**创新实践课程报告**

专 业 计算机科学与技术

班 级 17052312

学 号 17051225

姓 名 王坤

杭州电子科技大学计算机学院

2019年 6月

“云上木坊”小程序介绍文档

**一、小程序简介**

“云上木坊”是一个介绍木工教程和提供线下制作功能的小程序，开发此小程序是为了传播木工文化，让木工爱好者通过学习交流、亲手制作的过程对木匠工艺有更深刻的了解和认识。

二、**应用场景**

1. **为木工爱好者提供一个学习交流的平台**

随着科技发展，再加上传统手工业者们信息交流不畅，传统文化的流失日益严重。借助互联网和本小程序，木工爱好者们可以学习木工教程、分享自己的见解或上传自己的教程。此外，由于环境、工具等多方面因素的限制，他们可能无法制作喜欢的木具。本小程序的线下体验功能恰恰提供了相应的渠道。

2. **为上班族提供一种解压方式**

业余生活单调、压力较大的上班族可以通过小程序的线下体验服务，亲手制作出自己喜欢的木具出来，充实生活、释放压力的同时还能获得满足感和成就感。

3. **艺术类高校的辅助教学工具**

充实的小程序教程未来可以成为高校相关专业的辅助教学工具，降低木工类课程的学习门槛。不仅能够减轻学生的学习压力，还能让更多人接触到木工制作。

**三、问题和解决方案**

实际问题：人们对于传统木匠工艺缺乏了解；木工爱好者缺少分享平台

解决方案：通过线上的视频和图文教程，用户可以学习木工作品的制作流程。闲暇时，用户可以通过预约，在线下实体亲手制作自己的作品。通过以上步骤，用户可以对木工活有更深刻的认识。

木工爱好者可以在线上教程评论区发表自己的见解或分享自己制作特色木工作品的教程供他人学习和参考。

**四、功能设计**

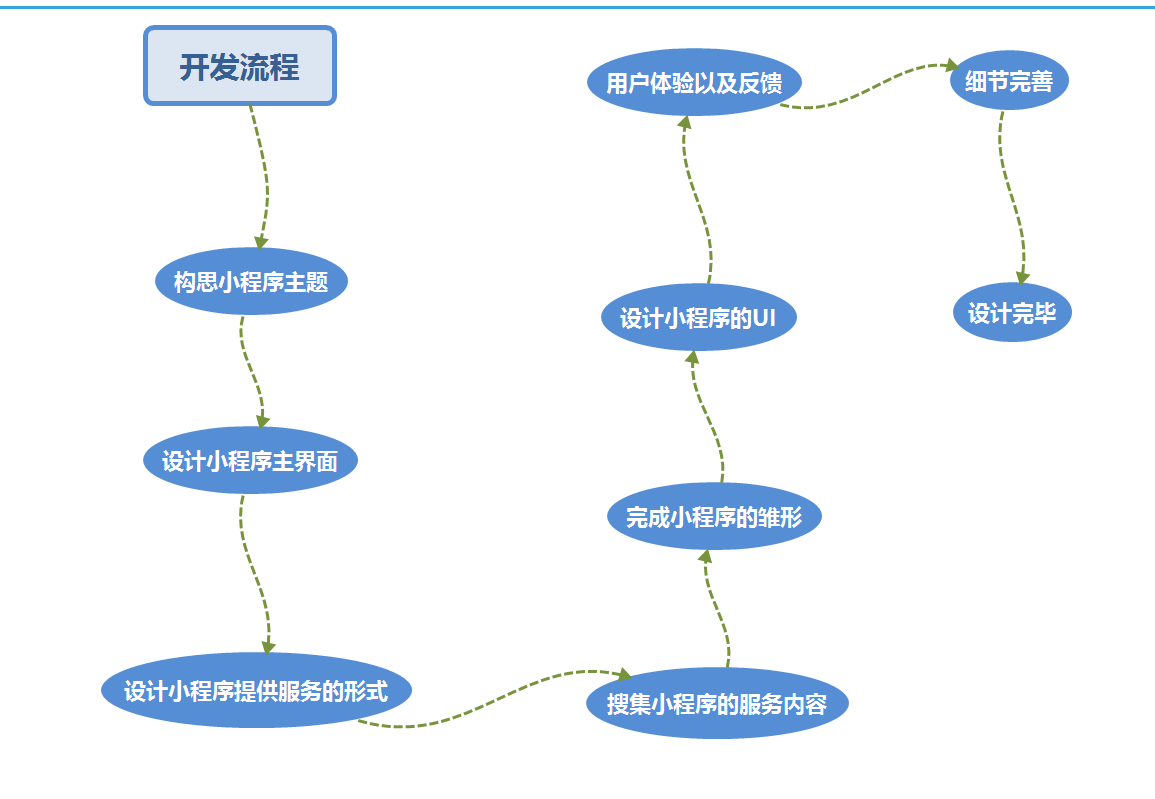
小程序整体以淡蓝色为主调，包括线上教学和线下预约体验功能。线上教学我们采用了图文、视频相结合的设计，同时辅以一个3D可操作结构图供用户更加直观地了解木工作品的结构。教程下方是评论区，供用户分享和交流。线下预约的页面，用户可以预约体验、查看预约记录或取消预约。“我的”页面中，提供了木工工具图鉴及相应的功能介绍供用户了解。

**五、技术方案**

**技术架构简介:**

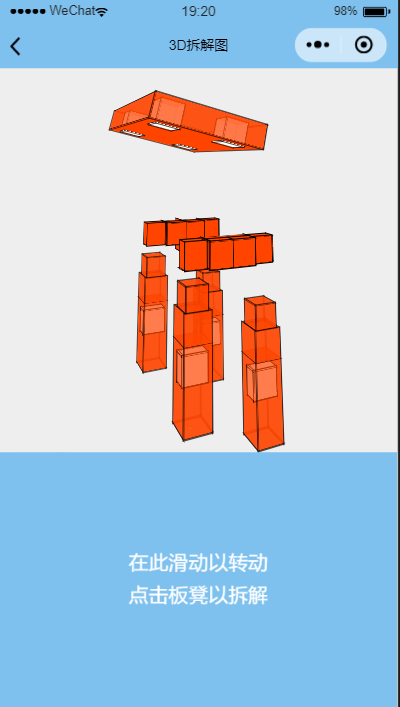
1. 前端使用的全是原生的语法,后台使用的是云开发,前端使用了部分插件包括腾讯地图,腾讯视频等,API主要是用在获取用户openid,即直接使用云函数调用getuserInfo,还有数据的增删改查,db.collection. 这一个接口对数据库进行增删改查,然后,还使用了wx.getsetting和wx.getuestInfo去检测用户的授权和获取用户数据
2. 小程序使用的是官方的开发工具<**微信开发者工具**>

具体开发流程如下:



**技术难点详细介绍**：

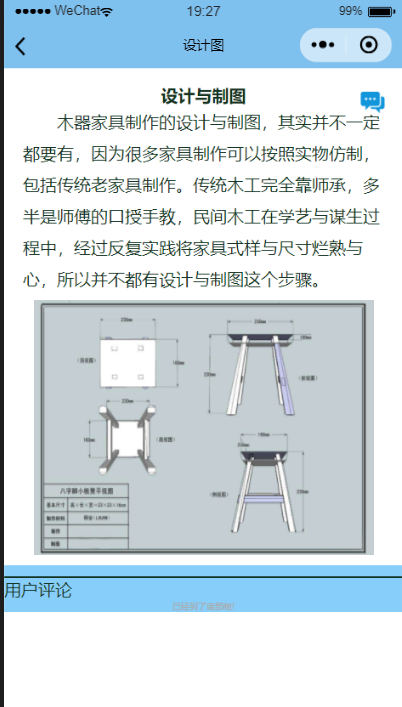
1. 板凳3D结构图



这个看起来很简单,但是做起来是非常的需要耐心的,因此这个使用了大量的css3因此,对于低端机的适配是一个很大的问题,也没法解决,这个3D图的制作的主要问题就在于各个组件的大小和位置的设计,设计之初并没有画详细的设计图纸,而是把一个一个的组件制作出来,一个一个的去调位置和适当的大小,最后还是先设计好每个组件的大小,然后再一步到位,把各个组件拼凑起来就成了!

**2)** 如何解决发表评论后上一个页面可以刷新并且返回不会返回到发表评论页面

页面如下



**页面一**  **页面二**

就是这两个页面之间的跳转问题,由于第一个页面是使用模板文件要根据上一个页面的点击来传递参数,然后去查询对应的数据库,这里我是这么解决的:首先使用navgateback关闭这个页面栈,然后,在第一次加载第一个图片所示的页面的时候,使用全局变量globaldata,在这里设计一个index,然后,在onload函数里面,写一个if语句,如果glabaldata.index存在,则使用glabaldata.index去查询对应的数据库,如果不存在,则使用上一个页面传递过来的index去查询数据库就好了.

运用同样的方法解决问题的还有就是’工具图鉴’这个页面了,也是第一次加载的时候,在globaldata里面设置一个mapinfo去保存数据库查询结果,然后,当我们第二次进入这个页面的时候,可以使用globaldata里面的结果,可以避免用户的等待,而不用第一次加载那么久了.

这里还有 一个引申的问题,就是发表评论之后,顺着页面栈返回到图文教程的页面,刷新评论?这里我是这样解决的:把数据库查询操作放在onshow里面,而不是放在onload里面,这面,当我们顺着页面栈返回到这个页面的时候,首先会展示之前的页面,不会直接刷新,然后,已渲染过的东西,在渲染的时候,肉眼是察觉不到的,然后,数据库查询完毕后,就可以直接刷新发表的评论了!

**3)**  关于云开发的相关增删改查

云开发的数据库和我们自己搭建的数据库大部分是一样的,但是也有一个 地方不一样,就是单条记录的查询,传统的数据库可以使用数据记录中的任何 一条属性去查询,但是云开发不可以,云开发在进行单条记录查询的时候,只能够使用该记录的’\_id’去查询结果,这一点设计非常的不人性化

**4)** 用户上传图片的时候,如何保证图片命名的唯一性

关于这个问题,我试过很多办法,比如使用随机数,使用时间,但是,这样并不能完全的避免重复命名的出现,因此,我是用了这个办法,代码如下:

S4: function () {

return (((1 + Math.random()) \* 0x10000) | 0).toString(16).substring(1);

},

guid: function () {

return (this.S4() + this.S4() + "-" + this.S4() + "-" + this.S4() + "-" + this.S4() + "-" + this.S4() + this.S4() + this.S4());

},

虽然使用的是随机数,但是,并不是使用一个随机数,而是使用了8个随机数,从概率学上来讲,这样的做出来的名字,重复的概率趋近于0,几乎是不存在的,因此是可以这样是合计的

**5) 用户openid的获取**

**由于小程序存在社交功能,因此需要获取**用户的openid,但是,对于openid的获取,第一种是在用户第一次对小程序授权的时候能够拿到openid,但是,这显然是不合适的,不可能让用户每次使用都需要授权,而应该是授权一次就可以了,因此,这样后续无法获取到openid,然后我去看网上的教程([参考链接](https://blog.csdn.net/qq_27626333/article/details/54614037)),第一种我上面讲过了,临时会话可以获取,第二种是使用code 换取 session\_key

​这是一个 HTTPS 接口，开发者服务器使用登录凭证 code 获取 session\_key 和 openid。其中 session\_key 是对用户数据进行加密签名的密钥。为了自身应用安全，session\_key 不应该在网络上传输。具体点的代码实例入下:



这种方法虽然可行,但是我后太并不是直接架构的,而是使用了云开发,这里云开发提供了一种更好更安全的方式,使用云函数从用户记录里面直接拿到,这种方式比上面这种方式更加的简单,安全,具体的代码如下:

**云函数部分**

// 云函数入口文件

const cloud = require('wx-server-sdk')

cloud.init()

// 云函数入口函数

exports.main = async (event, context) => {

return event.userInfo;

}

**小程序端调用代码如下:**

// 获取用户openid

getOpenid() {

let that = this;

wx.cloud.callFunction({

name: 'getOpenid',

complete: res => {

var openid = res.result.openId;

app.globalData.openId = openid;

that.setData({

})

console.log(app.globalData.openId)

}

})

},

这就是云开发的好处所在,不论是速度,性能,还是代码空间复杂度,都更低

**六、团队组成与分工**

王坤：负责程序开发、UI设计以及产品设计。

谢经纬：负责素材收集、视频剪辑和部分页面开发。

**七、创意来源**

由于本人出生与农村地区，祖父是一个木匠，当时村里的各类四角桌，板凳，竹椅，还有各类用于农务的农具像风车、团筛、炭筛、打谷机**、**木制犁、木制水车等等之类的都是祖父制作的，对于各类木具的制作和使用都是非常熟悉的，参赛之初是想开发一款关于传统木匠的小游戏，然后，经过各种方案的设想以及和组员、指导老师的讨论，最后还是制作成了小程序，但是由于教学素材的缺乏，只能勉强的制作出一个板凳的教程，家乡现在也没有一个木匠，没有人传承这门手艺，也没有人愿意传承这门手艺，木具的制作教程收集几乎于不可能，所以我希望通过让人们看到这款小程序，让有能力的人能参与到教程的制作中，让有兴趣的人能加入到木具的学习中来。

**八、小程序用户活动框图**



**九、开发心得**

其实这个小程序做到最后，给我的感觉就是，没有多大的技术含量，这里的技术不是广义的技术，就是狭义的技术的本身，一切和小程序开发有关的大小，都可以查询文档轻易的使用，而且，使用小程序的云开发，不用自己搭建后台，非常的方便，如果真的让我说哪个地方最难的话，应该就是那个3D的交互式板凳了，确实是花了不少的时间，而且在小程序的制作之初，我还刷了2套完整的小程序教学视频。然后，每周去向老师汇报，都是各种问题，这期间我甚至想另起炉灶，但是又不想这么放弃，还是弄了下来，而且教程的素材非常的难于寻找，这个是这个小程序制作最大的难点。在这个小程序制作完后，虽然我觉得的最终版本已经很不错，虽然还是被老师吐槽，但是我没有改了，我觉得，小程序，本身就是实例化的软件，和小游戏不一样，小游戏只要具有一定的趣味性，就会有人玩，即小程序本身导致了需求。但是小程序本身不具有趣味性，它的存在是给有需求的人，即需求吸引用户，而不是小程序让用户产生需求的，所以，对于一般大众而言这款小程序，毫无亮点，晦涩难用。但是这款小程序的交互性还是不够好，还是需要多多的学习和多多观察平时自己用的小程序的设计理念和模式才行！

**十、自评分**

|  |  |  |  |
| --- | --- | --- | --- |
| **考查点** | **分值** | **备注** | **得分** |
| 基本CRUD | 20 |  | 18 |
| 后端数据存取(云开发或自编后端均可) | 10 |  | 9 |
| UI设计与样式实现 | 10 |  | 8 |
| 导航(全局导航栏、tabbar、分类、分页等) | 10 |  | 9 |
| 手势操作或界面特效 | 10 |  | 8 |
| 登录与会话 | 10 |  | 9 |
| 多媒体相关 | 10 |  | 8 |
| 代码模块化或模板化 | 10 |  | 8 |
| 代码编写规范及其它 | 10 |  | 9 |

**总评：86**