

江西科技师范大学

毕业设计

题目（中文）： 基于 Web 客户端技术的个性化 UI 的设计和编程

（外文）： Customized UI design and Programming
based on Web client technology

院（系）： 元宇宙产业学院

专 业： 计算机科学与技术

学生姓名： 余振宇

学 号： 20213638

指导教师： 李健宏

目录

基于 Web 客户端技术的个性化 UI 的设计和编程	1
（Customized UI design and Programming based on Web client technology）	1
1. 前言	1
1.1 毕设任务分析	1
1.2 研学计划	2
1.3 研究方法	2
2. 技术总结和文献综述.....	3
2.1 Web 平台和客户端技术概述	3
2.2 项目的增量式迭代开发模式	5 错误!未定义书签。
3. 内容设计概要.....	6
3.1 分析和设计	6
3.2 项目的实现和编程	6
3.3 项目的运行和测试	7
3.4 项目的代码提交和版本管理	7
4. 移动互联时代的 UI 开发初步——窄屏终端的响应式设计	8
5. 应用响应式设计技术开发可适配窄屏和宽屏的 UI	8
6. 个性化 UI 设计中对鼠标交互的设计开发	错误!未定义书签。
7. 对触屏和鼠标的通用交互操作的设计开发.....	错误!未定义书签。
8. UI 的个性化键盘交互控制的设计开发	15
9. 谈谈本项目中的高质量代码.....	17
10. 用 gitBash 工具管理项目的代码仓库和 http 服务器.....	18
10.1 经典 Bash 工具介绍	18
10.2 通过 gitHub 平台实现本项目的全球域名	18
10.3 创建一个空的远程代码仓库	19
10.4 设置本地仓库和远程代码仓库的链接	19
参考文献:	23
写作指导:	错误!未定义书签。

基于 Web 客户端技术的个性化 UI 的设计和编程

（Customized UI design and Programming based on Web client technology）

科师大元宇宙产业学院 2021 级 余振宇

摘要：近十年来，html5 为核心的 web 标准的软件开发技术、Web 技术以其跨操作系统平台的优势成为了广泛流行的软件开发手段。本项目选择 html5 的 web 客户端技术为技术路线，展开对程序设计和软件开发的研究和实践。通过广泛查阅相关技术书籍、开发者论坛和文献，设计开发了一个个性化的用户界面（UI）的应用程序。在开发中综合应用了 html、css、javascript 等语言。本项目也采用了响应式设计编程，可以智能地适应移动互联网时代用户屏幕多样化的需要，以该响应式技术为支撑做到了最佳适配用户屏幕。在功能上以 DOM 技术和事件驱动模式的程序为支撑实现了对鼠标、触屏、键盘的底层事件响应和流畅支持，为鼠标和触屏设计了一个对象模型，用代码实现了对这类指向性设备的模拟。为了处理好设计和开发的关系，项目用了工程思想管理，使用了软件工程的增量式开发模式，共做了 6 次项目迭代开发，每次迭代都经历了开发 4 个经典开发阶段（A:Analysis, D:Design, I: Implementation, T:Testing）本项目还使用了 git 工具进行代码和开发过程日志记录，一共做了 12 次提交代码的操作，详细记录和展现了开发思路和代码优化的过程，最后通过 gitbash 把项目上传到 github 上，建立了自己的代码仓库，并将该代码仓库设置成为了 http 服务器，实现了本 UI 应用的全球便捷访问。

1. 前言

毕业设计是一个需要综合运用本科阶段学习的计算机科学与技术知识，尤其是程序设计和软件工程领域的知识结合自己的代码能力，找到感兴趣的技术路线，寻找合适的设计方向然后有条理地系统落实分析问题、建立模型、软件设计、系统实施、测试调试的等传统软件工程的全部的流程。践行毕业设计，总结开发文档撰写论文。

1.1 项目概述

本次课程的设计目标是实现学生独立自主完成一个网页界面的搭建，其内容包括但不限于网页界面的设计、异步代码的设计应用、在时间维度上对代码运行优化、对鼠标模型的捕捉响应、对键盘的个性化设计并优化，这些内容基本上涵盖了一个网站大部分的内容，能够对学生能力起到一个很好的锻炼效果。

1.2 研学计划

研学计划的理论包括面向对象的程序设计语言、数据结构和算法、操作系统、软件工程等。这些核心课程供理论指导非常抽象，加之基本上以理论知识为主。其设计的内涵就是大学理论的学习在实践层面做一次综合演练和总结，期间也需要配合涉及的有关核心课程的理论包括面向对象的程序设计语言、数据结构和算法、操作系统、软件工程等。若能在实践层面应用这些核心课程的关键知识则必然会在理解和技术二个维度提升自己的专业性。

1.3 研究方法

研究方法分二个阶段，首先选择一条自己感兴趣的技术实践路线，把核心的技术加以整合学习，以导师的案例项目为参考，主要是理解好各个技术之间的关系，在项目中的作用和分工，更重要的是在项目实施中提升自己的写高质量的代码能力。

当仿造导师的案例的技术基本实现后，则可以视为实践和理论基本打通，此时就可进入第二个阶段，开始真正做自己的毕设软件。第二阶段一般按软件工程的标准来规范开发:1、结合自己的问题做出定义和分析;2、设计一套合适的技术解决方案;3、按解决方案设计流程和编写相关代码，实现技术部署;4、调试代码、测试软件、性能调优。其中第 3、4 步可以发现前面步骤的问题，因此可能会在第 2，3，4 步多次循环，发现和解决第 2 步的设计失误或第 3 步的代码错误。

2. 技术总结和文献综述

2.1 Web 平台和客户端技术概述

Web 之父 Tim Berners-Lee 在发明 Web 的基本技术架构以后,就成立了 W3C 组织,该组织在 2010 年后推出的 HTML5 国际标准,结合欧洲 ECMA 组织维护的 ECMAScript 国际标准,几乎完美缔造了全球开发者实现开发平台统一的理想,直到今天,科学家与 Web 行业也还一直在致力于完善这个伟大而光荣的理想。学习 Web 标准和 Web 技术,学习编写 Web 程序和应用有关工具,最终架构一套高质量代码的跨平台运行的应用,是我的课程设计应用的技术路线。Web 应用的程序设计体系由三大语言有机组成: HTML, CSS, JavaScript。这三大语言的组合也体现了人类社会化大生产分工的智慧,可以看作三套相对独立体系实现了对一个信息系统的描述和控制,可以总结为: HTML 用来描述结构(Structure)、CSS 用来描述外表(presentation)、Javascript 用来描述行为(Behavior);这也可以用经典的 MVC 设计模式来理解 Web 平台架构的三大基石, Model 可以理解为 HTML 标记语言建模, View 可以理解为用 CSS 语言来实现外观, Controller 则可理解为用 JavaScript 结合前面二个层次,实现了在微观和功能层面的代码控制。

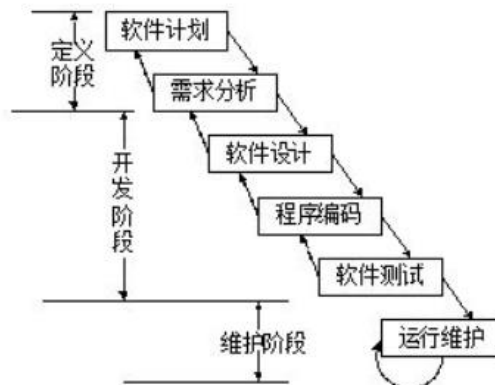
2.2.MVC 设计模式

MVC 全名为 Model·ViewController,即模型-视图-控制三层体系架构。它是一种设计模式,同时也反映了人类社会生产中的伟大分工智慧。它使用一种业务逻辑,数据与界面显示分离的方法来组织代码。在 Web 中,我们可以这样理解:M 对应的是 Model(模型层),可采用 HTML 标记语言进行建模;V 对应的是 View(视图层),单独用 CSS 语言来渲染外观;JavaScript 对应的是 Controller(控制层),它结合前二者,实现功能层面和微观代码的所有控制。这样将对象、显示、控制分离有助于提高代码的灵活性与复用性。

2.3 软件开发过程管理

2.3.1 瀑布模型

瀑布模型是将软件生存周期的各项活动规定为按固定顺序而连接的若干阶段工作，形如瀑布流水。1970 年温斯顿·罗伊斯（Winston Royce）提出了著名的“瀑布模型”，直到 80 年代早期，它一直是被广泛采用的软件开发模型。瀑布模型核心思想是按工序将问题化简，将功能的实现与设计分开，便于分工协作，即采用结构化的分析与设计方法将逻辑实现与物理实现分开。将软件生命周期划分为制定计划、需求分析、软件设计、程序编写、软件测试和运行维护等六个基本活动，并且规定了它们自上而下、相互衔接



的固定次序，如同瀑布流水，逐级下落

瀑布模型有以下优点

- (1) 为项目提供了按阶段划分的检查点。
- (2) 当前一阶段完成后，您只需要去关注后续阶段。
- (3) 可在迭代模型中应用瀑布模型。

增量迭代应用于瀑布模型。迭代 1 解决最大的问题。每次迭代产生一个可运行的版本,同时增加更多的功能。每次迭代必须经过质量和集成测试。

(4) 它提供了一个模板，这个模板使得分析、设计、编码、测试和支持的方法可以在该模板下有一个共同的指导。

同时瀑布模型有以下缺点

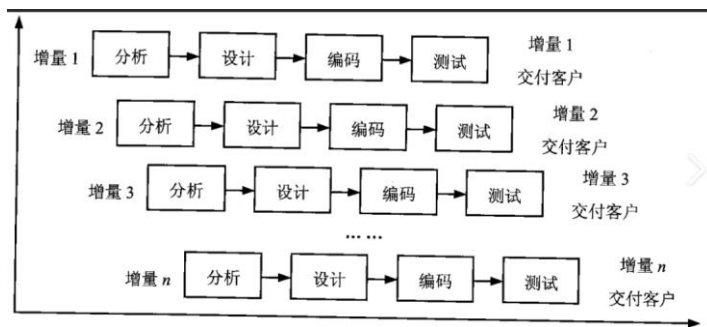
- (1) 各个阶段的划分完全固定，阶段之间产生大量的文档，极大地增加了工作量。
- (2) 由于开发模型是线性的，用户只有等到整个过程的末期才能见到开发成果，从而增加了开发风险。

(3) 通过过多的强制完成日期和里程碑来跟踪各个项目阶段。

(4) 瀑布模型的突出缺点是不适应用户需求的变化。

2.3.2 增量模型

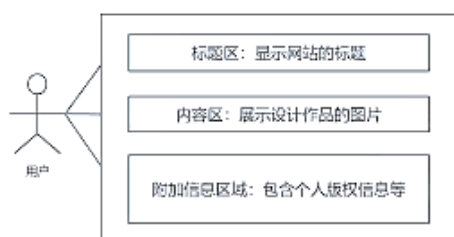
增量模型是把待开发的软件系统模块化，将每个模块作为一个增量组件，从而分批次地分析、设计、编码和测试这些增量组件。运用增量模型的软件开发过程是递增式的过程。相对于瀑布模型而言，采用增量模型进行开发，开发人员不需要一次性地把整个软件产品提交给用户，而是可以分批次进行提交



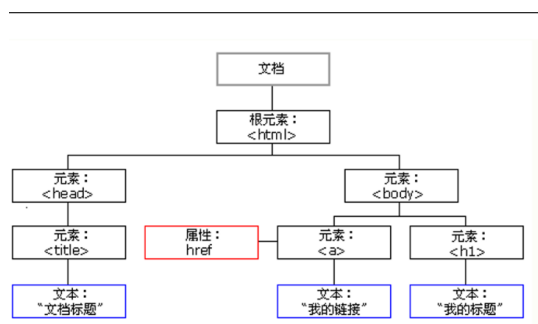
3. 内容设计概要

3.1 分析和设计

为了在互联网上更加方便且直观地展示我们所收集到的一些艺术作品的相关信息(包括图片以及作者,创作日期等内容),我计划搭建一个开源的个性化展示网站。同时,为了增加网站的可读性和美观性,我们首先给这些信息进行了分类,我将页面划分为三个主要部分包括标题区、内容区、附加信息区等,如图·4.1 所示。其中,Dom 结构树如图·4.2 所示。。



图·4.1



图·4.2

3.2 项目的实现和编程

使用 HTML5 创建了页面的结构模型,然后再分别创建子元素 header、main.footer 用于分别存储标题,作品内容以及个人版权等信息。接着用 CSS 对页面进行渲染,以便呈现出如图·4.3 所示的页面效果。其中,实现方式如代码块 4.1 所示

```
1 <header>
2 |   Design·Works·Display
3 </header>
4 <main>
5 |   Poster.Content
6 </main>
7 <footer>
8 |   copyright·2024余振宇3092666104@qq.com
9 </footer>
```

图 4.1

3.3 项目的运行和测试

《 我的毕设题目 》
我的主题内容: ‘读好书、练思维、勤编程’ @masterLijh 计算思维系列课程
CopyRight 余振宇 江西科技师范大学 2024-2025

3.4 项目的代码提交和版本管理

本项目的文件通过 gitBash 工具管理,作为项目的第一次迭代,在代码提交和版本管理环节,我们的目标是建立项目的基本文件结构,还有设置好代码仓库的基本信息:

如开发者的名字和电子邮件。

(1)项目代码的上传,

在编写并成功运行了 `index.html` 和 `myCss.css` 以及 `myJsjs` 的代码后,执行以下操作(如命令·4.2 所示)对第一次的项目代码进行提交。项目代码提交成功之后, `gitBash` 的反馈信息如图·4.4 所示。

```
DELL@LSP MINGW64 /d/abc (master)
$ git commit -m 项目第一版:“三段论”式的内容设计概要开发【master (root-commit) 2de024】项目第一版:“三段论”式的内容设计概要开发
[master 26ff41c] 项目第一版:“三段论”式的内容设计概要开发【master (root-commit) 32de024】项目第一版:“三段论”式的内容设计概要开发
9 files changed, 1261 insertions(+), 127 deletions(-)
create mode 100644 exp/1.3.html
```

图·4.4

(2) 项目日志查询

自此,项目代码仓库也就开启了严谨的历史记录,我们可以使用日志命令(`gitlog`)查询如图 4.5 所示

```
commit ae24c000a31475254990d57de668b32ed0605ba0
Author: Yu <3092666104@qq.com>
Date: Tue Apr 23 10:27:25 2024 +0800

“第一次提交,我们完成了软件的设计概要,完成了三个部分。header部分放了软件的标题,main部分放了软件的内容,footer放了软件的动态反馈。
```

图 4.5

4. 移动互联时代的 UI 开发初步——窄屏终端的响应式设计

由于成本和技术等因素,不同的设备(如电脑,手机等)拥有各种不同的屏幕尺寸和分辨率。为了应对这种差异,设计师们选择制定一般的布局准则,让浏览器自行解决如何在给定的计算机上显示页面,而不是为了每个设备类型提供专门的网页版本¹¹。这种方式有一个有趣的结果,那就是同一个网页在不同浏览器或硬件上可能会有略微不同的显示效果。例如,如果一个屏幕比另一个屏幕宽,则可以显示的文本行长度或图像大小会有所不同。这是因为,网页提供了关于期待呈现的一般指导方针,具体的显示细节由浏览器根据设备特性自行决定。

4.1 分析与设计

人们获取信息的途径是多样且不断变化的，在一些固定场所比如家中或教室，我们更倾向于使用电脑来进行学习和工作。但是，在一些变化的场景如出行时，我们却更倾向于使用手机和平板等设备来进行信息的快速查看。那现在我们的网站仅仅在电脑端实现了较为舒适的布局，如果在手机端呢?它可能就错位了!因此，我们需要一个同时适用于电脑端和手机端的网站，以确保无论在哪一种设备上，用户都能获得良好的使用体验。设计如图·5.1 和图·5.2 所示。.

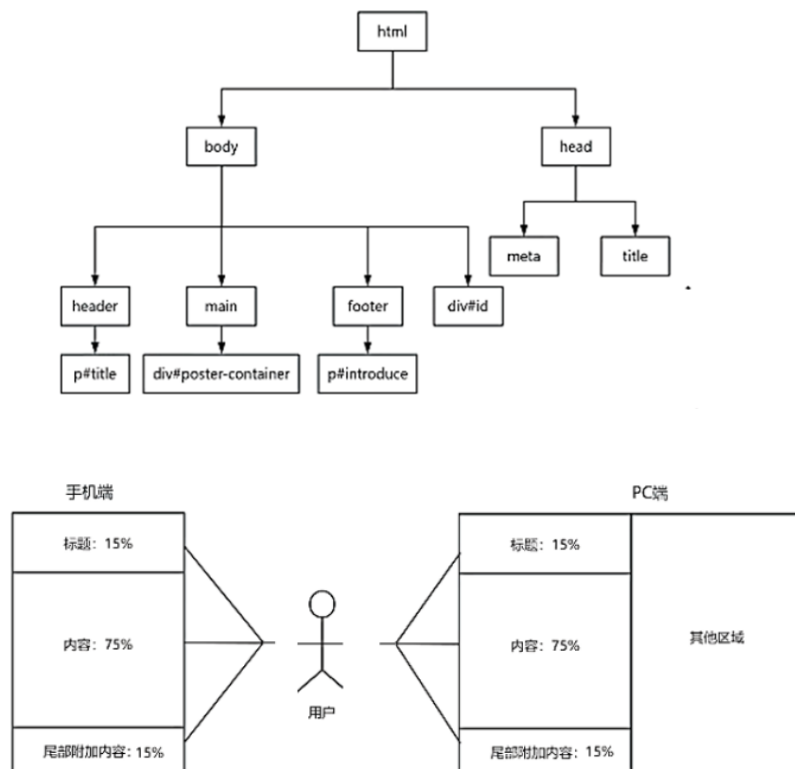


图 5.1 窄屏设计用例图

图 5.2 窄屏幕设计 DOM 树

4.2 编程与实现

为了让页面能够自动适应不同设备(例如 iPhone 12 Pro 和电脑端)的屏幕尺寸,使其呈现出良好的自适应效果,我采取了 JavaScript 和 CSS 来动态调整页面布局和样式。在宏观上,我利用 JavaScript 动态获取设备窗口的宽度和高度信息,以便于根据不同设备的特性,动态调整页面元素的大小。其次,我使用 CSS 对页面内容进行整体性的划分和样式的定义,来实现页面布局的自适应。

本阶段初次引入了 CSS 中比较高阶的语法,如 em 和%单位,这些单位在响应式设计中非常有效。通过利用%单位,我给 header, body, footer 分别赋予了相对于父容器的比例,如 15%, 75%, 10%。而通过 cm 单位,继承 body 的属性,灵活地设置了字体的大小。如代码块 5.1 所示。"

4.3 测试与运行

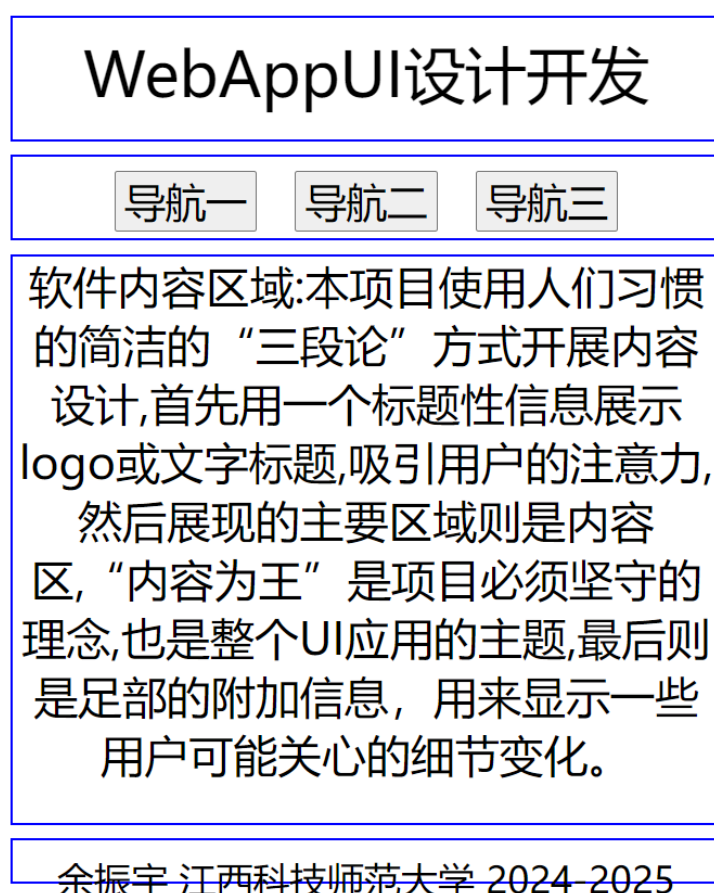


图 5.3 电脑页面显示图

4.4 代码提交与版本管理

```
<script>
  var UI = {};
  UI.appWidth = window.innerWidth > 600 ? 600 : window.innerWidth ;
  UI.appHeight = window.innerHeight;
  const LETTERS = 22 ;
  const baseFont = UI.appWidth / LETTERS;

  //通过更改body对象的字体大小，这个属性能够遗传其子子孙孙
  document.body.style.fontSize = baseFont + "px";
  //通过把body对象的宽度和高度设置为设备/屏幕的宽度和高度，实现全屏。
  //通过css对子对象百分比（纵向）的配合，从而实现响应式设计的目标。
  document.body.style.width = UI.appWidth - 2*baseFont + "px" ;
  document.body.style.height = UI.appHeight - 4*baseFont + "px";
```

图 5.4

5 个性化 UI 交互——鼠标模型

5.1 分析与实现

个性化 UI 交互，尤其是鼠标模型的设计与实现，是提升用户界面体验和互动性的重要环节。在各类应用程序和游戏中，具有良好交互性的鼠标模型不仅可以提升操作的直观性和便捷性，还能增强用户的整体使用体验。

5.2 编程与实现

编程与实现如图所示：

```

<script>
  var UI = {};
  UI.appwidth = window.innerWidth > 600 ? 600 : window.innerWidth ;
  UI.appHeight = window.innerHeight;
  const LETTERS = 22 ;
  const baseFont = UI.appwidth / LETTERS;

  //通过更改body对象的字体大小，这个属性能够遗传其子子孙孙
  document.body.style.fontSize = baseFont + "px";
  //通过把body对象的宽度和高度设置为设备/屏幕的宽度和高度，实现全屏。
  //通过CSS对子对象百分比（纵向）的配合，从而实现响应式设计的目标。
  document.body.style.width = UI.appwidth - 2*baseFont + "px" ;
  document.body.style.height = UI.appHeight - 4*baseFont + "px";

```

用例图如图 5.5 所示 dom 树如图 5.6 所示

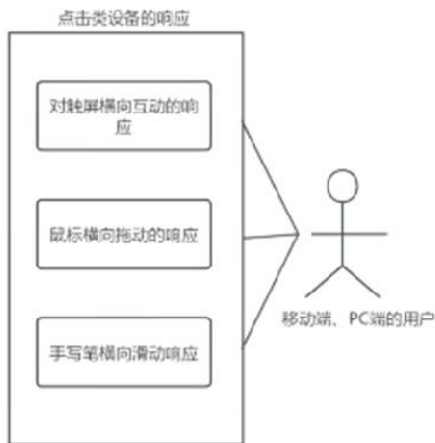


图 5.5

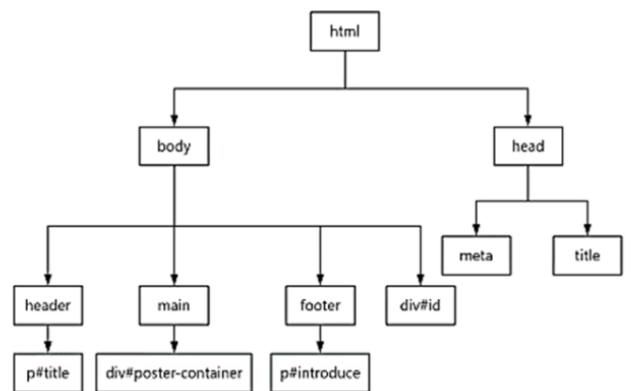


图 5.6

5.3 测试与运行

运行代码如图 6.1 所示

```

if(window.innerWidth < 900){
  $("aid").style.display='none';
}
$("aid").style.width=window.innerWidth - UI.appWidth - 2*baseFont + 'px';
$("aid").style.height= document.body.clientHeight + 'px';

//尝试对鼠标设计UI控制
var mouse={};
mouse.isDown= false;
mouse.x= 0;
mouse.deltaX=0;
$("bookface").addEventListener("mousedown",function(ev){
  let x= ev.pageX;
  let y= ev.pageY;

```



图 6.1

6.探索 UI 对于模拟触屏的应用

6.1 分析与设计

在当今的移动设备和触摸屏技术广泛应用的背景下，探索 UI 设计在模拟触屏方面的应用成为了一个重要的研究领域。这不仅涉及到如何通过 UI 设计改善用户与设备的交互方式，还包括如何利用现有的技术和工具来模拟和优化触屏操作，从而为用户提供更加直观、高效和愉悦的使用体验。

6.2 编程与实现

运行代码如图 6.3 所示

```

...
$("bookface").addEventListener("mouseout",function(ev){
    ev.preventDefault();
    mouse.isDown=false;

    $("bookface").textContent= "鼠标松开!";
    if(Math.abs(mouse.deltaX) > 100){
        |  $("bookface").textContent += " 这次是有效拖动! " ;
    }else{
        |  $("bookface").textContent += " 本次算无效拖动! " ;
        |  $("bookface").style.left = '7%' ;
    }
});

$("bookface").addEventListener("mousemove",function(ev){
    ev.preventDefault();
    if (mouse.isDown){
        console.log("mouse isDown and moving");
        mouse.deltaX = parseInt( ev.pageX - mouse.x );
        |  $("bookface").textContent= "正在拖动鼠标, 距离: " + mouse.deltaX +"px 。 ";
        |  $("bookface").style.left = mouse.deltaX + "%";
    }
});

```

图 6.3

实现代码如图 6.4 所示



7 个性化 UI 对键盘的控制

7.1 分析与设计

个性化 UI 对键盘的控制是实现用户与设备交互的关键技术之一，它允许开发者根据应用的需求和用户的偏好定制键盘的行为和外观

7.2 编程与实现

运行代码如图所示


```

if(ev.touches){console.log("touches1"+ev.touches);
Pointer.x = ev.touches[0].pageX ;
    Pointer.y = ev.touches[0].pageY ;
console.log("Touch begin : "+"("+Pointer.x +"," +Pointer.y +")" ) ;
    $("bookface").textContent= "触屏事件开始, 坐标: "+"("+Pointer.x+"," +Pointer.y+")";
}else{
    Pointer.x= ev.pageX;
    Pointer.y= ev.pageY;
console.log("PointerDown at x: "+"("+Pointer.x +"," +Pointer.y +")" ) ;
    $("bookface").textContent= "鼠标按下, 坐标: "+"("+Pointer.x+"," +Pointer.y+")";
}
};
let handleEnd = function(ev){
    Pointer.isDown=false;
    ev.preventDefault()
    //console.log(ev.touches)
    if(ev.touches){

```

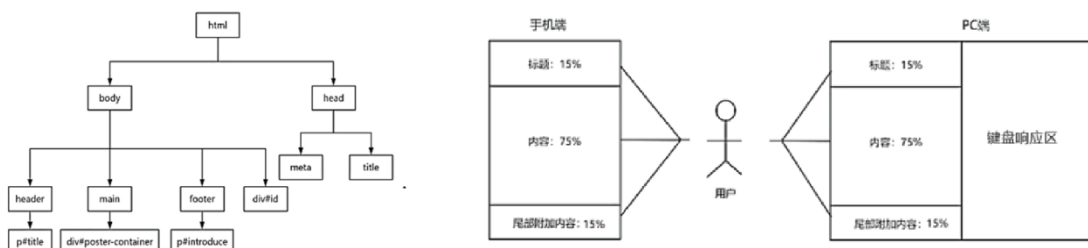
实现界面如图所示



8.UI 的个性化键盘交互控制的设计开发

8.1 分析与设计

在数字化的世界里，我们的输入方式越来越多样化，键盘输入也是一种不可忽略的方式，若是能和鼠标协同操作，能够大大增加我们的工作效率，设计结构如图所示



8.2 项目的实现与编程

使用 HTML5 创建了页面的结构模型,然后再分别创建子元素 header、main.footer 用于分别存储标题,作品内容以及个人版权等信息。接着用 CSS 对页面进行渲染,以便呈现出如图所示的页面效果。其中,实现方式如代码块所示

```
},
let handleMoving = function(ev){
  ev.preventDefault();
  if (ev.touches){
    if (Pointer.isDown){
      console.log("Touch is moving");
      Pointer.deltaX = parseInt( ev.touches[0].pageX - Pointer.x );
      $("bookface").textContent= "正在滑动触屏, 滑动距离: " + Pointer.deltaX + "px 。 ";
      $('bookface').style.left = Pointer.deltaX + 'px' ;
    }
  }else{
    if (Pointer.isDown){
      console.log("Pointer isDown and moving");
      Pointer.deltaX = parseInt( ev.pageX - Pointer.x );
      $("bookface").textContent= "正在拖动鼠标, 距离: " + Pointer.deltaX + "px 。 ";
      $('bookface').style.left = Pointer.deltaX + 'px' ;
    }
  }
}
```

8.3 项目的运行和测试

实现界面如图所示



8.4 项目的代码提交和版本管理

本项目的文件通过 gitBash 工具管理,作为项目的第一次迭代,在代码提交和版本管理

环节，我们的目标是建立项目的基本文件结构，还有设置好代码仓库的基本信息：如开发者的名字和电子邮件

(1) 项目代码的上传

```
DELL@LSP MINGW64 /d/abc (master)
$ git commit -m "在数字化的世界里，我们的输入方式越来越多样化，键盘输入也是一种不可忽略的方式，若是能和鼠标协同操作，能够大大增加我们的工作效率,为此我设计了一个能够同时响应鼠标和键盘的网页，以方便用户进行操作使用。"
[master c03ed8d] 在数字化的世界里，我们的输入方式越来越多样化，键盘输入也是一种不可忽略的方式，若是能和鼠标协同操作，能够大大增加我们的工作效率,为此我设计了一个能够同时响应鼠标和键盘的网页，以方便用户进行操作使用。
2 files changed, 61 insertions(+), 273 deletions(-)
```

(2) 项目日志查询

自此，项目代码仓库也就开启了严谨的历史记录，我们可以使用日志命令(gitlog)查询如图所示

```
commit c03ed8d0ead83a0415dc7995c0441cc7e3f69eff (HEAD -> master)
Author: Yu <3092666104@qq.com>
Date:   Wed Jun 19 00:36:17 2024 +0800

    在数字化的世界里，我们的输入方式越来越多样化，键盘输入也是一种不可忽略的方式，若是能和鼠标协同操作，能够大大增加我们的工作效率,为此我设计了一个能够同时响应鼠标和键盘的网页，以方便用户进行操作使用。
```

9. 谈谈本项目中的高质量代码

创建一个 Pointer 对象，践行 MVC 设计模式，设计一套代码同时对鼠标和触屏实现控制。

如下图所示，我们设计了一个功能（function）来监听事件的发生并作出回应：当触摸也就是鼠标拖动或者是在手机端进行手指滑动时，后台就会监听到事件，并且反馈相应的内容。这是我认为的项目中设计相当之好的一片代码。

```

let handleMoving = function(ev){
    ev.preventDefault();
    if (ev.touches){
        if (Pointer.isDown){
            console.log("Touch is moving");
            Pointer.deltaX = parseInt( ev.touches[0].pageX - Pointer.x );
            $("bookface").textContent= "正在滑动触屏，滑动距离: " + Pointer.deltaX + "px 。 ";
            $('bookface').style.left = Pointer.deltaX + 'px' ;
        }
    }else{
        if (Pointer.isDown){
            console.log("Pointer isDown and moving");
            Pointer.deltaX = parseInt( ev.pageX - Pointer.x );
            $("bookface").textContent= "正在拖动鼠标，距离: " + Pointer.deltaX + "px 。 ";
            $('bookface').style.left = Pointer.deltaX + 'px' ;
        }
    }
}

```

10 用 gitBash 工具管理项目的代码仓库和 http 服务器

10.1 经典 Bash 工具介绍

When we speak of the command line, we are really referring to the shell. The shell is a program that takes keyboard commands and passes them to the operating system to carry out. Almost all Linux distributions supply a shell program from the GNU Project called bash. The name is an acronym for bourne-again shell, a reference to the fact that bash is an enhanced replacement for sh, the original Unix shell program written by Steve Bourne.^[7]

Like Windows, a Unix-like operating system such as Linux organizes its files in what is called a hierarchical directory structure. This means they are organized in a tree-like pattern of directories (sometimes called folders in other systems), which may contain files and other directories. The first directory in the file system is called the root directory. The root directory contains files and sub directories, which contain more files and sub directories, and so on.^[7]

10.2 通过 gitHub 平台实现本项目的全球域名


10.3 创建一个空的远程代码仓库

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk ().*

Owner *

 masterLijh ▾

Repository name *

✓ userName.github.io is available.

Great repository names are short and memorable. Need inspiration? How about **expert-rotary-phone** ?

Create repository

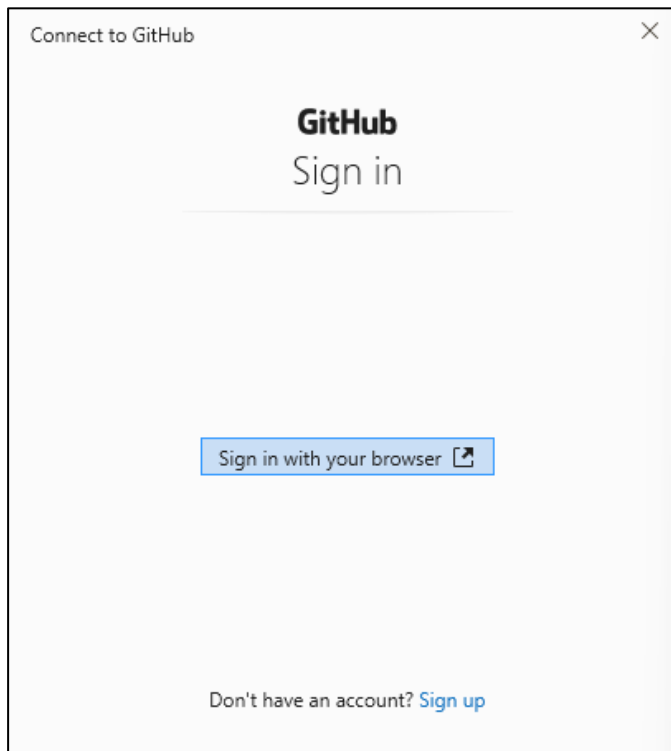
点击窗口右下角的绿色“Create repository”，则可创建一个空的远程代码仓库。

10.4 设置本地仓库和远程代码仓库的链接

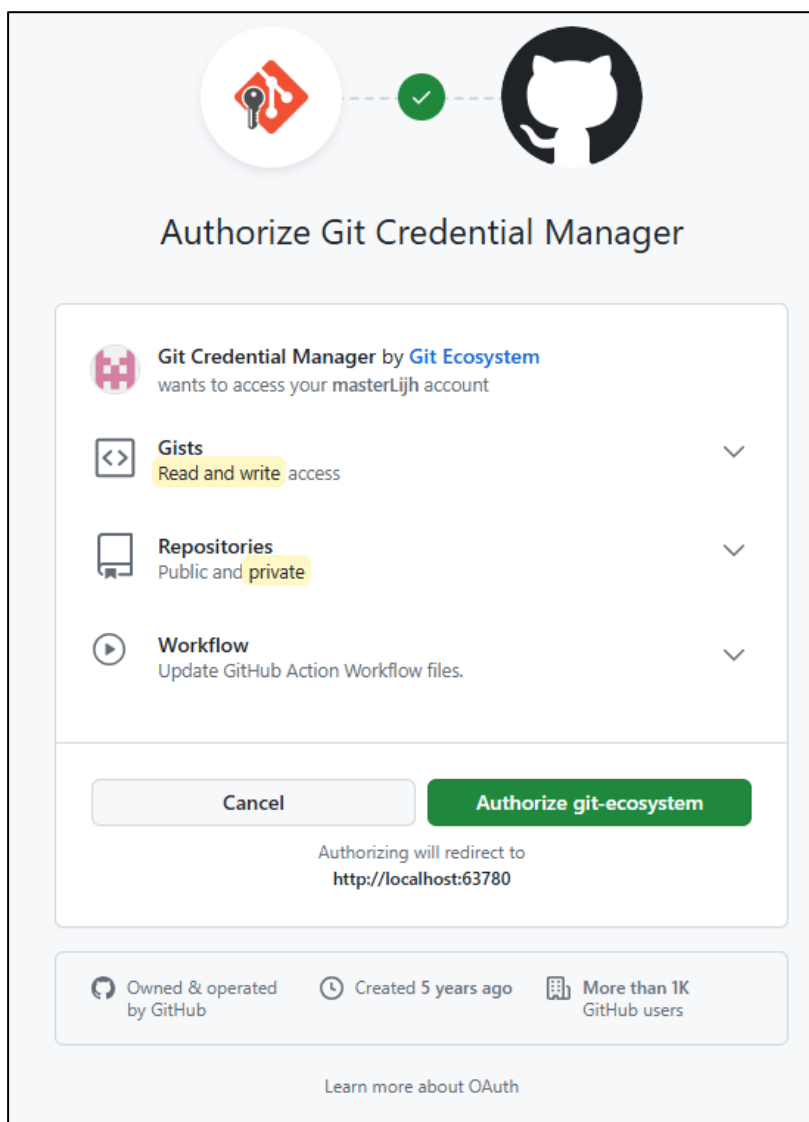
进入本地 webUI 项目的文件夹后，通过下面的命令把本地代码仓库与远程建立密钥链接

```
$ echo "WebUI 应用的远程 http 服务器设置" >> README.md
$ git init
$ git add README.md
$ git commit -m "这是我第一次把代码仓库上传至 gitHub 平台"
$ git branch -M main
$ git remote add origin
https://github.com/masterLijh/userName.github.io.git
$ git push -u origin main
```

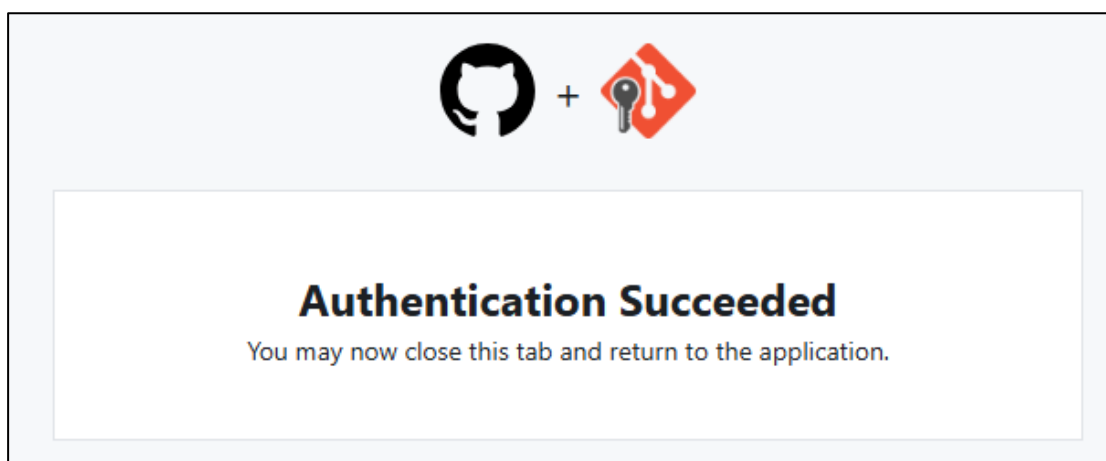
本项目使用 window 平台，gitbash 通过默认浏览器实现密钥生成和记录，第一次链接会要求开发者授权，如下图所示：



再次确认授权 gitBash 拥有访问改动远程代码的权限，如下图所示：

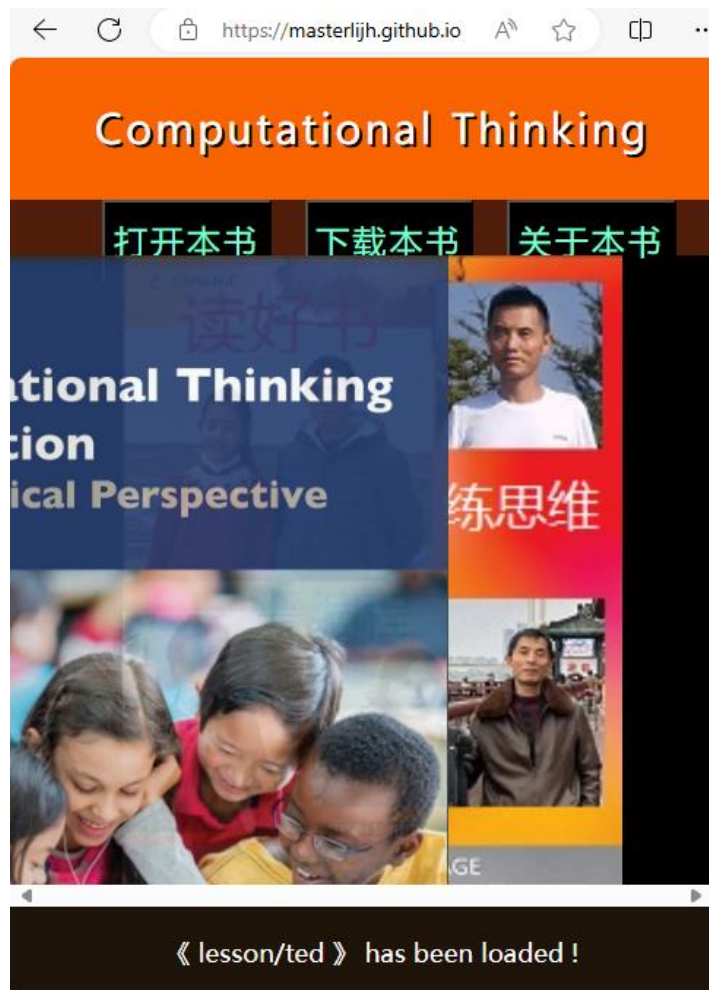


最后，GitHub 平台反馈：gitBash 和 gitHub 平台成功实现远程链接。



从此，我们无论在本地做了任何多次代码修改，也无论提交了多少次，上传远程时都会把这些代码和修改的历史记录全部上传 github 平台，而远程上传命令则可简化为一条：git push，极大地方便了本 Web 应用的互联网发布。

远程代码上传后，项目可以说免费便捷地实现了在互联网的部署，用户可以通过域名或二维码打开，本次使用 PC 的微软 Edge 浏览器打开，本文截取操作中间的效果图，如下所示：



全文完成，谢谢！

参考文献:

- [1] W3C. W3C's history. W3C Community. [EB/OL]. <https://www.w3.org/about/.https://www.w3.org/about/history/>. 2023.12.20
- [2] Douglas E. Comer. The Internet Book [M] (Fifth Edition). CRC Press Taylor & Francis Group, 2019: 217-218
- [3] John Dean,PhD. Web programming with HTML5,CSS,and JavaScript[M]. Jones & Bartlett Learning,LLC. 2019: 2
- [4] John Dean,PhD. Web programming with HTML5,CSS,and JavaScript[M]. Jones & Bartlett Learning,LLC. 2019: xi
- [5] Behrouz Forouzan. Foundations of Computer Science[M](4th Edition). Cengage Learning EMEA,2018: 274--275
- [6] Marijn Haverbeke. Eloquent JavaScript 3rd edition. No Starch Press,Inc, 2019.
- [7] William Shotts. The Linux Command Line, 2nd Edition [M]. No Starch Press, Inc, 245 8th Street, San Francisco, CA 94103, 2019: 3-7