (var i = 0; i < 3; i++) {
    for (var j = 0; j < 3; j++) {
        if (pos[i][j] == 1)
            ctx.drawImage('/images/png/X.png', i * 50, j * 50, 50, 50);
        if (pos[i][j] == 2)
            ctx.drawImage('/images/png/0.png', i * 50, j * 50, 50, 50);
     }
    }
    ctx.draw();
},</pre>
```

Iswin()按照8种赢的情况,依次遍历判断。

```
//输赢判断
Iswin: function() {
    //见前文
},
/**
* 用户单击右上角分享
*/
onShareAppMessage: function() {
  }
})
```

5.4.3 人机对战游戏页面

在微信小程序 pages 下新建文件夹 computerThree, 在其下新建 page, 命名为 computerThree, 用来实现人人对战游戏页面。

1. computerThree.wxml 视图文件

```
<!--pages/computerThree/computerThree.wxml-->
<view class='title'>人机对战</view>
<canvas canvas-id='myCanvas' style='border:1rpx solid' bindtouchstart=
"touchStart"></canvas>
<text>{{info}}</text>
```

computerThree.wxml 文件内部仅添加画布,并设置触屏事件函数。<text>组件显示游戏输赢信息。