

**项目总结报告**

**组 长** 张嘉羿

**组 员** 文思茗、刘烁飞、袁俊辉

**专 业** 软件工程

**课 程** 前端开发实训

**年 级** 2022级

**2024年6月**

目 录

第一章 团队介绍

团队成员工作内容

第二章 项目需求分析

2.1 项目概述

2.2 目标用户

2.3 功能需求

2.4 非功能需求

第三章 项目概要设计

概要设计&架构设计

第四章 项目详细设计

4.1 软件详细设计流程说明

4.2 主要模块设计

4.3 数据模块

第五章 项目实现

附录：

项目Github地址

个人收获

1. 团队介绍

在当前高度信息化的时代，前端开发工作成为了软件开发生态中的核心部分。为了深入学习前端开发知识，增长项目开发经验，以应对不断变化的市场需求和技术挑战，我们组建了一支四人组成的前端开发项目团队。

成员构成为：组长张嘉羿、组员：刘烁飞、文思茗、袁俊辉。本项目中，我们旨在构建一个网页游戏集合平台，通过多元有趣的休闲益智游戏给于用户丰富的游戏体验。项目分工方面，首先，由每人各自构成一个独立的网页小游戏，该模块完成后，张嘉羿、刘烁飞、文思茗三人负责完成了集合网页的设计构成、代码实现以及四个独立游戏的内容整合，而项目在GitHub上面的建立，完善与持续更新是由袁俊辉进行的。最后的项目总结报告与readme文档中，不同的游戏模块细化说明由各自的制作者来说明流程，其余部分是张嘉羿完成总结补充的。

1. 项目需求分析

1、项目概述

本项目旨在开发一个网页小游戏平台的前端部分，通过此平台，用户可以浏览、体验各类在线小游戏。前端部分将负责呈现游戏列表、用户交互界面以及游戏运行界面等。

2、目标用户

游戏爱好者：寻找新颖有趣的在线小游戏。

游戏开发者：展示自己的游戏作品，与玩家互动。

社交用户：寻找游戏伙伴，分享游戏经验和乐趣。

3、功能需求

（1）提供游戏列表展示

（2）展示游戏的基本信息

（3）显示游戏的截图、视频预览

（4）提供“开始游戏”按钮，引导用户进入游戏界面。

（5）游戏运行界面

（6）显示游戏进度、得分、排行榜等信息。

（7）允许用户进行游戏内操作，如移动、跳跃、攻击等。

（8）反馈与帮助

4、非功能需求

（1）界面设计简洁明了，易于导航和操作。

（2）游戏加载速度快，减少用户等待时间。

（3）安全性

（4）可扩展性

（5）设计合理的架构和模块划分，方便后续功能扩展。

（6）适配不同屏幕尺寸和设备类型（如手机、平板、电脑等）。

1. 项目设计概要

项目根目录/

├── FlappyBird/

├── GameBee-master/

├── gif/

├── GoogleDino/

├── iamges/

├── pacman-main/

├── styles.css

├── video/

└── youngyoung

文件中包含了四个小游戏的子文件和主页的设计以及一个彩蛋的额外小游戏，在主页中，正中央循环播放四个游戏的动态视频简介和整个集合网页的宣传海报。

主页上方导航栏，有一行动态的背景动态条，左侧是3288小游戏logo，导航栏目中央有：首页、关于我们、分类、联系我们 的二级菜单。在点击首页可以返回主页，鼠标悬停在“关于我们”上方会显示出我们的成员和设计初衷，分类包含几种游戏的分类，鼠标悬停在“联系我们”上方会显示出项目组群聊的二维码。

主页向下滚动，可以展现出四个游戏的图标，点击可以进行跳转到对应的游戏。另外还有一个敬请期待的图标，也可以点击，点击进去后跳转到一个小彩蛋。是仿照chrome浏览器断网时的小恐龙游戏制作的动画游戏。闯关跑酷游戏的主角替换成了两名组员的头像，而障碍物的图片为另外两名组员的头像，通过不断点击键盘的空格键控制主角跳跃完成跑酷。配文：点击Space主力开发！。主页的尾部，也是滚动的地面地址。

在子文件中，分别在新的网页实现小游戏，具体设计在第四部分介绍。

飞机大战通过键盘左右键左右移动，通过空格键射击，将全部敌机击落进入下一关。

飞翔的小鸟通过单击鼠标，保持小鸟的高度，通过管道障碍物。

吃豆人通过键盘方向键控制，躲避小怪，吃完所有豆为通关。吃到紫色的豆，在一定时间内可以反过来吃怪。

Youngyoung游戏就是近期最热的三消游戏，单击图片计入卡槽，集齐三个图片相同的卡片在卡曹中即可删除。

1. 项目详细设计

***（一）飞机大战***

**4.1 软件详细设计流程说明**

（1）初始化阶段

当页面加载完成（window.onload），首先获取页面上的“开始”按钮（假设有一个id为"start"的元素）。为该按钮添加点击事件监听器，点击后隐藏按钮，并初始化游戏（Game.init("panel")）。

（2）游戏初始化

在Game.init函数中，首先获取游戏主面板（假设有一个id为"panel"的元素）。创建计分板（createScore），并将其添加到游戏主面板上。

创建第一关的敌人（createEnemy(0)），这里使用关卡数据中的eMap来定义敌人的种类和布局。创建飞机（createPlane），并绑定飞机的事件监听器（移动和发射子弹）。

（3）敌人创建

在createEnemy函数中，根据当前关卡信息，循环创建小蜜蜂（敌人）元素，并设置其类型、血量、速度和得分。将创建的小蜜蜂元素添加到一个<ul>列表中，并设置列表的宽度和位置。为小蜜蜂列表添加定时器，实现小蜜蜂的左右移动和换行逻辑。

（4）飞机移动和发射子弹

在bindPlaneEvent函数中，为文档添加键盘按下和松开的事件监听器。当按下方向键时，根据按键方向更新飞机的位置。当按下空格键时，发射子弹（createBullet）。子弹发射后，会添加一个定时器来更新子弹的位置，直到子弹离开屏幕或击中敌人。

（5）敌人移动和单兵作战

在runEnemy函数中，通过定时器实现敌人的左右移动和换行逻辑。

额外添加一个定时器来执行oneMove函数，实现单兵作战的逻辑，即随机选择一个小蜜蜂朝飞机移动。如果小蜜蜂与飞机发生碰撞（hasPengZhuang函数检测），则游戏结束，重新加载页面。

（6）碰撞检测

hasPengZhuang函数用于检测两个对象（如子弹和小蜜蜂）是否发生碰撞。通过比较两个对象的边界位置来判断是否发生碰撞。

（7）关卡切换

当当前关卡的所有敌人都被消灭后，通过createEnemy(1)切换到下一关。这里的关卡切换逻辑是硬编码的，仅作示例，实际开发中可能需要更复杂的逻辑来处理关卡切换和进度保存。

**4.2 主要模块设计**

飞机大战项目编码部分主要是由一个HTML文件与一个JS文件构成，HTML文件主要是搭建游戏页面框架，同时设定各个元素的样式，保证游戏页面的完备与美观。JS文件则定义了该网页的游戏模式与操作设定。其中包括游戏的启动与结束、游戏关卡中蜜蜂的基本属性与刷新频率、蜜蜂的移动与积分统计、还有飞机的左右移动与子弹的发射攻击。

**4.3 数据模块**

数据方面，该游戏模块使用了四张.png的图片，分别对应游戏中的飞机以及三种不同类型的蜜蜂，还加入了一段音频作为游戏进行中的背景音乐。

***（二）吃豆人***

实现经典的吃豆人游戏，通过HTML5 Canvas绘制游戏界面，并使用JavaScript控制游戏逻辑。项目分为多个模块，分别处理不同的功能，以实现完整的游戏体验。

主要模块

1. Game.js

2. MovingDirection.js

3. Pacman.js

4. Enemy.js

5. TileMap.js

6. index.html

模块功能说明

1. Game.js

此文件是游戏的入口文件，主要负责初始化游戏环境，设置画布以及开始游戏循环。

初始化：创建画布和上下文，初始化TileMap、Pacman和Enemy对象。

游戏循环：通过setInterval调用gameLoop函数，不断更新游戏状态。

游戏状态更新：在游戏循环中调用TileMap、Pacman和Enemy的draw方法，更新游戏画面。

游戏结束检测：检测游戏是否结束或获胜，并播放相应的声音效果。

2. MovingDirection.js

此文件定义了四个移动方向的常量，用于在Pacman和Enemy模块中控制角色的移动方向。

常量定义：定义了四个方向（上、下、左、右），分别对应0到3的数值。

3. Pacman.js

此文件定义了Pacman类，负责处理吃豆人角色的所有行为。

构造函数：初始化Pacman的位置、速度、动画计时器和声音效果。

键盘事件处理：监听键盘事件，根据玩家输入更新移动方向。

绘制Pacman：根据当前状态绘制Pacman，包括移动和动画效果。

吃豆逻辑：处理吃豆人吃普通豆子和能量豆的逻辑，并触发相应的声音效果。

吃鬼逻辑：在能量豆激活时，处理吃豆人吃鬼的逻辑。

4. Enemy.js

此文件定义了Enemy类，负责处理鬼怪的所有行为。

构造函数：初始化鬼怪的位置、速度和方向。

绘制鬼怪：根据当前状态绘制鬼怪，包括在能量豆激活时的变色效果。

碰撞检测：检测鬼怪是否与吃豆人碰撞。

移动逻辑：随机选择方向移动，并检测是否碰撞墙壁。

随机方向：随机选择鬼怪运动方向。

5. TileMap.js

TileMap 类通过 HTML5 Canvas 技术绘制游戏地图，并管理游戏元素的布局、碰撞检测及交互逻辑。

加载和绘制游戏地图，包括墙壁、豆子、吃豆人、敌人等游戏元素。

处理游戏元素的碰撞检测，确保吃豆人和敌人不会穿过墙壁或碰到其他障碍物。

提供方法获取吃豆人和敌人的初始位置，并在游戏中动态更新它们的位置和状态。

***（三）飞翔的小鸟***

FlappyBird 小游戏是一个基于 HTML5 Canvas 和 JavaScript 实现的简单飞行游戏。本报告将详细讨论项目的技术实现、游戏元素及其功能，以及展示效果。

1. 样式和布局

使用全局样式设置了页面和 HTML 元素的基本样式，包括重置边距和溢出隐藏，确保游戏画面在不同设备上的一致性和最佳显示效果。在 <canvas> 元素中绘制了游戏场景，使用 JavaScript 控制游戏的逻辑和渲染。在 Canvas 元素下方放置了一个信息层，用于显示游戏信息和标题。

1. Background.js

使用game.ctx.drawImage() 方法绘制背景图片，实现了一种无缝滚动的效果。通过改变 this.speed变量的值，循环绘制背景图片，使得背景看起来是连续滚动的。

使用game.ctx.fillStyle和 game.ctx.fillRect()方法绘制了两个不同颜色的矩形，分别表示天空和地面部分的背景。

update方法用于更新背景位置，实现背景的滚动效果。

1. Bird.js

鸟类 Bird 负责管理游戏中小鸟的状态、动画和碰撞检测。Bird 类在构造函数中初始化了小鸟的图片、位置和状态。render 方法用于绘制小鸟在画布上的状态。update 方法用于更新小鸟的位置、动画和状态。fly 方法用于触发小鸟向上飞行的动作。

1. Game.js

获取 canvas 元素，并设置其宽高适配当前设备的屏幕大小。使用 XMLHttpRequest 对象加载 resource.json 文件，该文件包含游戏所需的所有图片资源的信息。在 startGame 方法中，主要执行游戏的主循环逻辑

1. Land.js

Land 类在初始化时加载了名为 "Land2" 的图片资源，并设定初始速度 speed 为 0。render 方法负责将地面图片在 Canvas 上进行绘制。update 方法用于更新地面的状态和位置。

1. Pipe.js

Pipe 类用于管理和绘制游戏中的管道元素。Pipe 类在初始化时加载了名为 "Pipe1" 和 "Pipe2" 的图片资源，并设置了管道的初始高度、空隙大小和速度等属性。render 方法负责在 Canvas 上绘制上下两个管道，以及定义的管道间隙。update 方法用于更新管道的状态和位置。

1. SceneManager.js

使用function函数实现场景的变换和道具的移动。

***（四）young了个young***

1. 基础设置：

- 设置了页面的字符集为UTF-8。

- 设置了视口属性，确保在不同设备上正确缩放页面。

- 标题 "Young Young 无敌版"。

- 引入了外部CSS文件 "style.css"，用于定义游戏的样式。

- 引入了外部js文件 "game.css"，用于定义游戏的逻辑。

2. 游戏界面：

- 包含了一个 div 元素，id 为 "game-container"，用于放置游戏板和卡片槽。

- #game-board 用于放置游戏卡片。

- #card-slot 用于展示匹配的卡片。

- button用来控制音乐播放。

CSS代码定义了游戏的整体样式，使得游戏界面具有良好的视觉效果。

1. body: 设置了页面的背景图像、居中对齐、无边距和填充，确保背景图像铺满整个容器并居中。

2. button: 定义了按钮的样式，包括宽度、高度、颜色、字体大小、边框和圆角。

3. #game-container: 游戏容器的样式，包括相对定位、固定宽度和高度、背景图像以及居中对齐。

4. #game-board: 游戏板的样式，绝对定位在容器顶部，全宽，高度占容器的80%，并设置了卡片的对齐方式。

5. #card-slot: 卡片槽的样式，绝对定位在容器底部，设置了背景图像、高度、宽度和圆角，并将卡片对齐到左侧。

6. .card: 卡片的样式，包括宽度、高度、背景图像的大小和重复性，以及卡片之间的间距。

7. .popup: 弹出窗口的样式，绝对定位在屏幕中心，设置了背景色、边框、内边距、z-index（确保在其他元素之上）和文本居中。

8. .popup img: 弹出窗口中图片的样式，限制最大宽度以适应窗口大小，保持纵横比。

9. .popup button: 弹出窗口中关闭按钮的样式，设置为块级元素，居中对齐。

用js设置游戏的逻辑，使其能够运行

1. Card 类：定义了卡片对象，包括图像、名称、DOM元素、点击事件处理和位置设置。

- constructor 初始化卡片，设置背景图像，添加点击事件监听器。

- setLocation 设置卡片的位置坐标。

- onClick 当卡片被点击时，将其添加到选中的卡片列表中，并移到卡槽位置。

2. 数组 images 和 cardNames 分别存储卡片的图像路径和名称。cardPool, selectedCards 数组分别用来存储所有卡片和当前选中的卡片。

3. myAudio 和 playing 用于音乐播放控制。playPause 函数用于切换音乐的播放和暂停状态。

4. initGame 函数：首先，它使用 shuffleArray 函数（洗牌函数，用于随机打乱数组中的元素顺序）对 cardNames 进行洗牌，然后创建 Card 对象并存入 cardPool。接着，它调用不同区域的初始化函数来将卡片按指定布局添加到游戏板上。

5.不同区域的初始化函数

-initTowerArea 函数：创建一个塔形布局，卡片从左上角开始，按照倒三角形排列。

-initRightArea2, initLeftArea2, initRightArea1, initLeftArea1 函数：使用 addCardToPanel 函数将指定数量的卡片按水平或垂直排列到游戏板的不同区域。

6. addCardToPanel 函数：接收一组卡片、起始位置、X轴和Y轴的偏移量，然后将这些卡片按照这些参数依次添加到游戏板上。

7.检查匹配卡片和处理匹配成功后逻辑

-checkForMatches 函数：遍历已选择的卡片，统计每个名称的卡片数量。如果某个名称的卡片数量达到3个或更多，说明找到了匹配的组合，然后调用 removeMatchedCards 函数。

-removeMatchedCards 函数：根据名称过滤出匹配的卡片，从游戏板和 selectedCards 列表中移除它们。最后，检查游戏板上的卡片数量，如果为0，则调用 showImagePopup 显示胜利图像。

-showImagePopup 函数：创建一个弹出窗口，包含指定的图片和一个关闭按钮。点击关闭按钮时，弹出窗口从DOM中移除。

***（五）小恐龙***

本项目是一个简单的网页游戏，角色（由图片表示）必须跳过障碍物（树）以避免碰撞。游戏通过HTML、CSS和JavaScript实现。

主要模块

HTML结构：定义网页的结构，包括游戏容器和角色与障碍物的图片。

CSS样式：提供网页的样式，包括布局、动画和响应设计。

JavaScript功能：实现游戏逻辑，包括角色移动和碰撞检测。

设计流程

HTML结构：

HTML文档以标准的doctype声明开始，并包含字符编码和视口设置的meta标签。

在HTML文件中直接定义了一个<style>块以定义CSS。

页面主体包含一个内容区（<div class="content">），其中有一段文字提示用户按空格键来加速开发。

游戏容器（<div id="game">）包含两张图片：一个表示角色（< img class="role">）和一个表示树障碍物（< img class="tree">）。

CSS样式：

为所有元素应用基本的重置样式，以确保一致的边距、填充和盒模型。

将主体设置为flex容器，使内容在水平和垂直方向上居中，并使用背景图片。

游戏容器具有固定的宽度和高度、边框和相对定位，以包含绝对定位的子元素。

角色图片定位在游戏容器的底部，而树图片定位在右下方，并设置动画从右向左移动。

定义stop和jump动画的关键帧，以控制树的移动和角色的跳跃动作。

JavaScript功能：

添加事件监听器以检测按键事件，特别是空格键（键码32）。

定义jump函数，在按下空格键时为角色添加跳跃动画。使用定时器在跳跃完成后移除动画类。

每隔10毫秒检查一次碰撞检测，如果检测到角色与树发生碰撞，则停止动画并清除定时器以结束游戏。

结论

本项目展示了使用HTML、CSS和JavaScript实现的基本网页游戏。主要模块包括HTML结构、用于布局和动画的CSS样式，以及用于游戏逻辑和碰撞检测的JavaScript。设计流程突出展示了这些技术的集成，以创建互动的用户体验。

1. 项目实现

项目成功实现了需求中的基本内容，完成了主页的展示，跳转。小游戏的基本操作和控制。也仍存在一些可以继续改进优化的地方，如代模式的统一等等。

附录：

项目Github地址：

https://github.com/NKUstudent/-Project/tree/master

个人收获 ：

熟悉了网页各种前端技术的使用，深化了理解，同时也得到了不少乐趣。