

华中科技大学

课程实验报告

课程名称： 新生实践课

专业班级 CS2405

学 号 U202414678

姓 名 杨胜钧

指导教师 陈加忠

报告日期 2021 年 11 月 11 日

计算机科学与技术学院

目 录

1 网页整体框架	1
2 主页设计	2
3 分页面设计	3
3.1 分页 1: 我的联系方式.....	3
3.2 分页 2: 我喜欢的音乐.....	4
3.3 分页 3: 我玩的游戏	5
3.4 彩蛋页	6
4 网页设计小结	7
5 课程的收获和建议	8
5.1 计算机基础知识	8
5.2 文档撰写工具 LaTeX.....	8
5.3 编程工具 Python	8
5.4 版本管理软件 Git	8
5.5 网页制作 Dreamweaver	9

1 网页整体框架

网页整体框架如下图所示

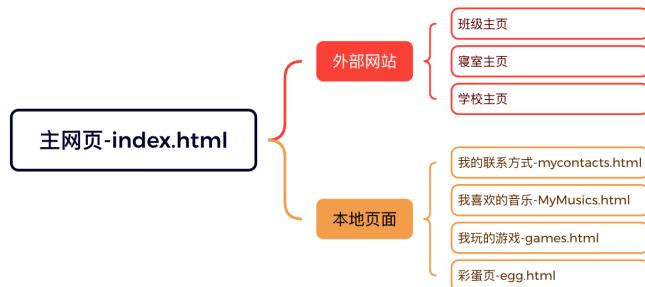


图 1-1 网页框架示意

网页采用“总-分”式结构，由主网页 index.html 串联起所有本地分支页面，分页面可以通过内置的返回键返回主页，但分页面间没有直接的互通途径。此外，主页里还留有通往班级主页、寝室主页和学校官网的路径。

2 主页设计

主页截图如下图所示。



图 2-1 主页

主页通过定义数个类并将其运用到元素上，实现了为部分元素添加动画，并使得日后的改进与添加等操作更为方便。主页的标题和按键部分通过 CSS 的关键帧动画功能实现了流畅的动画演示，让其在进入主页时具有“浮现”的动画效果。主页的元素叠加功能主要是通过为类设置“position:absolute”和“position:relative”的属性实现的，为父元素设置相对位置属性的同时为其子元素设置绝对位置属性，同时通过 z-index 属性设置其图层，即可相对灵活地布置网站页面。

此外，主页中还设置了一个隐藏区域，鼠标点击此区域可进入分页面“彩蛋页”，感兴趣的话可以自己找一找。

3 分页面设计

3.1 分页 1: 我的联系方式

截图如下。

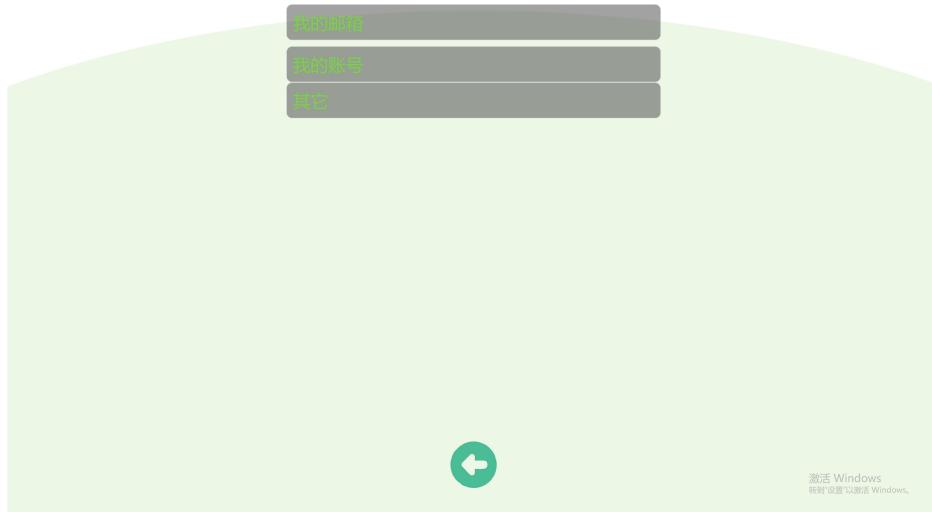


图 3-1 联系方式页面

联系方式页面布置了一个内含三行的 `` 块，用户可以通过点击列表标题来展开或折叠其内容。列表折叠与展开的内容运用了 JavaScript 块来实现，以下是其具体代码：

```
▼<script> == 30
document.querySelectorAll('.collapsible-header').forEach(function(header) {
  header.addEventListener('click', function() {
    var content = this.nextElementSibling;
    if (content.style.display === 'block') {
      content.style.display = 'none';
    } else {
      content.style.display = 'block';
    }
  });
});
```

图 3-2 `<script>` 块

代码为每一个使用“header”类的元素添加了事件监听器，当鼠标点击时便会激活其对邻近“content”元素的 `display` 属性以决定其是否被显示在网页中。

本页在页面底部添加了一个返回键，点击其就可以返回主页。同时，该返回键设置了绝对位置属性，使得无论页面如何移动其均固定于页面底部。

考虑到个人隐私问题，所有的联系方式都只以“XXX”示意。

3.2 分页 2: 我喜欢的音乐

截图如下。



图 3-3 我喜欢的音乐

音乐页面使用表格列出了八首用 `<audio>` 块显示的可播放的音乐，每首音乐有独立的控制条以控制其音量、进度条等设置。

本页还利用 JavaScript 块实现了每播放一首音乐，页面下方就会显示其对应的唱片封面的功能。以下是其具体代码：

```

<script> = >
    var audioElements = document.querySelectorAll('audio');
    var myimg = document.getElementById("myimage");

    // 为每个音频元素添加事件监听器
    audioElements.forEach(function(audio) {
        audio.addEventListener('play', function() {
            // 播放时显示对应的唱片封面
            switch (audio.id) {
                case 'audio1':
                    myimg.src = '../image/audio1.jpg';
                    break;
                case 'audio2':
                    myimg.src = '../image/audio2.jpg';
                    break;
                case 'audio3':
                    myimg.src = '../image/audio3.jpg';
                    break;
                case 'audio4':
                    myimg.src = '../image/audio4.jpg';
                    break;
                case 'audio5':
                    myimg.src = '../image/audio5.jpg';
                    break;
                case 'audio6':
                    myimg.src = '../image/audio6.jpg';
                    break;
                case 'audio7':
                    myimg.src = '../image/audio7.jpg';
                    break;
                case 'audio8':
                    myimg.src = '../image/audio8.jpg';
                    break;
                default:
                    break;
            }
        });
    });
</script>

```

图 3-4 `<script>` 块

代码功能已经在图3-4的注释中阐明。

3.3 分页 3: 我玩的游戏

截图如下。

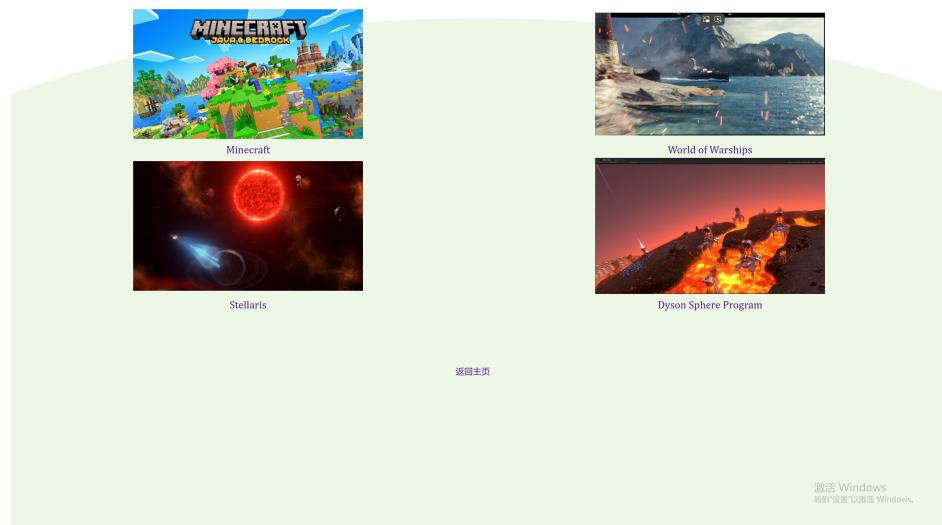


图 3-5 游戏

本页面利用表格和 `<div>` 分块列出四个游戏，每个游戏都给出了游戏名，并在上方有配图。点击游戏名超链接即可进入对应游戏官网。

而每个游戏配图都会在鼠标经过时放大，并在点击后将页面背景替换成对应游戏的宣传图。该功能同样通过 JavaScript 实现，具体代码如下：

```

<script>
    var myimg=document.getElementById('background');
    var buttons=document.querySelectorAll('.buttonposition');
    buttons.forEach(function(bgim){
        anim.addEventListener('mouseenter',function(){
            anim.style.transform = 'transform: scale(0.85)';
            anim.style.transition = 'scale(1.2) 2s';
        });
        anim.addEventListener('mouseleave',function(){
            anim.style.transform = 'scale(1.0)';
        });
    });
    buttons.forEach(function(bgi){
        bgi.addEventListener('click',function(){
            switch(bgi.id){
                case 'button1':
                    myimg.src='../image/Minecraft.png';
                    break;
                case 'button2':
                    myimg.src='../image/wows.jpg';
                    break;
                case 'button3':
                    myimg.src='../image/Stellaris.jpg';
                    break;
                case 'button4':
                    myimg.src='../image/DSP.jpg';
                    break;
                default:
                    break;
            }
        });
    });
</script>

```

图 3-6 `<script>` 块

代码为每一个具有“buttonposition”的元素添加了事件监听器，当鼠标移入时(事件“mouseenter”)会播放放大动画；当鼠标移出时(事件“mouseleave”)会播放缩小动画以重置图片；当鼠标点击时(事件“click”)会更换背景图的 `<src>` 属性，以将背景更换为对应的图片。

3.4 彩蛋页

截图如下。

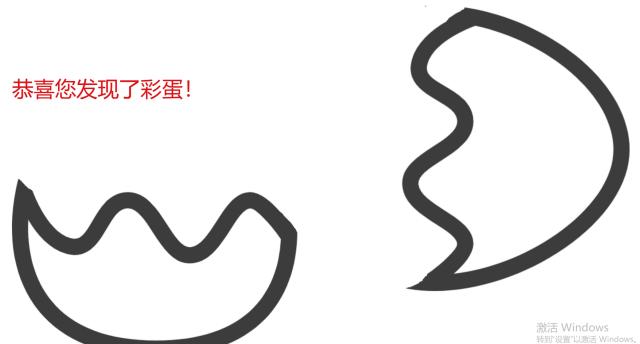


图 3-7 彩蛋页

本页面在主页没有明显的方法进入，需要找到主页上的一个隐藏区域以点击进入（所以算是埋藏的一个小彩蛋）。进入页面后会播放一个动画，图中的蛋会裂开并显示一行文字“恭喜您找到了彩蛋”。

本页的动画由 CSS 的关键帧功能实现，并通过延时实现了不同动画的顺序播放。本页内容较少，也没有用到 3-6 那样的 JavaScript 模块，但考虑到作者调动动画参数很麻烦，工作量其实也不少。

4 网页设计小结

作者的网页的一大特色便是通过 JavaScript 实现了一些复杂功能。通过课堂上的资料和网络上的部分教程，作者已经掌握了使用 HTML5 构建网页的一些基础知识。但在实际构建过程中也碰到了一些问题，列举如下：

- 1) 使用的工具 Dreamweaver 的功能有限，其内置的功能很多时候达不到想要的效果。
- 2) 很难将元素定位在想要的位置。
- 3) 第一次接触 Java 语言，对语法等不熟悉，且因为代码不会报错，故有时出错了也很难排查。

作者也做出了针对性的解决方法：

- 1) 直接手打代码，只将 Dreamweaver 作为一个实时查看工具和 IDE 来使用。
- 2) 多次更改参数进行调试，直到达到想要的效果为止。
- 3) 在遇到错误时使用 ai 工具进行改错与代码改进，同时通过阅读 ai 生成的代码，作者的 Java 水平也有了初步提升。

5 课程的收获和建议

5.1 计算机基础知识

在这门课程中学习了计算机的相关历史和基本使用知识，对于刚入学的新生来说十分实用。老师提出的“少装电脑管家”等建议成功让许多同学免受一些流氓软件的折磨，而有关资源管理器等电脑基本工具的使用更是让“电脑盲”们受益匪浅。

5.2 文档撰写工具 LaTeX

在课前相关资源的准备和下载过程中就已经学到了许多内容，而 LaTeX 工具本身对于将来会频繁撰写实验报告和论文等的大学生也有很大帮助。美中不足的是课堂上所学的指令较为单薄，编译器本身也可能有些问题，如图片、表格等元素插入位置不准确等。

5.3 编程工具 Python

在本专题中初步接触了一种全新的语言 Python 并体会到了这门语言强大的灵活性，不同语言的学习也可以帮助我们在学习其它语言的同时转换思维，增强逻辑能力。很可惜课时有限，许多内容来不及讲，但还是建议在之后的 Python 教学中增加对流程控制语法的讲解。

5.4 版本管理软件 Git

通过了解版本管理软件 Git 的使用方法，我们得以在日后的项目实践中更加方便地对自己的项目文件进行管理，同时也得到了一个获取网络上各种资源的便捷渠道。然而作者在学习本课程后仍然认为自己对 Git 了解不足，只知复制粘贴指令，却不知到底问什么要用这些指令，本质上还是对 Git 中本地仓库、远程仓库、工作区等概念不熟悉，建议日后的教学活动中可以对这些仓库、区域等概念与它们的结构、各自之间沟通要用到的指令等多加介绍。

5.5 网页制作 Dreamweaver

在本课程中，初步了解了网页制作的基本语言 HTML5 并掌握了其编辑工具 Dreamweaver 的基本使用方法。但该工具用法复杂、能力有限，很多时候直接对代码区编辑甚至比通过 Dreamweaver 进行编辑还要方便得多，故建议在日后的教学中可以将重心放在 HTML5 的语法与功能介绍中，并适当舍去有关 Dreamweaver 的部分内容。