

学号：20193016

姓名：蔡欣彤

1、请填写完整论文的 URL 地址。

完整论文 URL 地址	<a href="https://codercxt.github.io/WebUI.html">https://codercxt.github.io/WebUI.html</a>
-------------	---

2、请填写完整论文的二维码。



3、完成论文的摘要的编写。

### 基于 Web 客户端技术的个性化 UI 设计和实现

**摘要：**近十年来，html5 为核心的 web 标准的软件开发技术以其跨平台、开源的优势广泛地运用在各个领域的应用软件开发中。通过分析本次毕设任务，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 语言进行内容建模、css 语言展开 UI 的外观设计、javascript 语言编程实现 UI 的交互功能，除直接使用了 web 客户端最底层的 API 外，本项目的每条代码都是手工逐条编写，没有导入他人的任何的代码（框架和库）。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；另外大量地运用了面向对象的程序设计思想，比如用代码构建了一个通用的 pointer 模型，该模型仅用一套代码就实现了对鼠标和触屏的控制，实现了高质量的代码，这也是本项目的亮点。从工程管理的角度看，本项目采用的增量式开发模式，以逐步求精的方式展开了六次代码的增量式重构 ADIT（A:分析 D:设计 I:实现 T:测试），比较愉快地实现项目的设计开发和测试。从代码的开源和分享的角度看，本项目采用了 git 工具进行版本管理，在漫长的开发过程中重构代码六次并正式做了代码提交，另外在测试中修改提交了代码两次，最后利用 gitbash 工具 把本项目的代码仓库上传到著名的 github 上，再利用 github 提供的 http 服务器，本项目实现了 UI 应用在全球互联网的部署，我们可以通过地址和二维码便捷地跨平台高效访问这个程序。

**关键词：**Web 客户端技术；html5；github；css；javascript；代码编译器 gui；

#### 4、完成论文的第2章技术总结和文献综述的编写（要求图文配合，格式规范）。

## 1.引言

在信息化快速发展的今天，Web 应用已成为日常生活和工作的核心，Web 应用作为互联网的重要组成部分，其界面设计直接影响用户的使用体验和满意度。传统的 Web 界面设计往往采用统一的风格和布局，缺乏个性化和用户交互的考虑，导致用户在使用过程中感到单调乏味，难以获得良好的体验。因此，如何设计和实现个性化的 UI 界面，提高用户的使用体验和满意度，成为当前 Web 应用设计领域的重要研究方向。

### 1.1项目概述

本次课程设计的目标是基于 HTML5 技术，设计和实现一个个性化的用户界面(UI) 应用程序。通过运用 HTML5、CSS3 和 JavaScript 等前端技术，我们构建了一个功能丰富、交互流畅的 Web 应用，允许用户根据个人喜好自定义界面风格、布局和交互效果。在项目开发过程中，我们充分利用了 GitHub 这一强大的协作工具，进行代码托管、版本控制、团队协作和代码审查，确保项目开发的高效与规范。我们致力于为用户提供更加符合个人喜好的网页界面，提升整体用户体验。

### 1.2研学计划

#### （1）第 1 周

了解项目要求，分析需求和功能，初步构建应用程序的结构和逻辑。熟悉常用的 UI 设计工具的基本操作，包括画布设置、图层操作、样式设置等。了解响应式设计的重要性，学习如何为不同设备和屏幕尺寸设计 UI。

#### （2）第 2 周

实现网页窄屏代码和鼠标模型 1.0。显示鼠标的实时坐标。学习如何合理排版页面中的品牌标识、图片、文字等元素，确保用户能够轻松获取信息。

#### （3）第 3 周

实现鼠标模型 2.0，支持触屏功能。实现页面元素的拖动，显示拖动距离。学习如何构建清晰、易于理解的信息架构，使用户能够快速找到所需信息。设计简单明了的导航菜单，考虑用户的行为习惯，创建合理的层级结构。

#### （4）第 4 周

实现键盘控制 1.0，显示按下键的字符以及 key 和 keyCode 值。

(5) 第 5 周

实现键盘控制 2.0，利用 for 循环，显示单个字符。

(6) 第 6 周

实现键盘控制 3.0，实现文本删除与换行功能。

(7) 第 7 周

测试 web 应用程序，优化项目代码，全部上传到 GitHub 仓库。

## 1.3 研学方法

为了确保本项目的科学性和实用性，本论文采取了六种研究方法。

### 1.3.1 文献综述法

文献综述法是我们进行研学活动的起点和基础。在此阶段，我们将广泛搜集、整理和分析关于 Web 客户端技术、个性化 UI 设计以及实现方法的相关文献。这些文献可以来自学术期刊、技术博客、专业书籍、在线论坛等。通过对这些文献的深入研究，我们可以全面了解 Web 客户端技术的最新发展趋势、个性化 UI 设计的原则和最佳实践，以及在实际开发中常用的技术和工具。同时，文献综述还能帮助我们识别出当前领域的研究热点和难点，为后续的研究工作提供方向。

### 1.3.2 原型设计法

原型设计法是一种在 UI 设计过程中常用的方法。在个性化 UI 设计中，我们将采用原型设计法来快速构建出 UI 设计的初步方案。通过原型设计，我们可以将抽象的设计概念转化为具体的视觉形式，让开发者和用户能够直观地看到设计效果。同时，原型设计还可以帮助我们进行用户测试和反馈收集，以便及时发现设计中存在的问题并进行改进。在原型设计过程中，我们将使用专业的设计工具（如 Sketch、Figma 等）来创建高保真度的 UI 设计原型，并通过多次迭代和优化来完善设计方案。

### 1.3.3 增量开发法

增量开发法是一种将软件开发过程划分为一系列小步骤的方法。在个性化 UI 设计和实现的过程中，我们将采用增量开发法来逐步推进项目的开发进度。具体而言，我们将根据原型设计的结果，将 UI 设计划分为若干个模块或组件，并为每个模块或组件制定详细的开发计划。然后，我们将按照计划逐步开发这些模块或组件，并在每个阶段进行严格的测试和评估。通过增量开发法，我们可以确保项目的质量和进度得到有效控制，同时降低开发风险。

### 1.3.4 交互设计

交互设计是个性化 UI 设计中至关重要的一环。在交互设计过程中，我们将关注用户与 Web 应用之间的交互方式和流程。我们将深入研究用户的操作习惯、心理需求和期望，并据此设计出符合用户需求的交互方式和流程。具体而言，我们将关注以下几个方面：

- (1) 导航设计：设计清晰、直观的导航结构，帮助用户快速找到所需信息。
- (2) 信息架构：合理组织页面内容，确保信息的层次结构和逻辑关系清晰明了。
- (3) 交互反馈：及时响应用户操作，并给予明确的反馈，提高用户的使用体验。
- (4) 可用性测试：通过用户测试来评估交互设计的有效性，并根据测试结果进行改进。

### 1.3.5 测试评估

测试评估是确保 Web 应用质量和用户体验的关键环节。在个性化 UI 设计和实现的过程中，我们将采用多种测试方法来评估 Web 应用的质量和性能。具体而言，我们将进行以下测试：

- (1) 功能测试：验证 Web 应用的功能是否按照需求规格说明书的要求正确实现。
- (2) 性能测试：测试 Web 应用的响应时间、吞吐量、并发用户数等性能指标，确保应用能够满足用户需求。
- (3) 兼容性测试：测试 Web 应用在不同浏览器、操作系统和设备上的兼容性。
- (4) 安全性测试：测试 Web 应用的安全性，包括防止 SQL 注入、跨站脚本攻击等安全漏洞。
- (5) 用户体验测试：通过用户测试来评估 Web 应用的易用性、美观度和用户满意度等指标。

### 1.3.6 版本控制法

版本控制法是一种用于管理代码变更的分布式版本控制系统。在个性化 UI 设计和实现的过程中，我们将采用版本控制法来管理项目的代码和文档。我们将为每个项目创建一个 Git 仓库，并在开发过程中将代码和文档提交到仓库中进行版本控制。通过 G 版本控制法，我们可以方便地追踪代码的变更历史、协作开发、合并代码和回滚错误等。同时，Git 还支持分支管理功能，允许我们在不影响主分支的情况下进行新功能或修复错误的开发。这有助于我们保持项目的稳定性和可维护性。

## 2.Web 平台和客户端技术概述

### 2.1Web 平台

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，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[1]。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，是我的毕设项目应用的技术路线。

Web 平台，又称为全球广域网或万维网（World Wide Web），是一个庞大的、分布式的、基于超文本的信息系统。该平台由各种网站、网页和应用构成，为用户提供丰富的信息资源和服务。Web 平台是全球性的信息交流平台，允许用户通过浏览器访问各种网络资源。它具有高度的跨平台性，用户可以在不同操作系统和设备上访问 Web 应用。

### 2.2Web 编程

Web 编程是一个很大的领域，不同类型的 Web 编程由不同的工具实现。但是，基本上所有的工具都使用核心语言 HTML。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，可以看作用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。

### 2.3软件开发的过程管理——增量式开发模式

软件开发生命周期通常被划分为四个阶段，每个阶段都有其特定的任务和关键活动。这四个阶段分别是：需求分析阶段、设计阶段、开发阶段、测试阶段。

软件生命周期模型有很多种，它们有各自的特色、优缺点和适用领域。一般来说，采用不同模型开发的软件产品，其生命周期也有所不同。常见的软件生命周期模型包

括:瀑布模型、快速原型模型、增量模型、螺旋模型、喷泉模型、统一软件开发模型过程, 以及敏捷模型与极限编程。在这里, 我们只讨论两个最常见的软件开发模型。

### 2.3.1瀑布模型

瀑布模型是 20 世纪 80 年代之前最受推崇的软件开发模型, 它是一种线性的开发模型, 具有不可回溯性。开发人员必须等前一阶段的任务完成后, 才能开始进行后一阶段的工作, 并且前一阶段的输出往往就是后一阶段的输入。由于它的不可回溯性, 如果在软件生命周期的后期发现并要改正前期的错误, 那么需要付出很高的代价。传统的瀑布模型是文档驱动的, 如图 2.1 所示。

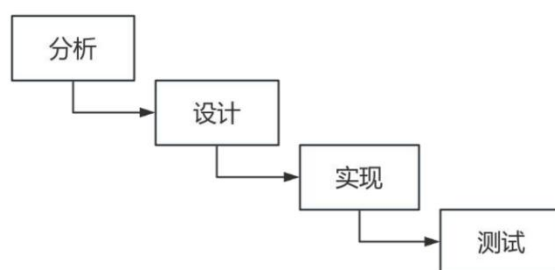


图 2.1 瀑布模型

瀑布模型的优点是过程模型简单, 执行容易; 缺点是无法适应变更。瀑布模型是一种注重规划和文档化、强调阶段明确和单向转移、严格控制变更和强调质量和测试的软件开发过程模型。它适用于需求稳定、可预测的大型、复杂软件项目, 但开发周期较长, 无法适应快速变化的需求。

### 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。

一般情况下, 开发人员会首先实现提供基本核心功能的增量组件, 创建一个具备基本功能的子系统, 然后再对其进行完善。增量模型如图 2.2 所示。

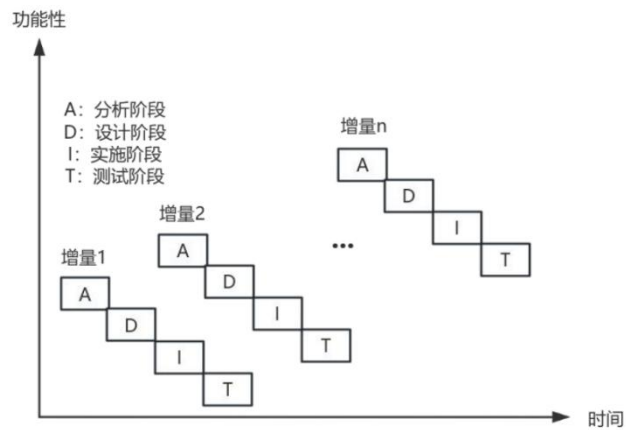


图 2.2 增量模型

增量模型的最大特点就是待开发的软件系统模块化和组件化。基于这个特点，增量模型具有以下优点。

(1) 将待开发的软件系统模块化，可以分批次地提交软件产品，使用户可以及时了解软件项目的进展。

(2) 以组件为单位进行开发降低了软件开发的風險。一个开发周期内的错误不会影响到整个软件系统。

(3) 开发顺序灵活。开发人员可以对组件的实现顺序进行优先级排序，先完成需求稳定的核心组件。当组件的优先级发生变化时，还能及时地对实现顺序进行调整。

增量模型的缺点是要求待开发的软件系统可以被模块化。如果待开发的软件系统很难被模块化，那么将会给增量开发带来很多麻烦。

