

《计移动项目开发》课程

实验报告

姓 名： 管书豪 学号： 2017210218046 班 级： 信息 171

指导教师： 马国春 实验日期： 2019/9/23 实验成绩： _____

实验名称： 制作猜数字小程序 _____

小程序介绍：这是一个猜数字游戏，当小程序开始运行时，系统后台会随机生成一个 0~9 的数字，我们的任务就是猜对那个数字是多少。当我们点击初始页面上的数字时，如果系统后台数字比我们的数字大，则跳出提示语句，too little；如果系统后台数字比我们的数字小，则跳出提示语句，too large；如果选择正确，则跳转进入 result 界面，显示选择次数，并且显示恭喜语句。并且点击数字时，数字的图片会从黑色变成红色。

小程序演示：



点击



显示 too little
点击



显示 too large
点击



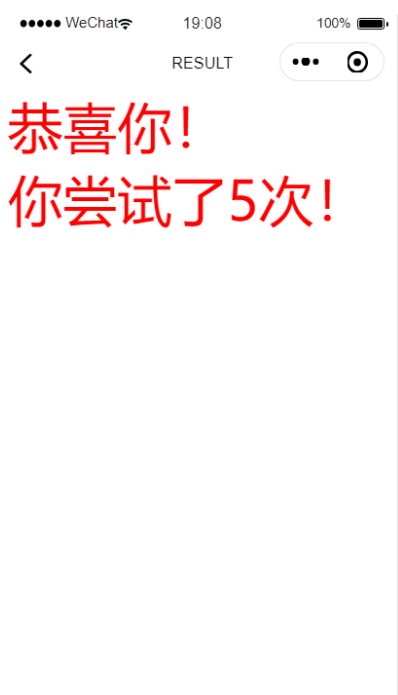
点击



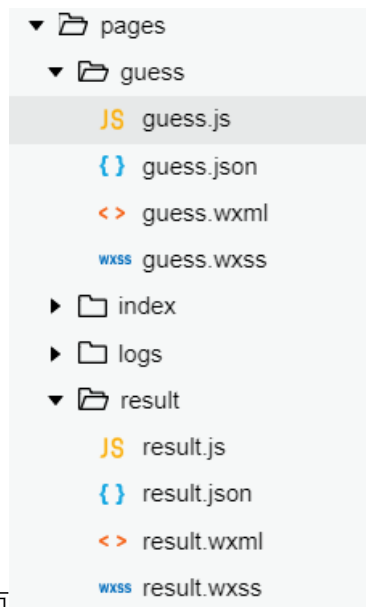
点击



跳转



代码分析：



共有 2 个主要界面

首先我们要在 app.js 中设置一个全局变量 count

```
//app.js
App({
  count: 0,
  onLaunch: function () {
    console.log('welcome')
  },
  globalData: {
    userInfo: null,
  }
})
```

方便记录点击次数。用来最后输出。

这是 GUESS.js 中的点击逻辑

```
click:function(e){
  console.log(e.currentTarget.id)
  var temp=this.data.d;
  temp[e.currentTarget.id]='red';
  this.data.times = this.data.times+1;
  app.count = app.count+1
  this.setData({
    d:temp
  })
  if(e.currentTarget.id>this.data.targetnumber){
    console.log("count:",app.count)
    wx.showToast({

      title: 'too large',
      icon:'none'
    })
  }else{
    if (e.currentTarget.id == this.data.targetnumber){
      console.log("count:", app.count)
      wx.showToast({
        title:'Congraduation',
        duration:3000,
        success:function(){
          wx.navigateTo({
            url: '/pages/result/result',
            events:{"ttt":12345}
          })
        }
      })
    }
  }

  }
  else{
    console.log("count:", app.count)
    wx.showToast({
      title: 'too little',
      icon: 'none'
    })
  }
}
},
```

```
app.count = app.count+1
```

这一句语句表示了，每当点击一次按钮，那么全局变量 count 就会加一。

为了获取全局变量的值。我们需要在每一个页面加上一个语句

```
var app=getApp()
```

有了这个我们就可以在其他页面中引用定义在 app.js 中的全局变量，比如说在点击按钮的时候让全局变量 count+1。

但是全局变量不能直接在 wxml 文件中引用，比如说{{app.count}}，如果这样子用这个全局变量的话，数据并不会在最终页面中显示出来。所以我们需要在 result 页面加载的之前，在 result.js 中的 data 中定义一个 count 值为 null，然后在 result 页面要显示之前把全局变量中的 count 值赋给 result 中的 count，用下方的程序

```
/**
 * 生命周期函数--监听页面加载
 */
onLoad: function (options) {
  this.setData({
    count: app.count
  })
},
```

最终显示页面中的代码是：

```
<view>
  <text class="t2">恭喜你! </text>
  <view>
    <text class="t2">你尝试了{{count}}次! </text>
  </view>
</view>
```

这是 guess.JS 中的数据

```
/**
```

```

    * 页面的初始数据
    */
    data: {
      //d:1,
      mycolor:['black','red'],
      d: ['black', 'black', 'black', 'black', 'black', 'black', 'black', 'black',
'black', 'black'],
      clicked:[0,0,0],
      targetnumber:Math.round(Math.random()*9),
      times:0
    },

```

每当点击一个按钮，d 中的 id 位置的‘black’就会变成‘red’，以替换图形，让按钮变成红色。

全局变量不能直接在 wxml 文件中引用，比如说{{app.count}}，如果这样子用这个全局变量的话，数据并不会在最终页面中显示出来。所以我们需要在 result 页面加载的之前，在 result.js 中的 data 中定义一个 count 值为 null，然后在 result 页面要显示之前把全局变量中的 count 值赋给 result 中的 count。

下面是 result.js 文件中的所有代码

```

// pages/guess/guess.js
var app=getApp()
Page({

  /**
   * 页面的初始数据
   */
  data: {
    //d:1,
    mycolor:['black','red'],
    d: ['black', 'black', 'black', 'black', 'black', 'black', 'black', 'black',
'black', 'black'],
    clicked:[0,0,0],

```



```

    targetnumber:Math.round(Math.random()*10),
    times:0
  },

  /**
   * 生命周期函数--监听页面加载
   */
  onLoad: function (options) {

    console.log('welcome')
  },
  click:function(e){
    console.log(e.currentTarget.id)
    var temp=this.data.d;
    temp[e.currentTarget.id]='red';
    this.data.times = this.data.times+1;
    app.count = app.count+1
    this.setData({
      d:temp
    })
    if(e.currentTarget.id>this.data.targetnumber){
      console.log("count:",app.count)
      wx.showToast({

        title: 'too large',
        icon:'none'
      })
    }else{
      if (e.currentTarget.id == this.data.targetnumber){
        console.log("count:", app.count)
        wx.showToast({
          title:'Congraduation',
          duration:3000,
          success:function(){
            wx.navigateTo({
              url: '/pages/result/result',
              events:{"ttt":12345}
            })
          }
        })
      }
    }

  }
  else{

```

```
        console.log("count:", app.count)
        wx.showToast({
            title: 'too little',
            icon: 'none'
        })
    }
}
},

/**
 * 生命周期函数--监听页面初次渲染完成
 */
onReady: function () {

},

/**
 * 生命周期函数--监听页面显示
 */
onShow: function () {
    wx.showToast({
        title: 'start',
    })
},

/**
 * 生命周期函数--监听页面隐藏
 */
onHide: function () {

},

/**
 * 生命周期函数--监听页面卸载
 */
onUnload: function () {

},

/**
 * 页面相关事件处理函数--监听用户下拉动作
 */
onPullDownRefresh: function () {
```

```
    },  
  
    /**  
     * 页面上拉触底事件的处理函数  
     */  
    onReachBottom: function () {  
  
    },  
  
    /**  
     * 用户点击右上角分享  
     */  
    onShareAppMessage: function () {  
  
    }  
  })
```