

js项目实践：导航网页与网页小游戏制作

导航网页（index1-4）

本项目的主要功能是对北京大学学生常用网站做了一个集合，并为了方便添加了简单的todo list功能。除去官方网站，我们还添加了三个简单的小游戏聊以娱乐。

- 网页实现

本项目将常用网站分为了“休闲娱乐”、“上课作业”、“信息获取”三个部分，通过超链接将三个主页链接在一起，实现主页间的跳转。通过仿照北大官方门户，完成了页面美化。

- todo list

实现了一个基本的待办事项列表应用，用户可以通过输入框添加新任务，并通过按钮删除任务。任务列表利用浏览器的本地存储功能实现持久化，确保页面刷新后仍能显示之前添加的任务。

核心函数：

- `addEventListener()`：实现对DOMContentLoaded的监听。
- 任务操作函数：`addTask()`函数添加任务，`saveTasks()`保存任务函数，`loadTasks()`加载任务

网页小游戏

在本项目中，一共制作了2048, flappy-bird, 五子棋三个小游戏。在开发的过程中，使用了一些课程中未涉及的css内容，以及html中的canvas等元素，并使用其从精灵图绘制。

2048

该小游戏主要采取面向过程的编程范式。

- 项目概述

本项目是经典的2048游戏的一个简单实现。2048是一款单人游玩的滑动方块游戏，目标是在4x4的网格中通过合并具有相同数字的方块来创建一个2048方块。游戏通过HTML, CSS, 和 JavaScript实现，提供了基本的游戏界面和逻辑。

- 核心函数：

-
- 方块生成：`generate()`函数在空白位置生成数字为2的方块。
 - 方块移动：`moveUp()`, `moveDown()`, `moveRight()`与`moveLeft()`四个函数允许玩家用键盘来移动方块。相同数字方块在移动时会合并生成它们的和。

- 两种需要注意的情况: $2\ 2\ 2\ 2 \Rightarrow 4\ 4$, $2\ 2\ 2 \Rightarrow 2\ 4$ (按下→键), 为了优雅地解决这个问题, 采取了队列的数据结构。
- 实际上, 在按下某个方向后, 如果不会发生任何的移动与合并, 不应该生成新的方块, 这里为了简易起见没有做这一点。
- 游戏结束检测: 当所有格子都被填满, 且无相邻格子可合并时, 游戏结束。

- 动画细节:

通过对格子的`classList`添加与删除`class`, 实现了不同数字的不同显示效果, 以及方块合并、移动时的动画效果。

Flappy-bird

该小游戏是面向对象式编程的一次实践。

- 项目概述

本项目是经典移动游戏Flappy Bird的一个网页版简易实现。玩家通过点击或触摸屏幕来控制一只小鸟飞行, 避免碰撞到上下移动的管道。游戏的目标是尽可能地飞行更远, 通过更多的管道。

- 核心功能

-
- 小鸟控制: 玩家可以通过点击或触摸屏幕使小鸟“拍打翅膀”上升, 释放后小鸟会因重力下降。
 - 管道生成与移动: 游戏会不断生成上下对开的管道, 玩家需要控制小鸟穿过管道的空隙。
 - 游戏结束条件: 小鸟碰到管道或飞出屏幕范围, 游戏结束。
 - 图形绘制: 调用`drawImage()`函数, 从`SPRITE.png`文件中选取部分区域绘制在画布上。注意绘制顺序。(说实话, 量这些数据真的很麻烦)

- 游戏逻辑

-
- 运动关系: 由于运动是相对的, 因而可以使镜头跟随小鸟, 体现在游戏上即为小鸟只做上下运动而管道只向左移动。
 - 管道的生成: 每隔定长(200)就会添加一对管道, 管道总是成对的。确定某一边的长度和空隙, 则另一边可知。管道需要处理自己的消失和与小鸟的碰撞。这里将两者均抽象为矩形, 便于做碰撞检测。
 - 小鸟: 小鸟受重力影响下落, 并由于点击而跳跃。另外, 小鸟不能超出画布的边界。
 - 游戏循环(`gameLoop`): 并没有用`while(true) {}`的循环, 而是用递归的方式实现, 这样是为了更简易地实现一个具有稳定帧率的游戏。

- 一个有趣的bug修复:

问题

在第一次加载页面时, 背景与logo不被绘制。这是因为我将两者的绘制放在了`DOMContentLoaded`的监听上, 而这个事件只是确保了`sprite`图像的加载和绘制逻辑在文档加载完成后执行, 但没有确保

`sprite`图像本身已经加载完成。由于图像可能还未加载完成，因而在尝试绘制背景与logo时，实际上没有任何内容被绘制。

解决方案

将背景与logo的绘制逻辑放在`sprite.onload`事件处理函数中，确保只有在`sprite`图像完全加载后才执行绘制操作。

五子棋

- 项目概述

本项目简单实现了一个五子棋小游戏，可以选择双人模式或人机对战模式。

- 核心功能

-
- 通过模式选择进入双人模式或者人机对战。并在不同的模式下使用不同的逻辑进行下棋。
 - 图形绘制：调用`drawBoard()`函数绘制棋盘。
 - 游戏结束：五个同色棋子在横/竖/斜方向连成一条线。

- 核心函数

-
- 初始化游戏：`startGame()`当监听到模式选择按钮被点击，进行游戏模式的初始化。
 - 下棋：通过对click的监听，以`handleMove(event)`处理点击事件并根据当前模式进行下一步处理。
 - 双人模式：`handleTwoPlayerMove(x, y)`处理双人模式下棋子的位置，并判断是否胜利。
 - 人机对抗：`handlePlayerMove(x, y)`和`handleComputerMove()`处理人机对战模式下玩家和电脑棋子的位置，并判断是否胜利。
 - AI逻辑：利用minimax算法进行下棋位置的选择，`evaluateBoard()`评估当前棋盘分数，`findBestMove`找到当前的评估下最优的下棋点。

分工情况

- 李天宇（2200013188）： 2048、flappy-bird、网页在github上的部署
- 李佳璇（2200013191）： 五子棋、主页
- 仓库地址：<https://github.com/lx1xxx/js>
- 网页部署地址：<https://wxm-233.github.io>