

1.技术基础

微信小程序开发基础

小程序技术背景

小程序在技术架构上非常清晰易懂。JS 负责业务逻辑的实现，而表现层则 WXML 和 WXSS 来共同实现就是一种微信定义的模板语言，而后者类似 CSS。所以对于擅长前端开发，或者 WEB 开发的广大开发者的开发可谓降低了不少门槛

小程序框架基础

框架程序包含一个描述整体程序的 app 和多个描述各自页面的 page。一个框架程序主体部分由三个文件放在项目的根目录：app.js 小程序(全局)逻辑 app.json 小程序(全局)公共设置，决定页面文件的路径、配置网络超时时间、设置多 tab 等 app.wxss 小程序公共(全局)样式表

项目文件类型与作用

js 页面逻辑

wxml 页面结构，框架设计的一套标签语言，结合基础组件、事件系统，可以构建出页面的结构。

wxss 是一套样式语言，用于描述 WXML 的组件样式。用来决定 WXML 的组件应该怎么显示。

json 页面配置

es6 函数式编程基础

json 数据格式

json是一种与语言无关的数据交换的格式，使用Json的格式与解析方便的可以表示一个对象信息，json有①对象格式：{"key1":obj,"key2":obj,"key3":obj...}、②数组/集合格式：[obj,obj,obj...]。json是 JavaScript Notation 的首字母缩写，单词的意思是javascript对象表示法，这里说的json指的是类似于javascript对象格式，目前这种数据格式比较流行，逐渐替换掉了传统的xml数据格式。与json对象不同的是，json数据名称和字符串值需要用双引号引起来，用单引号或者不用引号会导致读取数据错误。json的另外一个数组，和javascript中的数组字面量相同

小程序样式、布局与事件响应

样式与布局基础

```
width: fit-content;
font-size:20px; /*设置文字字号*/
color:red; /*设置文字颜色*/
font-weight:bold; /*设置字体加粗*/
```

border:1px solid red;/*添加边框样式（粗细为1px， 颜色为红色的实线）*/

font-family:"宋体";/*设置字体为宋体*/

font-style:italic; /*文字排版--斜体*/

text-decoration:underline; /*文字排版--下划线*/

text-decoration:line-through; /*文字排版--删除线*/

text-indent:2em; /*段落排版--缩进*/

line-height:1.5em; /*段落排版--行间距（行高）*/

letter-spacing:50px; /*段落排版--中文字间距*/

word-spacing:50px; /*字母间距*/

text-align:center; right ; left ; /*段落排版--对齐*/

display:inline-flex; /*将对象作为内联块级弹性伸缩盒显示*/

display:block; /*设置为块状元素*/

display:inline; /*设置为内联元素*/

display:inline-block; /*设置为内联块状元素*/

word-break:keep-all; /* 不换行 */

white-space:nowrap; /* 不换行 */

vertical-align:middle; /*把此元素放置在父元素的中部。*/

border-style（边框样式）常见样式有：（border-color,border-width）边框相关设置

dashed（虚线）| dotted（点线）| solid（实线）。

border-bottom border-top border-right border-left 上下左右线单独设置

box-sizing: border-box; //当使用padding的时候不影响大小

padding-top padding-right padding-bottom padding-left

margin-top margin-right margin-bottom margin-left (margin:10px 10px 10px 10px; top、right、bottom、left)

浮动与定位

static：元素框正常生成。块级元素生成一个矩形框，作为文档流的一部分，行内元素则会创建一个或多个于其父元素中，static是position的默认值。

relative：元素框偏移某个距离。元素仍保持其未定位前的形状，它原本所占的空间仍保留。

absolute：元素框从文档流中完全删除，并相对于其包含块定位，包含块可能是文档中的另一个元素或块。对于absolute来说，包含块是离当前元素最近的position为absolute或relative的父元素，如果父元素absolute或relative布局的元素，那么包含块就是根元素。使用position布局后，元素原先在正常文档流中间会关闭，就好像该元素原来不存在一样。元素定位后生成一个块级框，不论原来它在正常流中生成何

fixed：元素框的表现类似于将position设置为absolute，不过其包含块是视窗本身

Flex 布局基础

Flex 是 Flexible Box 的缩写，意为"弹性布局"，用来为盒状模型提供最大的灵活性。容器默认存在两根轴（main axis）和垂直的交叉轴（cross axis）。主轴的开始位置（与边框的交叉点）叫做main start，结束位置叫做main end；交叉轴的开始位置叫做cross start，结束位置叫做cross end。项目默认沿主轴排列。单个项目占据的主轴空间叫做main size，占据的交叉轴空间叫做cross size。

flex-direction属性决定主轴的方向（即项目的排列方向）。

row（默认值）：主轴为水平方向，起点在左端。

row-reverse：主轴为水平方向，起点在右端。

column：主轴为垂直方向，起点在上沿。

column-reverse：主轴为垂直方向，起点在下沿。

justify-content属性定义了项目在主轴上的对齐方式。

flex-start（默认值）：左对齐

flex-end：右对齐

center：居中

space-between：两端对齐，项目之间的间隔都相等。

space-around：每个项目两侧的间隔相等。所以，项目之间的间隔比项目与边框的间隔大一倍。flex-grow属性定义项目的放大比例，默认为0，即如果存在剩余空间，也不放大。

flex-shrink属性定义了项目的缩小比例，默认为1，即如果空间不足，该项目将缩小

事件响应基础

事件严格来说并不是js中的一个概念，而是在视图层，是视图层到逻辑层的通讯方式。它可以将用户的逻辑层进行处理。也就是说，事件是在wxml中，通过绑定在组件上，当出现触发事件，就会执行逻辑层中处理函数。对应的这个处理函数，就是事件响应，写在页面的js文件中。通过绑定，能够从wxml页面中获取信息，如id, dataset, touches等

小程序组件

组件定义与属性视图容器组件

组件的定义

配置文件

要编写一个自定义组件，首先需要在json文件中进行自定义组件声明（将 component 字段设为true）。使用已注册的自定义组件前，首先要在页面的json文件中进行引用声明。此时需要提供每个自定义组件对应的自定义组件文件路径（标签名称只能是小写字母、中划线和下划线的组合，组件根目录名不能以“_”结尾）。

wxml文件

在组件模板中可以提供一个节点，用于承载组件引用时提供的子节点。默认情况下，一个组件的wxml中只能有一个slot。需要使用多slot时，可以在组件js中声明启用options: {multipleSlots: true}，以不同的name来区分slot。

view 组件

主要属性：

flex-direction：主要两个特性"row"横向排列"column"纵向排列

justify-content 主轴的对齐方式（如果flex-direction为row则主轴就是水平方向）

可选属性 ('flex-start', 'flex-end', 'center', 'space-between', 'space-around')

align-items 侧轴对齐方式如果flex-direction为row则侧轴就是垂直方向)

可选属性 ('flex-start', 'flex-end', 'center')

scroll-view 组件

Scroll-view 是负责可滚动视图区域的一个组件，称为滚动视图。当在一个屏幕的像素显示不下绘制的 UI 以采用滑动的方式，使控件显示

滑块视图组件

滑块视图容器，用于展示图片，可以通过用户拖拽和设置自动切换属性控制图片的切换

基础组件

icon 组件

这是微信小程序自带的图标组件

text 组件

在使用小程序时，如果想通过长按文字进行复制文字内容，就要把该内容写在text中

progress 组件

这是进度条组件

表单组件

button 组件

size: default、mini——default为块级按钮、mini为小按钮

type: primary、default、warn——primary提交成功、default默认灰色、warn警告色

plain: true、false——按钮是否镂空，背景色透明

disabled: true、false——是否禁用

loading: true、false——名称前是否带 loading 图标

input 组件

输入框：该组件是原生组件，使用的时候要注意相关的设置

属性名 类型 默认值 说明

value String 输入框的内容

type String text input的类型，有效值：text,number,idcard,digit,time, date

password Boolean false 是否是密码类型

placeholder String 输入框为空时占位符

placeholder-style String 指定placeholder的样式

placeholder-class String input-placeholder 指定placeholder的样式类
disabled Boolean false 是否禁用
maxlength Number 140 最大输入长度，设置为0的时候不限制最大长度
auto-focus Boolean false 自动聚焦，拉起键盘。页面中只能有一个input设置
auto-focus属性
focus Boolean false 使得input获取焦点
bindchange EventHandle 输入框失去焦点时，触发bindchange事件，event.
detail={value:value}
bindinput EventHandle 除了date/time类型外的输入框，当键盘输入时，触发 input事件，event.detail=
{value:value}，处理函数可以直接return一个字符串，将替换输入框的内容。
bindfocus EventHandle 输入框聚焦时触发，event.detail = {value:value}
bindblur EventHandle 输入框失去焦点时触发，event.detail =
{value:value}

radio 组件

单组件单选框

导航组件

导航系统起着组织内容和功能的作用，让它们按照产品的信息架构图进行连接，展现在用户面前，导航内容和功能组织成了一个完成的有结构的系统，有时我们需要把更多的内容放置在导航栏的位置，因此需要面板

导航面板是导航栏的一个扩展，从导航栏部分拖拽出导航面板，展示更多的入口

小程序功能 API

数据缓存功能

微信小程序可以通过wx.setStorage (wx.setStorageSync)、wx.getStorage (wx.getStorageSync)、wx.clearStorage (wx.clearStorageSync) 对本地缓存进行设置、获取和清理。本地缓存最大为10MB

音频播放功能

调用wx.createInnerAudioContext()接口可以返回一个InnerAudioContext对象，然后就可以使用这个对象定义它的属性

网络请求功能

调用wx.request()接口，可以请求网络接口，不过微信小程序有要求，只能发起https请求
每个微信小程序需要事先设置通讯域名，小程序只可以跟指定的域名进行网络通信。包括普通 HTTPS 请求 (wx.request)、上传文件 (wx.uploadFile)、下载文件 (wx.downloadFile) 和 WebSocket 通信 (wx.connectSocket) 。

从基础库 2.4.0 开始，网络接口允许与局域网 IP 通信，但要注意 不允许与本机 IP 通信

2.总体设计与详细设计

总体设计

一共三个页面

主页：负责进入 tabbar 页面，tabbar 页面有两个页面,首页和记录页

首页：可点击按钮生成随机数,然后输入数字,点击确定按钮进行判断并提示偏大或偏小
能显示猜的次数,点击重置按钮可重置随机数并清空输入和猜的次数

记录页：显示每次猜对的随机数,次数,时间,标有序号

页面统一采用暖色调, 主体部分使用 绝对定位 , 内部使用 flex 布局

使用本地存储数据 (wx.getStorageSync , wx.setStorageSync)

详细设计

主页



欢迎来到猜数字小游戏



点击生成随机数

填个数试试

确定

猜的次数

重置

首页

记录

记录页

我的战绩

清空

1

2

3

...

3.代码实现

页面与布局

log.wxml

```

<view class="body">
  <!-- title -->
  <view class="title">
    <text>我的战绩</text>
    <button bindtap="cleanRecord">清空</button>
  </view>
  <!-- list -->
  <scroll-view class="recordlist" enable-flex="true" scroll-y="true">
    <view class="recorditem" wx:for="{{recordList}}" wx:key="item">
      <view class="recordinfo">{{index + 1}}</view>
      <view class="recordinfo">
        >随机数 <text class="randomnum">{{recordList[index].randomNum}}</text>
      </view>
      <view class="recordinfo">
        >猜了 <text class="times">{{recordList[index].times}}</text> 次</view>
      >
      <view class="recordinfo">{{recordList[index].time}}</view>
    </view>
  </scroll-view>
  <!-- 提示 -->
  <view class="hint {{hadrecord?'hadrecord':''}}">
    <text>目前啥也没有</text>
  </view>
</view>

```

log.wxss

```
/* 可视区域 */
.body {
  position: fixed;
  background-color: #fdebc3;
  width: 100%;
  height: 100vh;
}

/* title */
.title {
  top: 0;
  width: 100%;
  height: 80rpx;
  line-height: 80rpx;
  background-color: #fdebc3;
  box-shadow: 0 2px 4px #ffa551;
  text-align: center;
  font-size: 2rem;
  font-weight: 800;
  color: #e97d4c;
  z-index: 10;
}

/* 清空按钮 */
.title button {
  width: 120rpx !important;
  padding: 0;
  line-height: 60rpx;
  position: absolute;
  top: 12rpx;
  right: 10rpx;
  background-color: #ffb168;
  color: #702f11;
}

/* scroll-view 必须给定高度才会实现滚动 */
.recordlist {
  height: 96%;
  position: absolute;
  top: 58%;
  left: 50%;
  transform: translate(-50%, -50%);
  width: 94%;
  margin: 0 auto 100rpx;
  z-index: -1;
}

/* 隐藏滚动条 */
::-webkit-scrollbar {
  width: 0rpx;
  height: 0rpx;
  background-color: transparent;
}

/* 每条记录 */
.recorditem {
  display: flex;
```

```

    flex-direction: row;
    align-items: center;
    justify-content: space-around;
    width: 98%;
    height: 80rpx;
    background-color: #ffcf98;
    box-shadow: 2px 4px 6px #e97d4c;
    margin: 16rpx auto;
    border-radius: 10px;
}

/* 变量高亮 */
.recordinfo:nth-of-type(1),
.randomnum,
.times {
    font-weight: 800;
    color: #e97d4c;
}

.hint {
    position: absolute;
    top: 40%;
    left: 50%;
    transform: translate(-50%, -50%);
    font-size: 2rem;
    color: #ffbf8a;
    font-weight: 600;
    display: block;
}

.hadrecord {
    display: none;
}

```

vivew 组件,利用绝对定位把标题置顶

button 组件,利用 bindtap 绑定后台 cleanRecord 方法,点击后调用后台 cleanRecord() 方法弹框提示, 点击确定后把记录清空

记录的每一项里的内容采用 flex 弹性布局

scroll-view 组件,当有很多记录时,设置了 scroll-y="true",可沿 y 方向滚动

数据判断

```

Page({
  data: {
    recordList: [],
    hadrecord: false,
  },
  // onShow() 每次打开页面都会调用一次
  onShow: function (options) {
    this.setData({
      recordList: wx.getStorageSync("record"),
    });
    if (this.data.recordList.length > 0) {
      this.setData({
        hadrecord: true,
      });
      return;
    }
  },
  // 清空记录
  cleanRecord() {
    let that = this;
    if (this.data.recordList.length == 0) {
      wx.showToast({
        title: "您还没有任何记录",
        icon: "none",
        duration: 1500,
      });
    } else {
      wx.showModal({
        title: "提示",
        content: "这将清空所有记录",
        success: function (res) {
          if (res.confirm) {
            wx.setStorageSync("record", []);
            that.setData({
              recordList: wx.getStorageSync("record"),
            });
          } else {
          }
        },
      });
      this.setData({
        hadrecord: false,
      });
    }
  },
  //
});

```

利用微信小程序自带的 `onShow()` 方法，每次打开页面都会调用一次，调用小程序自带的 api：

`wx.getStorageSync('record')`，读取存储,赋值给 `recordList[]`，这样就能在每次打开页面显示所有的记录
 点击清空按钮,调用 `cleanRecord()` 方法，先判断存储里是否为空，是则弹窗提示为空，若不为空则弹窗提示是否清空，确认则调用小程序自带的 api：`wx.getStorageSync()` 把 `record` 设置为空，并利用 `setData()` 赋值给 `recordList`，实现清空

4.功能测试



我的战绩

清空

1 随机数 **9** 猜了 **2** 次 2020/12/01 20:26:50

2 随机数 **78** 猜了 **1** 次 2020/12/01 20:26:59

3 随机数 **51** 猜了 **3** 次 2020/12/01 20:27:16

提示

这将清空所有记录

取消

确定



我的战绩

清空

目前啥也没有



首页



记录