**网络应用开发课程设计第三组——游戏开发**

**前言：**

本小组旨在开发一款集成多款解压小游戏的应用平台，命名为“解压盒子”。用户访问“解压盒子”网页后，首先会进入一个游戏选择界面。户的点击事件，系统能够引导用户进入所选择的游戏模块。在各游戏模块内部，用户可通过点击“更多游戏”按钮实现不同游戏间的无缝跳转。截至目前，我们已初步完成了每位小组成员各自设计的游戏，分别是2048-心情版小游戏、海绵宝宝矿工游戏、飞机大战苦瓜星人游戏以及开心小兔快通过监听用点跑游戏。

**组员信息：**

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 姓名 | 学号 | 开发游戏项目 |
| 1 | 罗晴 | 2022212527 | 2048-心情版小游戏、海绵宝宝矿工游戏 |
| 2 | 王安琪 | 2022212523 | 飞机大战苦瓜星人游戏 |
| 3 | 马梦婷 | 2022212493 | 开心小兔快点跑游戏 |

**一、2048-心情版小游戏规则及逻辑分析**

**1. 游戏规则**

1.1 基本操作

玩家需要使用键盘方向键（↑、↓、←、→）操作，将所有单元格中的词语向某一方向滑动。当两个相邻的词语等级相同时，它们会合并为更高级的词语，并为玩家增加相应的分数。滑动操作完成后，游戏会在随机的空单元格中生成一个较低等级的词语，继续填充网格空间。游戏提供实时的分数更新功能，分数会随着合并操作累计，而当前最高的词语也会同步显示在界面上，便于玩家跟踪游戏的进展。

1.2 胜负判定

游戏的胜利条件是当网格中的任意单元格出现最高等级的词语“至福”，玩家即获胜，游戏结束。如果所有网格均被填满且没有任何可合并的词语时，游戏也会结束，提示玩家失败。因此，玩家在游戏过程中不仅需要简单地滑动方块，还需要规划每一步操作，尽可能地为未来的合并创造条件，以延长游戏的进程并提升最终的分数。

1.3 词语等级

游戏中的情感词语分为11个等级，每个词语对应的等级和分值如下：

| **等级** | **词语** | **分值** |
| --- | --- | --- |
| 0 | 伤心 | 2 |
| 1 | 焦虑 | 4 |
| 2 | 疲惫 | 8 |
| 3 | 轻松 | 16 |
| 4 | 希望 | 32 |
| 5 | 快乐 | 64 |
| 6 | 幸福 | 128 |
| 7 | 极乐 | 256 |
| 8 | 激励 | 512 |
| 9 | 热爱 | 1024 |
| 10 | 至福 | 2048 |

1.4 游戏模式选择

为了增加游戏的趣味性和挑战性，游戏提供了三种难度模式供玩家选择：经典模式（Classic Mode）：采用4×4的网格，初始生成2个“伤心”词语，玩法与原始2048游戏相同。

**① 经典模式（Classic Mode）：**采用4×4的网格，初始生成2个“伤心”词语，玩法与原始2048游戏相同。

**② 简单模式（Easy Mode）：**采用5×5的网格，初始生成3个“伤心”词语，提供更多的空间来合并词语。

**③ 困难模式（Hard Mode）：**采用3×3的网格，初始生成一个“轻松”词语，减少了可用空间并提高了挑战性。

在游戏开始时，玩家可以通过选择不同的模式来调整游戏的难度，进而影响游戏的玩法和最终的目标。

**2. 技术实现**

游戏通过JavaScript实现了动态的网格尺寸和词语生成规则，根据不同难度模式（经典模式、简单模式、困难模式）调整网格的大小以及初始生成的词语数量。此外，游戏支持实时显示当前分数、最高词语以及游戏结束时的提示框，玩家可以选择重新开始或进入下一轮。

以经典模式为例：在技术实现方面，游戏使用HTML构建了4×4的网格，配合CSS实现了不同情感词语对应的动态背景颜色。玩家的每一步操作通过JavaScript实现，滑动的逻辑包括过滤空单元格、排列词语和执行合并操作，而游戏界面则实时更新。游戏的状态由一个4×4的二维数组表示，每个单元格对应一个数组元素，数组的变化直接驱动界面的渲染。为了增强游戏的可玩性和逻辑严谨性，胜负的判定逻辑包括两部分：一是判断是否存在空单元格，二是判断是否存在相邻可合并的词语。这两者均为否时，游戏才会显示结束提示。

**3.游戏体验优化**

为了提升玩家的交互体验，游戏在界面设计上加入了分数显示、当前最高词语显示和游戏结束弹窗等功能。当玩家成功或失败时，系统会通过弹窗形式明确提示，并提供“重新开始”的按钮，帮助玩家快速进入新一局游戏。此外，游戏的背景采用了柔和的色调设计，不同等级的词语也使用了渐变的颜色表示，进一步增强了视觉上的层次感。

模式选择功能的加入使得游戏的挑战性和可玩性得到了进一步的提升，玩家可以根据自己的兴趣和游戏目标来选择最合适的模式，享受更加丰富的游戏体验。 2048-心情版在经典玩法的基础上，增加了创意与情感的元素，不仅提高了趣味性，还为玩家提供了一种通过简单游戏释放压力的方式。

**二、海绵宝宝矿工游戏规则及逻辑分析**

**1. 游戏规则**

1.1 基本操作

玩家在游戏中控制一只钩子，目标是利用钩子抓取屏幕上不同的物品（如：北邮校徽、小蜗、痞老板等）。玩家可以通过点击鼠标或使用键盘来控制钩子的放出与收回。当钩子接触到物品时，物品会被钩起并增加相应的分数。每个物品都有不同的分值，玩家应尽量抓取高分物品（北邮校徽），以获得更多的分数。

游戏有时间限制（初始为60秒），当时间结束时，游戏会自动结束并显示得分。如果玩家在时间内达到目标分数，则进入下一关卡，关卡难度逐渐增加。玩家需要根据时间和分数来制定策略，决定是否继续抓取物品或者加速钩子的回收。

1.2 胜负判定

* **胜利条件：**游戏胜利的条件是，当玩家在规定时间内达到了目标分数时，游戏自动进入下一关。每个关卡的目标分数会根据关卡级别逐渐增加。
* **失败条件：**如果玩家未能在规定时间内达到目标分数，游戏将显示失败提示，并允许玩家重新开始当前关卡。

1.3 物品及其分值

在游戏中，不同类型的物品（如北邮校徽、小蜗、痞老板等）被分配了不同的分值。玩家通过钩子抓取这些物品来获得相应的分数。物品类型及其分值如下：

| **物品类型** | **分值** | **说明** |
| --- | --- | --- |
| 小蜗 | 50-250 | 小蜗分为三种不同大小，分值从50到250不等 |
| 痞老板 | 11-30 | 痞老板有三种不同大小，分值较低，用来增加游戏难度 |
| 北邮校徽 | 600 | 北邮校徽为高分物品，分值固定为600 |

物品会随着关卡的推进变得更加多样化和具有挑战性，玩家需要快速决策以获取更高分数。

**2. 技术实现**

游戏开始时，玩家首先看到一个启动页面，点击“开始游戏”按钮后进入游戏画面。在游戏中，玩家通过控制钩子抓取不同类型的物品，每个物品的分值不同，抓取后会增加相应的分数。物品抓取后会被移除，并在屏幕上重新生成，同时时间不断流逝。游戏结束时，系统会根据玩家的得分判断是否达到了目标分数，若达标则进入下一关，否则显示失败提示。

在技术实现方面，游戏采用 HTML5 和 JavaScript 创建了一个基于画布（<canvas>）的 2D 游戏。游戏通过 JavaScript 实现了以下功能：

**① 物品生成与更新：**每种物品在每轮游戏中以随机位置生成。物品的类型、大小、速度和分值会随着游戏进程发生变化，增加了游戏的策略性。

**② 钩子的控制：**玩家通过鼠标点击或键盘操作来控制钩子的发射与回收，钩子的行为受物品距离和钩子本身的物理属性影响。钩子可以延伸并抓取不同大小的物品。

**③ 时间与分数管理：**游戏有倒计时功能，时间结束时，自动判断是否达到了目标分数。如果得分达到目标，玩家将进入下一关卡，否则游戏结束。

**④ 关卡与目标分数：**每一关的目标分数会根据关卡级别增加，玩家需要不断提升自己的得分能力以应对越来越高的目标。

**3. 游戏体验优化**

为了提升玩家的游戏体验，游戏在多个方面进行了优化：

**① 界面设计：**游戏的启动页面和结束页面都具有良好的视觉效果，使用了半透明背景和柔和的色调，使得整体风格更加友好和富有趣味性。

**② 物品的视觉表现：**不同类型的物品使用了不同的图像和动画效果，增强了视觉的丰富性和层次感。每种物品的大小、速度和分值不同，给玩家带来了不同的游戏挑战。

**③ 动态分数与关卡进度：**游戏实时显示玩家的分数、目标分数和当前关卡，确保玩家能够随时掌握游戏进度，并根据情况调整策略。

**④ 游戏音效与反馈**：游戏的音效和视觉反馈增强了操作的反馈感，当玩家成功抓取物品时，会有音效提示，并显示分数变化，提升了游戏的互动性。

**⑤ 时间管理：**游戏的时间管理系统使得玩家必须在有限的时间内抓取尽可能多的高分物品，增加了游戏的紧张感和挑战性。

**三、飞机大战苦瓜星人游戏规则及逻辑分析**

**1.游戏规则**

1.1基本逻辑

在这款“飞机大战苦瓜星人”的小游戏中，玩家的目标是用鼠标控制我方飞机发射子弹尽可能多地击落敌方飞机以积累分数，同时需要躲避敌方苦瓜星人的攻击，若我方飞机与敌方相撞则游戏结束。

1.2具体操作及胜负界定

游戏开始时，玩家点击“开始游戏”按钮，随后使用鼠标来移动我方飞机。游戏过程中，玩家可以通过点击鼠标来暂停游戏。游戏界面提供了实时的分数显示，帮助玩家跟踪当前的表现。积分积累每五万积分，游戏结束的鼓励语会有所不同，当积分达到二十五万时，玩家将成功战胜苦瓜星人。

1.3 不同飞机分值标准及积分鼓励语

| **物品类型** | **分值** | **说明** |
| --- | --- | --- |
| 小苦瓜脸 | 1000 | 被击毁得到的分数是 1000 分。 |
| 苦瓜兔子 | 5000 | 被击毁得到的分数是 5000 分。 |
| 苦瓜黄鸭 | 30000 | 被击毁得到的分数是 30000 分。 |

| **积分范围** | **鼓励语** |
| --- | --- |
| (−∞,50000] | "不错哟，你已经成功避开了一些苦瓜星人的攻击，继续保持！" |
| (50000,100000] | "干得漂亮！你击退了不少苦瓜星人，看来你已经掌握了一些技巧" |
| (100000,150000] | "太棒了！你已经击败了许多敌人，你的实力不容小觑。" |
| (150000,200000] | "哇，你的表现令人印象深刻！你几乎要成为天空的主宰了。" |
| (200000,250000] | "不可思议！你已经击败了海量的苦瓜星人，你简直就是飞行大师。" |
| (250000,∞) | "恭喜你，天空已经没有敌人是你的对手了！你是真正的天空之王，坏心情一扫而空！" |

**2.技术实现**

“飞机大战苦瓜星人”游戏的技术实现主要包括HTML、CSS和JavaScript的综合应用。HTML文件（index.html）定义了游戏的结构，包括开始按钮、分数显示、暂停菜单和游戏结束界面等元素。CSS文件（main.css）负责游戏的视觉表现，通过背景图片、元素定位和样式设置来美化界面，确保游戏元素在页面上的正确显示和布局。

在main.js文件中，JavaScript代码负责实现“飞机大战苦瓜星人”游戏的核心逻辑和动态行为。首先，代码通过document.getElementById方法获取HTML中的各个界面元素，如主界面、开始界面、分数显示界面、暂停界面和游戏结束界面。这些元素的引用用于后续的游戏状态控制和界面更新。

游戏定义了一个plan类来创建飞机对象。这个类通过构造函数初始化飞机的位置、生命值、分数、大小、速度等属性，并通过原型方法定义了飞机的移动和初始化行为。类似地，bullet类用于创建子弹对象，包含子弹的位置、攻击力、大小等属性，以及移动行为的方法。敌机类enemy继承自飞机类plan，通过调用父类的构造函数并添加特定属性来创建敌机对象。本方飞机类ourplan也继承自飞机类plan，并添加了特定的属性和行为，如本方飞机的图片和爆炸效果。

事件监听器在游戏的交互中起着关键作用。代码通过addEventListener方法为游戏元素添加事件监听器，以响应玩家的鼠标操作。例如，mousemove事件用于控制本方飞机的移动，而click事件用于暂停游戏。这些事件监听器确保了游戏的互动性和响应性。

游戏的主循环由start函数实现，负责更新游戏状态，包括敌机和子弹的移动、碰撞检测、分数更新等。碰撞检测通过计算飞机和子弹的位置和尺寸来实现。如果检测到碰撞，游戏会更新相应的对象状态，如敌机的生命值和本方飞机的爆炸效果以及播放音效。

分数更新和游戏结束的处理也是游戏逻辑的重要组成部分。游戏通过更新DOM元素的内容来显示实时分数。当本方飞机被击中时，游戏结束，并显示最终分数和鼓励信息。此外，游戏还定义了begin、zanting（暂停）、chongxinkaishi（重新开始）和jixu（继续）等函数来控制游戏的开始、暂停和结束。

**四、开心小兔快点跑游戏规则及逻辑分析**

**1. 游戏规则**

1.1 基本操作

玩家通过键盘方向键（↑、↓、←、→）来控制游戏角色。 通过“↑、↓”方向键控制跳跃和下蹲，通过“←、→”方向键移动到路两边。成功执行特定动作，如在障碍物出现时及时跳跃或在需要时下蹲，可以帮助玩家收集道具和宝石，增加得分。

1.2 道具种类及效果

**① 宝石：**收集宝石可以增加玩家的宝石总数，有助于进行更多的升级或获得更高的得分。

**② 胡萝卜：**胡萝卜道具可使玩家的角色进入“无敌”状态，在无敌状态下，玩家的角色不会受到障碍物的阻挡，可以顺利通过。

**③ 磁吸：**磁吸道具使玩家的角色能够自动吸引附近的宝石，而不需要进行额外操作。

**④ 加速：**加速道具使玩家的角色移动速度加快，从而可以更快地通过障碍物和收集宝石。

**⑤ 跳跃：**跳跃道具使玩家的角色能够进行更高或更远的跳跃，玩家可以轻松越过障碍物，到达更高的平台或收集到更多的宝石。

1.3游戏升级规则

游戏会记录玩家的游戏进度，以及所收集的宝石数量，难度等数据。当玩家有足够的宝石时便可以升级，每次升级等级增加，并且消耗一定数量的宝石。初始等级设置为0，最高等级为10。升级所需的宝石消耗如下：

| **等级** | **宝石消耗** |
| --- | --- |
| 1 | 20 |
| 2 | 20 |
| 3 | 50 |
| 4 | 50 |
| 5 | 80 |
| 6 | 80 |
| 7 | 120 |
| 8 | 120 |
| 9 | 175 |
| 10 | 175 |

可以升级的选项有四个，每个升级选项及其功能介绍如下:

* **开始速度：**设置游戏的初始速度是400像素/秒 ，每次升级速度会增加25像素/秒。
* **速度提升：**玩家可以通过收集宝石和完成特定动作来获得速度提升。每次提升，速度增加25像素/秒。速度提升还受到随机因素的影响，从而增加游戏的挑战性和不可预测性。
* **更多道具：**升级后可增加游戏中的道具种类、数量和道具额外的能力，如提高道具的使用时间等，以提高玩家在游戏中生存的能力，取得更好的成绩。
* **更多宝石：**升级后，宝石的分布和数量也会相应增加。这是通过增加单位时间内生成的宝石数量或者增加玩家角色的收集范围来实现的。

**2.技术实现**

游戏的整体架构设计从资源加载到用户输入处理，再到游戏状态管理和渲染流程，确保了各个组件之间的紧密协作，从而保证游戏流畅运行并提供优质的用户体验。本项目使用HTML5 Canvas进行界面绘制，通过ctx对象执行图形渲染。游戏中的界面元素（如按钮、得分板）由Elements对象管理，而音效和背景音乐依赖Howler.js库来实现播放、暂停等操作。复杂的动画效果，例如角色动作和界面过渡，则是通过TweenMax库完成。

**① 初始化阶段：**游戏启动时，首先进行一些必要的初始化工作,包括设置游戏状态、加载资源、初始化用户输入监听器等。游戏的基本状态，例如游戏是否处于加载中、暂停或运行状态等，由通过gameState变量来管理并针对每种状态调用相应的函数来进行初始化或更新。此外，需要初始化用户输入监听器，这些监听器用于检测用户的鼠标点击和键盘按键事件，并将相应的事件传递给游戏逻辑处理。

**② 游戏主循环：**主循环的核心在于 updateGameEvent() 函数，它负责持续更新游戏状态和渲染 。具体来说，主循环的任务包括根据用户的输入指令更新游戏逻辑，如角色移动、碰撞检测、应用物理规则等。在每次更新游戏状态时，同步更新游戏的得分和其他动态信息，并使用Canvas API将当前帧的画面绘制到屏幕上。一次循环完成后通过请求下一帧更新，形成反复循环。

**③ 游戏结束处理：**通过判断小兔是否掉下或撞到障碍物来监听游戏结束信号。当满足结束条件时，触发游戏结束事件。游戏结束事件的处理一系列操作来完成，例如，停止背景音乐的播放；清除游戏中的所有动态信息；将最终得分保存到本地存储中；展示游戏结束的画面。

