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
程序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、窗
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 JavaScrip
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的另外一个数据
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者块。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种
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
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
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
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。它可以将用户的行
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
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“
缀）。
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
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

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，展现在在用户面前，导航
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
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
（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

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

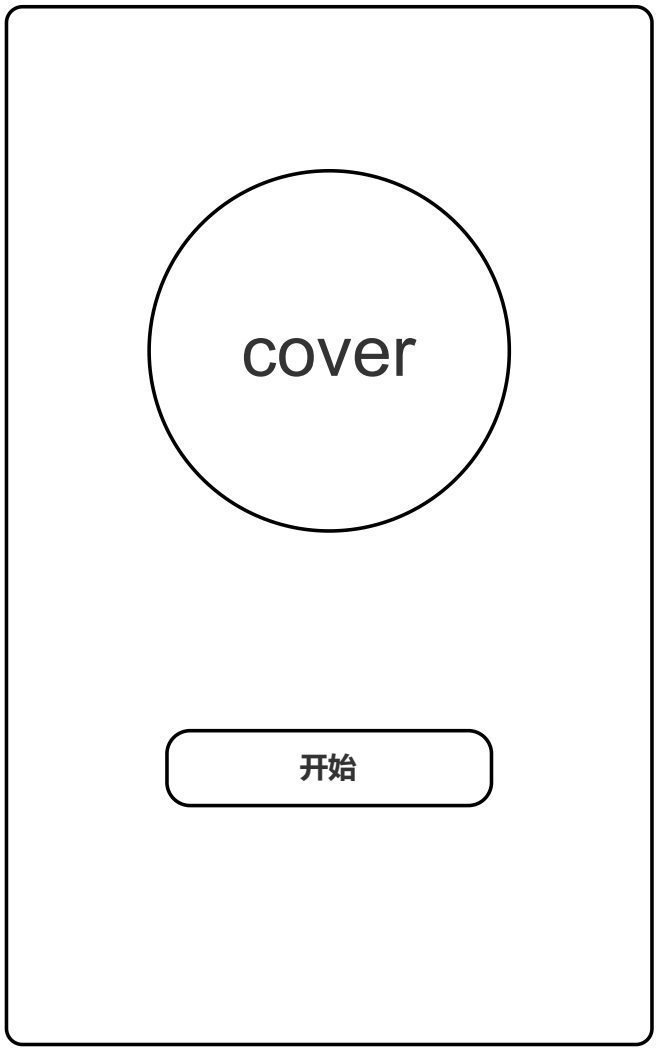
记录页 ： 显示每次猜对的随机数,次数,时间,标有序号

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 ）

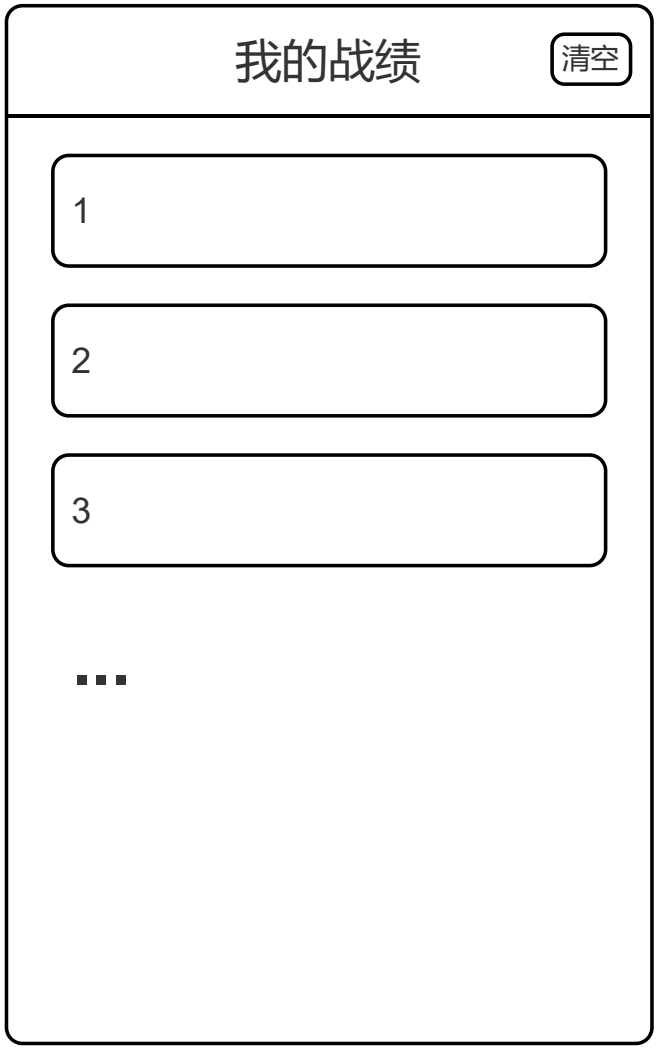
详细设计

主页



首页

记录页



3.代码实现

页面与布局

```
<button class="sure" bindtap="sure">确定</button>
<view class="times"> 猜的次数 : {{times}} </view>
<button class="reset" bindtap="reset">重置</button>
```

```css
.main {
  position: absolute;
  top: 46%;
  left: 50%;
  transform: translate(-50%, -50%);
  display: flex;
  flex-direction: column;
  justify-content: center;
}

.input,
.createRandNum,
.sure,
.reset,
.times {
  background-color: #ffd79c;
  border-radius: 10px;
  margin: 12px 0;
  box-shadow: 2px 5px 6px #e97d4c;
  transition-duration: 0.3s;
}

.createRandNum:active,
.sure:active,
.reset:active {
  box-shadow: 2px 2px 2px #e97d4c;
  transform: translateY(5px);
}

.input,
.times {
  text-align: center;
  color: #ff641c;
  width: 340rpx;
  height: 74rpx;
  line-height: 74rpx;
  font-weight: 600;
}

/* 答对正确的提示框(默认隐藏) */
.hint {
  position: absolute;
  top: 45%;
  left: 50%;
  width: 460rpx;
  height: 100rpx;
  background-color: #fff1c1;
  border-radius: 20px;
  border: 6px solid #ffba79;
  display: none;
}

.hintcontent {
  color: #e97d4c;
  width: inherit;
  height: inherit;
  padding: 0 40rpx;
```

```
  position: relative;
  font-size: 1.8rem;
  font-weight: 600;
  display: flex;
  flex-direction: row;
  align-items: center;
  justify-content: space-around;
  transition: all 1s;
}

/* 关闭按钮 */
.closehint {
  position: absolute;
  top: -20rpx;
  right: -32rpx;
  width: 44rpx;
  height: 44rpx;
  border: 7px solid #ffa754;
  border-radius: 50%;
  background-color: #ffcc9c;
  overflow: hidden;
  transition: all 2s ease-in-out;
  z-index: 10;
}

/* 关闭按钮内的两条线(叉×) */
.closehint .linel,
.closehint .liner {
  width: 100%;
  height: 8rpx;
  border-radius: 4px;
  background-color: #ff891b;
  position: absolute;
  top: 38%;
  border: none;
  box-shadow: 2rpx 2rpx 4px #f1752d;
}

.closehint .linel {
  left: 0;
  transform: rotate(45deg);
}

.closehint .liner {
  right: 0;
  transform: rotate(135deg);
}

/* 回答正确后显示提示框 */
.correct {
  display: flex;
  transform: translate(-50%, -50%);
}
```

```
"tabBar": {
    "color": "#000000",
    "selectedColor": "#e97d4c",
    "borderStyle": "black",
    "backgroundColor": "#fdebc3",
    "list": [{
      "selectedIconPath": "/imgs/homeselected.png",
      "iconPath": "/imgs/home.png",
      "pagePath": "pages/home/home",
      "text": "首页"
    }, {
      "selectedIconPath": "/imgs/logsselected.png",
      "iconPath": "/imgs/logs.png",
      "pagePath": "pages/logs/logs",
      "text": "记录"
    }]
  }
```

`tabbar` 组件，导航首页和日志页

`button` 组件，点击确定调用后台 `sure()` 方法进行判断，偏大或偏小弹框提示,猜对后弹框提示答对，并计入一次记录

利用 `mustache` 表达式动态显示猜的次数

点击重置按钮,调用后台 `reset()` 方法，重置随机数并清空输入和次数

数据判断

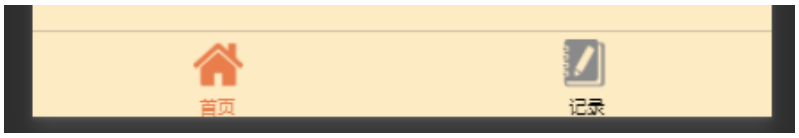
```
// 确定
    sure() {
        if (this.data.randomNum == "") {
            wx.showToast({
                title: '请先点击生成随机数',
                icon: "none",
                duration: 2000
            })
        } else {
            if (this.data.guessNum == "") {
                wx.showToast({
                    title: '请输入数字',
                    icon: "none"
                })
                return;
            }
            // 匹配 1-100 的正整数
            if (/^([1-9]|[1-9]\d|100)$/.test(this.data.guessNum)) {
                // 判断大小
                this.tellNum()
            } else {
                wx.showToast({
                    title: '请输入1-100的正整数',
                    icon: "none"
                })
            }
        }
    },
    // 判断猜的数偏大或偏小
    tellNum() {
        // 大了
        if (this.data.guessNum > this.data.randomNum) {
            wx.showToast({
                title: '换个小点的数试试',
                icon: "none",
                duration: 2000
            })
            this.count()
        }
        // 小了
        else if (this.data.guessNum < this.data.randomNum) {
            wx.showToast({
                title: '换个大点的数试试',
                icon: "none",
                duration: 2000
            })
            this.count()
        }
        // 猜对了
        else {
            // 弹窗提示答对啦
            this.setData({
                correct: true,
            })
            this.count()
            // 写入记录
            this.recordToStorage()
```

```
            this.resetNoHint()
        }
    },
    // 写入记录
    recordToStorage() {
        // 最近一次的记录
        var currentRecord = {
            randomNum: this.data.randomNum,
            times: this.data.times,
            time: util.formatTime(new Date())
        }
        // 获取已有的记录
        var record = Array.from(wx.getStorageSync('record'))
        // 把最近一次的记录加进去
        record.push(currentRecord)
        // 写入存储
        wx.setStorageSync('record', record)
    },
    // 重置
    reset() {
        let that = this
        wx.showModal({
            title: "提示",
            content: "重置 随机数 并 清空输入 和 次数",
            success: function (res) {
                if (res.confirm) {
                    that.setData({
                        randomNum: "",
                        guessNum: "",
                        times: 0
                    })
                } else {}
            }
        })
    },
    // 计数一次
    count() {
        this.setData({
            times: this.data.times += 1,
        })
    },
    //
    resetNoHint() {
        this.setData({
            randomNum: "", // 随机数
            guessNum: "", // 猜的数
            times: 0 // 猜的次数
        })
    },
    // 关闭回答正确后的提示框
    closeHint() {
        this.setData({
            correct: false
        })
    }
```

点击确定后调用 sure() 方法，没点击生成随机数或没有输入都有提示，利用正则表达式 /^([1-9]|[1-9]\d|100)$/.test(this.data.guessNum) 匹配 1-100 的正整数，输入其他的都有提示，按规则输入后进行判断大小，调用 tellNum() 方法，猜对后利用弹框提示 "答对啦" (用的自己写的提示框),然后调用 count() 计数一次，调用 recordToStorage() 方法写入记录：利用 json 新建一次记录 currentRecord={} ,把 randomNum , times , time 写在里面,调用 api ： wx.getStorageSync("record") ，把已有的记录保存到 record 数组里，利用 push() 方法把 currentRecord 添加进去，写入存储 wx.setStorageSync('record', record)
点击重置按钮调用 reset() 方法，弹框提示，确认后利用 setData() 把记录里的每一项清零

4.功能测试


WeChat
21:03
100%
猜数游戏
欢迎来到猜数字小游戏
点击生成随机数
提示
重置 随机数 并 清空输入 和 次数
取消
确定
重置
首页
记录


首页
记录