弹射球游戏开发日志

dev log

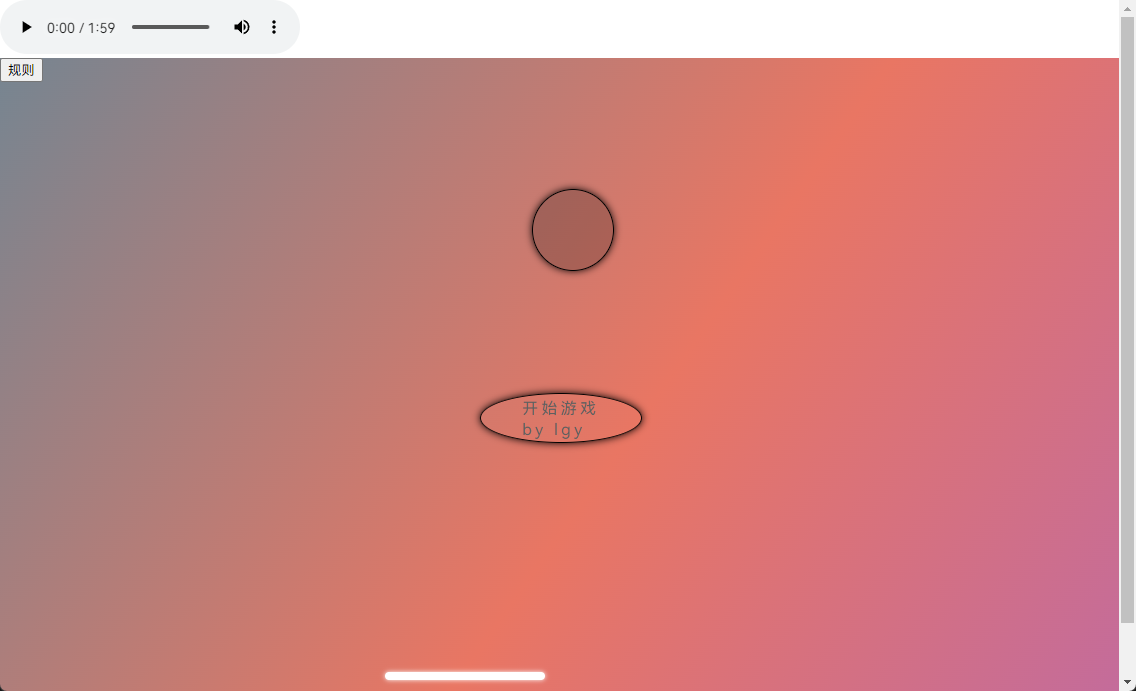
21332228 林国烨

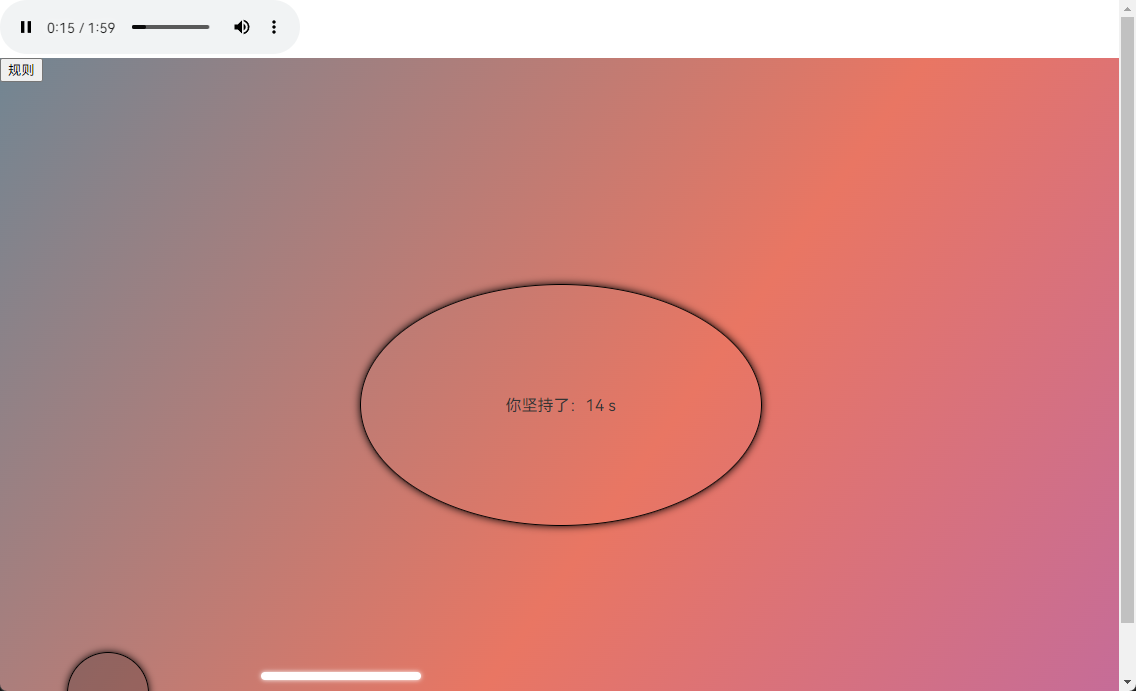
2023

# 游戏简介和规则

这是一款通过用挡板接住弹射的球来得分的小游戏，游戏的主要目的是接住球来赢取尽量高的分数。键盘操作为： 左右键控制挡板，。球会以高速在游戏区域内弹射，玩家需要在球落地前接住球，每接住一次得10分，球每落地一次扣30分。游戏左上角会显示当前分数和已持续时间。如果分数低于0则判定游戏失败。

# 游戏截图





# 开发日志 Dev Log

7月6日、7月7日：

构思：

设计一个类似于pinball的弹珠台游戏程序，键盘操作为： "A"和"D"键控制左右两个挡板，游戏的主要目的是发射球，然后通过命中区域内随机生成的目标来赢取尽量多的分数。场景内同时只会有一个目标存在。球击中目标5次即可将目标消灭，每通过直接碰撞消灭一个目标得1分。目标上会随机生成弱点，球击中弱点则可直接将目标消灭，同时每通过击中弱点消灭目标得5分。如果球落下则判定游戏失败。

7月8日：

问题：

假设你是一个资深程序员，帮我用html、css和JavaScript语言设计一个类似于3D pinball的弹珠台游戏程序，游戏中球会模拟重力下落，键盘操作为： 空格键控制左右两个弹射挡板，游戏的主要目的是发射球，然后通过命中区域内随机生成的目标来赢取尽量多的分数。场景内同时只会有一个目标存在。目标为红色长方形，球击中目标5次即可将目标消灭，每通过直接碰撞消灭一个目标得1分。目标边缘上会随机生成弱点，弱点显示为一个绿色的小三角形。球击中弱点则可直接将目标消灭，同时每通过击中弱点消灭目标得5分。如果球落到两块挡板之间则判定游戏失败。游戏要有“开始界面”，可让玩家选择开始游戏。游戏开始后页面左上角显示时间，右上角显示已得分数，游戏结束后显示“game over”字样。

回答略，但游戏点击“开始游戏”按钮后即无法运行，且gpt生成的游戏与想象中的3D pinball相去甚远，且挡板无法触发弹射，排查原因中，样式也不美观。

7月9日：

排查了一天，问gpt3.5问了很久，还是无法排查出原因，因基础框架中的bug无法解决，遂全部推到重来。

新的构思：

简化设计，取消弱点机制

设计一个弹珠台游戏程序，键盘操作为： 左右键控制挡板，游戏的主要目的是接住球来赢取尽量多的分数。球会以高速在游戏区域内弹射，玩家需要在球落地前接住球，每接住一次得10分，球每落地一次扣30分。游戏左上角会显示当前分数和已持续时间。如果分数低于0则判定游戏失败。游戏失败后会显示玩家在游戏中坚持了多少秒。

问题：

假设你是一个资深程序员，请你帮我设计一个弹珠台游戏程序，键盘操作为： 左右键控制挡板，游戏的主要目的是接住球来赢取尽量多的分数。球会以高速在游戏区域内弹射，玩家需要在球落地前接住球，每接住一次得10分，球每落地一次扣30分。游戏左上角会显示当前分数和已持续时间。如果分数低于0则判定游戏失败。

回答：略，gpt4.0给出了整个游戏的基础框架，但只是雏形。而且gpt生成的内容中游戏是靠键盘左右键操控挡板的，实际游玩中体验并不好。

问题：分析base.js，这是一个小游戏的js代码。如果我想将操控滑块的方式改为由鼠标操控，我该怎么改？

回答：要将滑块的控制方式改为鼠标控制，你需要添加一个鼠标移动事件监听器来更新滑块的位置。在 base.js 文件中，找到滑块的定义部分，然后添加以下代码：

*// 鼠标移动事件监听器*

document.addEventListener('mousemove', function(*e*) {

    this.baffle.style.left = e.clientX - this.baffle.offsetWidth / 2 + 'px';

}.bind(this));

这段代码会在鼠标移动时更新滑块的位置，使其始终位于鼠标的下方。e.clientX 是鼠标的水平位置，this.baffle.offsetWidth / 2 是滑块宽度的一半，这样可以使滑块的中心与鼠标对齐。  
  
注意，我们使用了 .bind(this) 来确保事件处理函数中的 this 指向游戏对象，而不是 document。  
  
这样，你就可以通过移动鼠标来控制滑块了。

问题：我觉得背景太单调了，我该如何修改呢？

回答：要修改游戏的背景颜色，你可以修改 base.css 文件中的 #app 的样式。在 background-image 属性中，你可以更改渐变的颜色值来改变背景颜色。以下是一个示例：

#app {

  background-image: linear-gradient(125deg, #FF0000, #00FF00);

}

这个示例将背景颜色从红色渐变到绿色。你可以根据需要使用任何有效的 CSS 颜色值来定义渐变的颜色。

最终查询网上资料后，我设计了一个渐变色的背景，并在css代码中将它放大后循环移动，实现了在游戏中渐变背景的效果。

问题：分析base.js，这是一个小游戏的js代码。如果我想将操控滑块的方式改为由鼠标操控，我该怎么改？

回答：要将滑块的控制方式改为鼠标控制，你需要添加一个鼠标移动事件监听器来更新滑块的位置。在 base.js 文件中，找到滑块的定义部分，然后添加以下代码：

*// 鼠标移动事件监听器*

document.addEventListener('mousemove', function(*e*) {

    this.baffle.style.left = e.clientX - this.baffle.offsetWidth / 2 + 'px';

}.bind(this));

这段代码会在鼠标移动时更新滑块的位置，使其始终位于鼠标的下方。e.clientX 是鼠标的水平位置，this.baffle.offsetWidth / 2 是滑块宽度的一半，这样可以使滑块的中心与鼠标对齐。  
  
注意，我们使用了 .bind(this) 来确保事件处理函数中的 this 指向游戏对象，而不是 document。  
  
这样，你就可以通过移动鼠标来控制滑块了。

问题：分析base.js，将其中计时的部分改成以秒为单位

回答：在 base.js 文件中，计时部分在 this.timing 函数中。原来的代码是以毫秒为单位的，我们需要将其改为以秒为单位。以下是修改后的代码：

*// 计时*

this.timing = function() {

    var d = new Date();

    return Math.floor((d.getTime() - this.actionTime) / 1000);

};

这样，计时就会以秒为单位显示了。

问题：能帮我优化游戏中球的运动轨迹吗？

回答：好的，这是优化过的代码：*// 球的运动*

    this.ballMove = function() {

        this.timer = setInterval(() => {

            this.boundaryJudge();

            this.setBallPosition();

            this.infoTime.innerText = this.timing();

        }, 15);

    };

问题：如何移动一个背景动画

回答：@keyframes 是 CSS 中用于定义动画的关键帧的规则。它允许您指定在动画的不同时间点上元素应该具有的样式。  
  
@keyframes bgMove 定义了一个名为 bgMove 的动画，它控制了背景图像的位置，从而实现了背景颜色的变化效果。  
  
以下是 @keyframes bgMove 的代码示例：

@keyframes *bgMove*{

  0%{

      background-position: 0 50%;

  }

  50%{

      background-position: 100% 50%;

  }

  100%{

      background-position: 0 50%;

  }

}

这段代码定义了三个关键帧：0%、50% 和 100%。在每个关键帧中，通过修改 background-position 属性的值来改变背景图像的位置，从而实现背景颜色的变化效果。  
  
关键帧的百分比值表示动画的进度，0% 表示动画的开始，100% 表示动画的结束。在这个例子中，背景图像的位置在动画的开始和结束时都是 (0, 50%)，在动画的中间时是 (100%, 50%)。  
  
要修改背景颜色的变化效果，您可以修改 @keyframes bgMove 中的关键帧的样式，例如修改 background-position 的值或添加其他样式属性。  
  
请注意，@keyframes 规则只是定义了动画的关键帧，要应用动画效果，还需要将该动画应用到元素上，例如通过 animation 属性。

问题：如何使用@keyframe实现不同颜色渐变的背景？

回答：根据你所提供的代码，以下是示例：

#app{

  background-image: linear-gradient(125deg,#D32F2F,#C2185B,#7B1FA2,#512DA8,#303F9F);

  background-size: 400%;

  animation: bgMove 6s infinite;

}

@keyframes *bgMove*{

  0%{

      background-position: 0 50%;

  }

  50%{

      background-position: 100% 50%;

  }

  100%{

      background-position: 0 50%;

  }

}

background-image 属性定义了一个线性渐变，颜色从 #D32F2F 到 #C2185B，再到 #7B1FA2，然后是 #512DA8，最后是 #303F9F。  
  
animation 属性定义了一个名为 bgMove 的动画，该动画的持续时间为 6 秒，并且会无限次重复。  
  
@keyframes bgMove 定义了 bgMove 动画的关键帧。这个动画改变的是背景图像的位置，使得背景颜色看起来在不断变化。

问题：能否设置为一点击开始游戏就立即播放背景音乐？

回答：Yes, you can set the background music to play immediately when the game starts by using JavaScript. Here's an example of how you can achieve this:  
  
1. Add an id attribute to the <audio> element in your HTML file:

<audio id="backgroundMusic" src="your\_music.mp3" loop></audio>

2. Add the following JavaScript code to your HTML file, preferably just before the closing </body> tag:

<script>

    document.getElementById("actionBtn").addEventListener("click", function() {

        var backgroundMusic = document.getElementById("backgroundMusic");

        backgroundMusic.play();

    });

</script>

In this example, we added an event listener to the "开始游戏" button (assuming it has the id "actionBtn"). When the button is clicked, the JavaScript code will play the background music by calling the play() method on the <audio> element with the id "backgroundMusic".  
  
Make sure to replace "your\_music.mp3" with the actual filename and extension of your music file.  
  
With this code, the background music will start playing immediately when the "开始游戏" button is clicked.

问题：能否给我设计的游戏在开始前加上额外的“规则“按钮，使得玩家按下”规则按钮“后就跳出说明栏告诉玩家游戏玩法？

回答：当然可以。你可以按照以下步骤来实现：  
  
1. 在HTML文件中添加"规则"按钮。将按钮放在<div id="app">元素内，<div id="gameInformation">元素之前。例如：

<button id="rulesBtn">规则</button>

2. 在现有的<div id="result">元素下面添加一个新的<div>元素。这个新的<div>元素将用作显示游戏说明的容器。给它一个id，比如"rulesContainer"。例如：

<div id="result"></div>

<div id="rulesContainer"></div>

3. 添加JavaScript代码来处理"规则"按钮的点击事件，并显示游戏说明。将此代码放在<script>标签内，最好放在关闭</body>标签之前。例如：

<script>

    document.getElementById("rulesBtn").addEventListener("click", function() {

        var rulesContainer = document.getElementById("rulesContainer");

        rulesContainer.innerHTML = "游戏玩法：这里是游戏的规则说明。";

    });

</script>

在这个例子中，我们为"规则"按钮（假设它的id是"rulesBtn"）添加了一个事件监听器。当按钮被点击时，JavaScript代码将设置"rulesContainer"元素的innerHTML来显示游戏说明。你可以将占位符文本"游戏玩法：这里是游戏的规则说明。"替换为你的实际游戏说明。  
  
通过这些更改，你的游戏将有一个额外的"规则"按钮，当点击时，将在指定的容器中显示游戏说明。

问题：能为球的每一次碰撞添加音效吗？

回答：当然可以。你可以通过在碰撞检测函数中添加音效播放的代码来实现。首先，你需要在HTML文件中添加一个<audio>元素来存放音效文件：

<audio id="collisionSound" src="your\_sound.mp3"></audio>

然后，在JavaScript文件中的judgeCollision函数中添加音效播放的代码：

this.judgeCollision = function() {

    if (this.ball.offsetLeft + this.ball.offsetWidth > this.baffle.offsetLeft && this.ball.offsetLeft < this.baffle.offsetLeft + this.baffle.offsetWidth &&

        this.ball.offsetTop + this.ball.offsetHeight > this.baffle.offsetTop - 20 && this.ball.offsetTop + this.ball.offsetHeight < this.baffle.offsetTop && this.verticalSpeed > 0) {

        this.verticalSpeed = -this.verticalSpeed;

        this.addFraction();

*// 播放音效*

        document.getElementById('collisionSound').play();

    }

};

这样，每次球与滑块碰撞时，都会播放音效。请确保替换"your\_sound.mp3"为你的音效文件的实际文件名和扩展名。