网络应用开发课程设计

小游戏合集设计报告

**组员信息：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 姓名 | 学号 | 成员贡献 | 课程设计成绩 |
| 1 | 田皓月 | 2021212143 | 文件夹MissXia07 |  |
| 2 | 李晋铖 | 2021212127 | 文件夹LJC、主页 |  |
| 3 | 徐新桐 | 2021212142 | 文件夹XXT、主页 |  |
| 4 | 明帅 | 2021212146 | 文件夹maz1caine |  |
| 5 |  |  |  |  |

https://github.com/hougedsl/web\_app\_dev\_course\_design

# **引言**

本系统设计的目的是实现多个网页小游戏，并打包为一个网页小游戏合集。

# 功能与设计

## 功能介绍

主页：包括主、副标题和四张图片，每张图片对应一个小游戏。当玩家点击图片上对应的游戏名称时，可以跳转到该游戏界面并体验游戏。

CARD MEMORY：实现一款网页小游戏。在start.html按下回车键进入Demo.html，在Demo.html按下回车键开始游戏。网页游戏中有12张卡牌背面朝上，点击卡牌将卡牌翻到正面。每有两张卡牌正面朝上，就进行判定，若图案相同则匹配成功并消除两张卡牌，若不同则重新将卡牌翻到背面并开始新一轮点击。在15秒内尽可能多地消除卡牌。

打砖块：打开网页后会自动播放背景音乐。网页左上角为游戏规则，点开后可查看。游戏界面共有45个砖块、1个小球和1个挡板，小球在移动时撞击砖块会改变方向，通过移动下方挡板接住小球并将其弹回，每撞击1个砖块积1分，撞击完所有砖块后该局游戏胜利。若小球落到下方边界，该局游戏失败。

Kun Matching Game：打开网页时，三张图片以不同的速度左右滚动；当鼠标点击到其中一格时，该格停止滚动；当三个格子均被点击一次后，玩家可以看到自己拼图是否成功，并可以选择是否重新开始游戏。

PlaneWars:首先选择4种游戏难度，分别对应不同的敌机数量、移速、大小。玩家通过鼠标移动自己的飞机并自动射击，当自己的飞机触碰到敌机被摧毁，游戏结束。玩家通过击毁敌机获得积分，游戏结束会根据所得积分获得称号。

## 系统设计

主页：在页面中放置标题、四张游戏图片、对应的游戏名称和开发人员。点击其中一个游戏名称后，可跳转到该游戏界面并调用该游戏对应部分的代码。

CARD MEMORY：

start.html：在页面中排布4\*3个卡牌背面，设置翻转动画。定义一个位置检查数组，初始调用Flip函数。Flip函数生成4个不在位置检查数组中的1-12之间的随机整数作为翻转位置，另生成4个1-6之间的随机整数作为卡牌正面，将4个翻转位置上的卡牌添加翻转效果，然后按照卡牌正面进行翻转，等待1秒后调用FlipBack函数。FilpBack函数把翻转位置的卡牌全部翻回背面并移除翻转效果，立即调用Flip函数开始新一轮翻转。

在上层添加一个位于页面中心的矩形，用于提示用户准备进行游戏。

添加事件监听器，按下回车键跳转到Demo.html。

Demo.html：在页面中排布4\*3个卡牌背面，设置翻转动画。设置排布图片数组，1-6每个数将在该数组中出现两次，然后打乱顺序。添加事件监听器，获取用户鼠标点击。被单击的卡牌添加翻转动画，然后按照排布图片数组翻到卡牌正面。两次点击后进行判定，若卡牌正面相同，则隐藏卡牌元素，更新隐藏卡牌数；否则将两张图片翻回背面，开始新一轮点击。

在网页上部添加一个进度条，设置计时器，按照计时器更新进度条。

在上层添加一个位于页面中心的矩形，用于提示用户进行游戏。

在上层添加一个位于页面中心的矩形，用于提示用户游戏结果，隐藏。

添加事件监听器，按下回车键隐藏提示信息并开始游戏。

在计时结束或卡牌全部匹配完后调用结束游戏函数，不允许点击，设置计时器进度为100%，清除计时器，显示游戏结果。

Brick Game：

主界面包括左上角的“阅读规则”按钮和中间的游戏窗口。

“阅读规则”部分：点击“阅读规则”按钮后，显示游戏规则界面；点击“关闭”按钮后，游戏规则界面隐藏。添加事件监听器，获取用户鼠标点击。

游戏窗口部分：在窗口中定义1个小球、1个挡板、45个砖块、分数和边界五个元素，并设置一些基本变量，例如速度、位置、颜色等。绘制得分、小球、挡板和砖块；设置边界和挡板的移动判定；设置小球碰撞到砖块、挡板、边界后的运动情况，并说明若小球撞到下方边界，分数归零；每当小球撞击一个砖块，得分加1，撞击完所有砖块后，得分清零；小球撞击到所有砖块后，砖块恢复初始数量；更新所有绘制函数和动画；设置按键的按下与释放效果。添加事件监听器，获取用户鼠标点击和键盘按键的按下与释放。

Kun Matching Game的核心要点在于控制复选框的动画与停止。在css文件中，三个复选框被竖向排列。三张1:1大小的图片被横向分为三个部分，放入三个复选框中，并创建css动画。此动画效果为图片从复选框最左边滚动到最右边，循环滚动并且只显示三次，即左、中、右三个部分。玩家在点击复选框后，该复选框动画停止，玩家可以自行判断拼图是否完成。若没有完成，玩家可以选择重新开始游戏。

在上述游戏开始时，网页会自动播放动感的音乐。在游戏的过程中，当玩家点击复选框时，该复选框不允许再被点击。当玩家选择重新开始游戏后，解除复选框点击限制。

PlaneWars:

选择不同的游戏难度，根据游戏难度创建背景板；

创建玩家战机，鼠标进入游戏面板后其随鼠标轨迹运动；

创建子弹，使得子弹周期性的在玩家战机处发出并让其向上移动

创建敌机，让敌机周期性的在游戏背景板顶部随机位置产生，并让其向下移动；

定义碰撞函数，子弹和敌机相遇时进行碰撞检测，计算得分；战机碰撞结束游戏

定义结算函数，显示得分和荣誉称号。

# **功能实现**

技术方案：

主页：

HTML：定义网页的结构和内容，包括标题、图片等，并添加四个网页游戏的链接

CSS：设置网页的外观和布局，包括全局样式设置、背景图片设置、页面头部和文字元素、游戏开发者信息等。

CARD MEMORY：

HTML用于创建网页结构和内容，CSS用于定义网页的样式和布局，JavaScript用于实现网页的交互和动态效果。在这个项目中，使用这三种技术实现了一个简单的卡牌记忆游戏。

HTML部分定义了网页的基本结构，包括标题、提示信息、卡牌。进度条等元素。

CSS被用于样式设计，涵盖了布局、颜色、阴影、动画效果等方面，以确保游戏界面的美观和用户友好性。

JavaScript则负责实现游戏的逻辑功能，包括隐藏与显示、交互和重置、计分与计时等功能。

Brick Game：

HTML：用于创建网页结构和内容。

CSS：用于定义网页的样式和布局。

JavaScript：用于实现网页的交互和动态效果。

Canvas：用于在网页上绘制图形和动画。

Audio：用于在网页上播放音频文件。

相对路径：使用相对路径引用了外部的样式表、脚本文件和音频文件。

Kun Matching Game：使用HTML、CSS、JavaScript实现一个简单的图片匹配游戏。

HTML：定义网页的结构，包括标题、提示信息、开发者信息、音频、复选框、重置按钮。

CSS：对网页进行样式设计，包括字体、布局、颜色等。

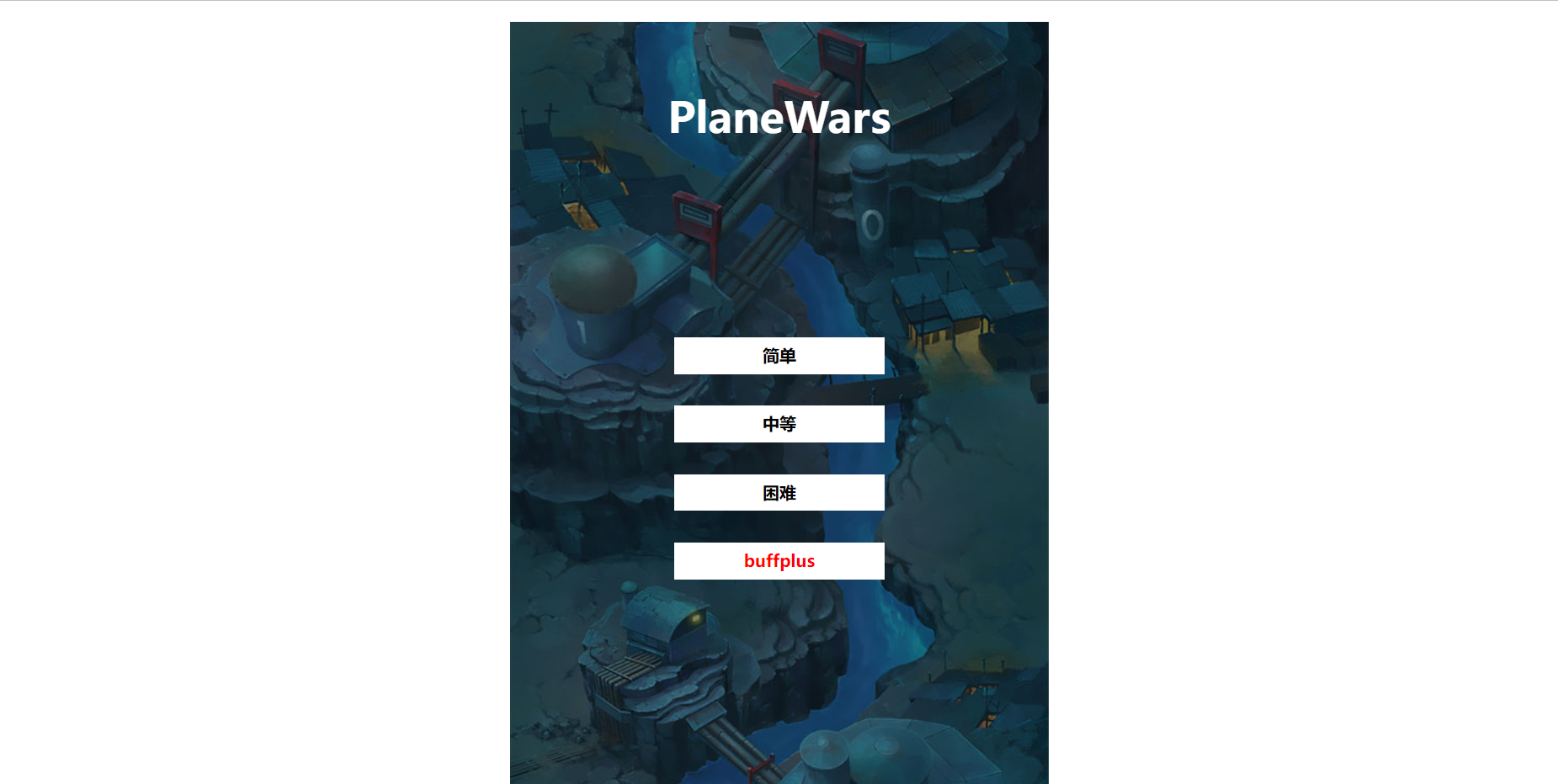
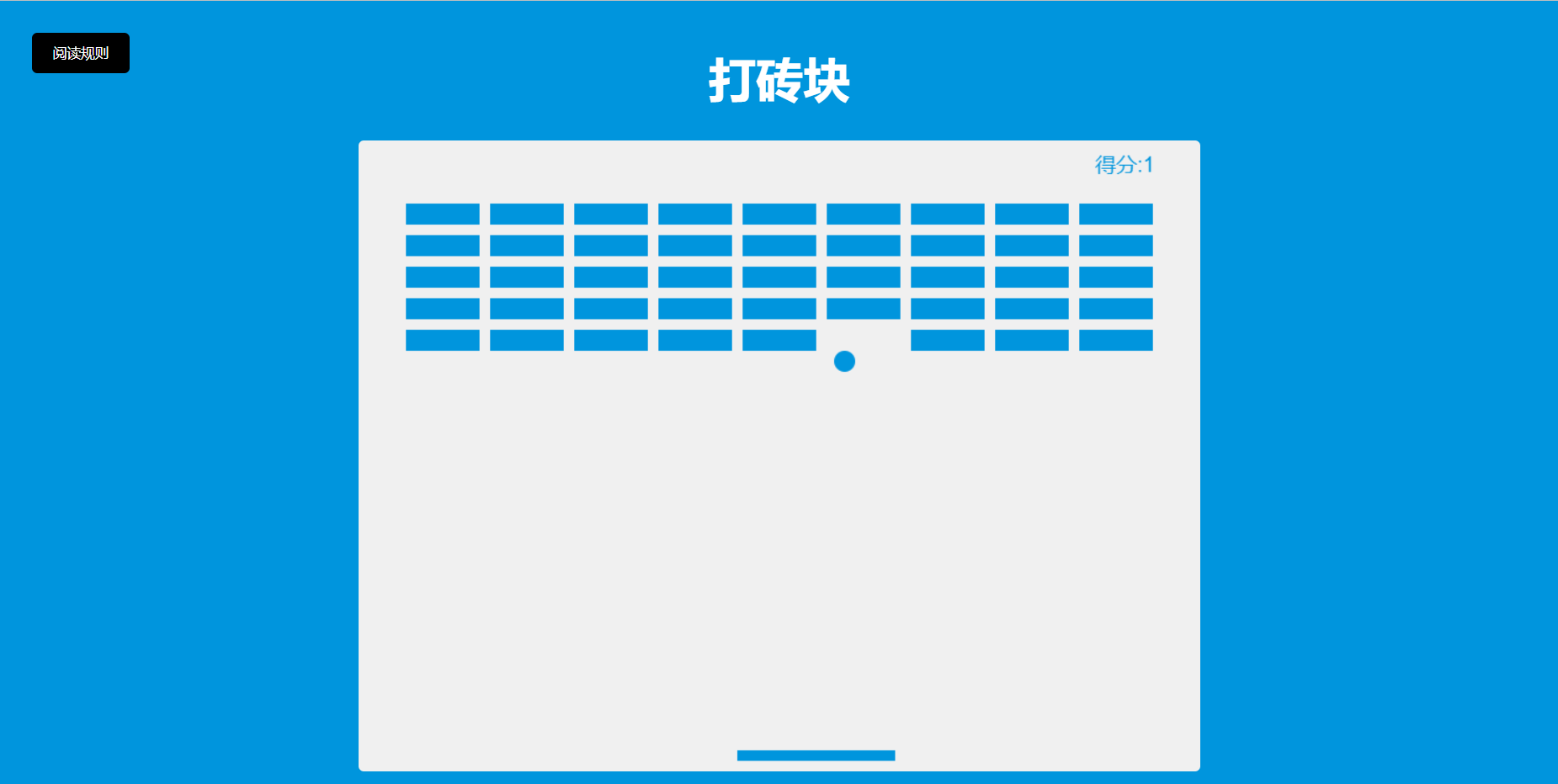
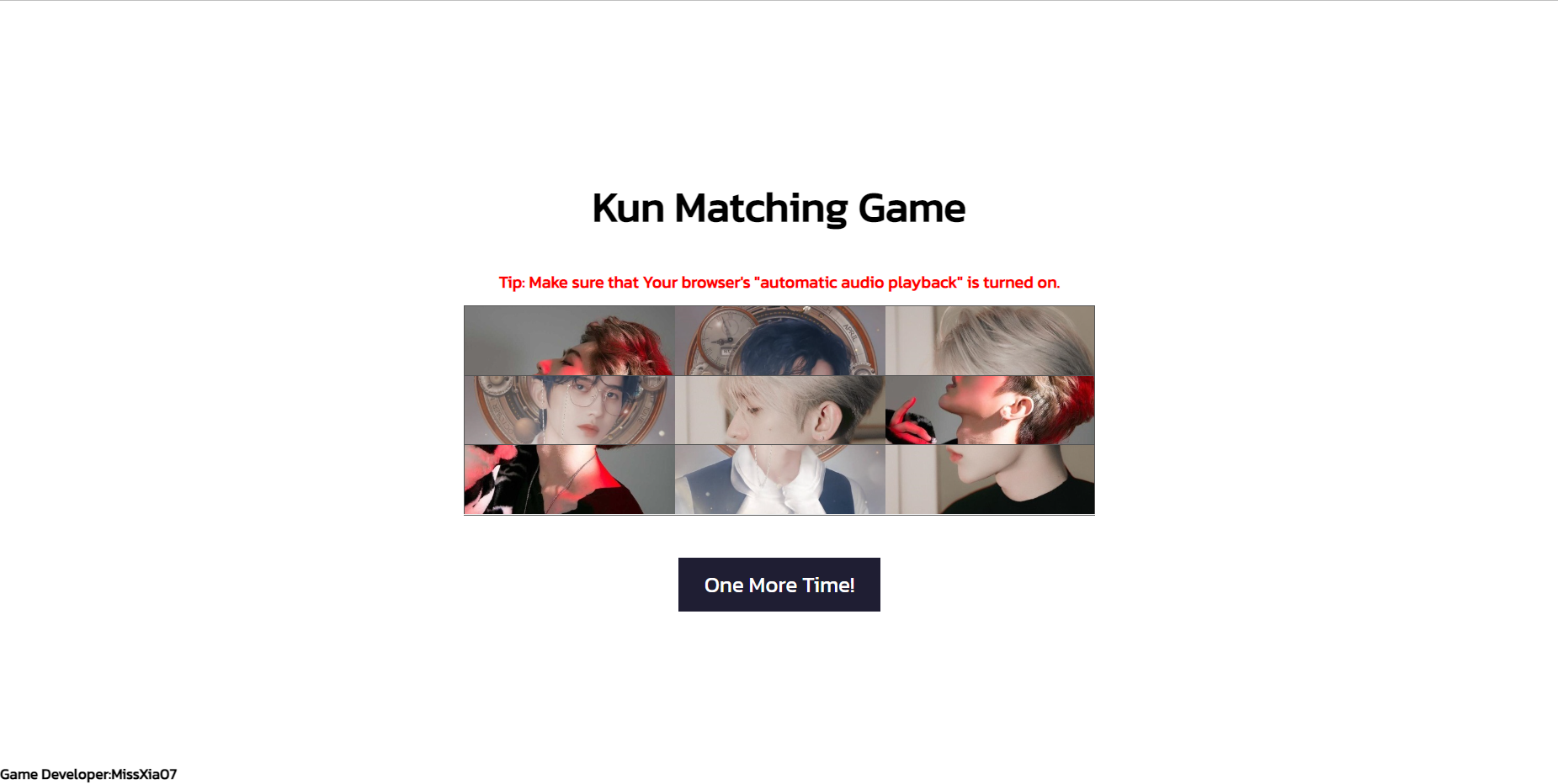
JavaScript：实现游戏的逻辑功能，包括复选框的交互与重置功能的实现。

PlaneWars:

基于HTML，CSS，JavaScript完成动画交互，利用document.createElement(element)创建元素，element.id、element.style、element.innerHTML 等为其设置相关属性，document.removeChild(element)删除HTML元素，可以结合父元素使用，如我们删除脱离游戏面板的子弹，先找到子弹，再删除该子弹自身，其实就是删除该元素的父元素的子元素；document.appendChild(element)添加HTML元素，每一个创建完的DOM元素都需要添加后才能显示，如我们将创建的我方战机添加到body中。

# **功能展示与评测**

通过对代码进行整合和测试，结果表明每个部分的功能都能够正常显示和运行，即该系统有很高的可用性和完整性。部分具体运行结果如下：





# **总结**

田皓月：做成Kun Matching Game这个小网页游戏使我对网页设计有了初步的实际认识。其中也有一些问题没有解决，比如拼图游戏中"如何判断是否成功"模块，在尝试了一些方法之后仍不知如何实现。之后会持续学习，希望会有更加深刻的理解。

徐新桐：Brick Game是我自己做的第一个网页小游戏。通过对本学期知识的学习和应用，编写的游戏能够实现一些基本的功能，包括创建并移动元素、设置分数等。但是还有一些功能在对代码进行修改后仍未实现，例如：①游戏开始时小球位于挡板上，通过鼠标点击屏幕使小球开始运动；②使挡板的运动方向和速度能够影响小球的运动方向和速度。我之后还会对代码进行修改和完善，从代码简化、页面美化、功能优化等方面争取进一步的提高，增强自己的前段代码编写水平。

明帅：我在PlaneWars小游戏的开发中，使用了一些常见的前端技术，如DOM操作、事件监听和动画效果等。通过对游戏界面元素的创建和定位，以及事件的监听和处理，成功实现了飞机的移动和射击，敌军飞机的生成和移动，以及碰撞检测和爆炸效果等功能。这些开发过程提升了我对前端开发的实际应用能力，同时我发现了代码中的一些优化点。例如，可以将一些重复的代码提取为函数，提高代码的重用性和可维护性。另外，可以使用更加语义化的标签和类名，使代码更易于阅读和理解。此外，对于游戏开发，需要注意性能优化和代码结构的规划。在游戏中，需要处理大量的元素和事件，所以需要注意性能方面的优化，避免出现卡顿和延迟的情况。合理划分代码结构，使代码逻辑清晰明了，这样可以方便后续的维护和扩展。

李晋铖：在制作这款卡牌记忆小网页游戏的过程中，我获得了对程序设计全新的认识。其中，最具挑战性的部分在于确定start.html中需要翻转的卡牌位置，在尝试了利用循环生成随机数、等待全部翻回完成后开始新一轮翻转、在上一张卡牌未翻回时开始新一轮翻转等设计后，出于程序正确运行的需要和美观的需要，最终确定了随机加数组的方式，确保功能的实现。未来，我会继续深入学习，以期对程序设计有更深入的理解。