

**网页游戏设计**

**姓名： 何鑫**

**学号：21332223**

**授课教师：刘合翔**

**完成时间：2023年7月10日**

目录

[一、游戏策划 2](#_Toc139839249)

[（一）游戏描述 2](#_Toc139839250)

[（二）界面设计 3](#_Toc139839251)

[（三）游戏设计 3](#_Toc139839252)

[二、游戏 5](#_Toc139839253)

[（一）游戏玩法 5](#_Toc139839254)

[（二）游戏截图 5](#_Toc139839255)

[三、游戏代码详细 6](#_Toc139839256)

[四、游戏开发日志 7](#_Toc139839257)

# 一、游戏策划

## （一）游戏描述

我想设计的游戏是有以下内容构成：

游戏左上角有耐久值5格，右上角有开始和暂停按钮，当开始时总会倒计时3秒。

有个聚宝盆跟着鼠标移动，然后玩家利用鼠标移动聚宝盆接住画面中每隔1秒随机直线四射的宝石，并且有30%的概率随机直线四射石头，当聚宝盆被石头砸中会扣除一格耐久值，当耐久值为0游戏结束。

宝石分为大中小宝石，且有红、蓝、绿三种颜色，聚宝盆是圆形，并且每隔5秒随机变化为红、蓝、绿三种颜色，当接中相同颜色加5分，接中不同颜色加1分，积分达到100、200时聚宝盆往外扩大1半径长度的范围。

聚宝盆初始大小为设定了半径的圆。

当玩家耐久度耗尽自行结束，则统计得分。

## （二）界面设计

1.界面右上角有开始和停止按钮。两个按钮为圆角5px矩形，边长为20px，灰色背景，白色字体，字体加粗，字体为18px。

2.界面左上角为5格耐久度，每格耐久度为心形，不描边，填充为橙色。

3.界面背景设置为图片，聚宝盆为贴图。

4.宝石不描边。

5.所有游戏内按钮在鼠标hover时显示阴影。

## （三）游戏设计

1.游戏类型：游戏类型为休闲。

2.游戏规则：玩家用鼠标操作聚宝盆，接住随机而来的宝石，躲避随机而来的石头，从而获得积分，完成通关。

3.游戏功能：

宝石发射模式和颜色变化规律不同。

聚宝盆初始大小为1cm半径的圆。当聚宝盆接触到一个宝石，则得分，接触到一个石头，则扣耐久度。

宝石随机直线或曲线穿过屏幕，而宝石分为小（半径0.2cm）、中（半径0.5cm）、大（半径1cm）的圆类型，又分为红、黄、蓝三个颜色填充。石头随机直线或曲线穿过屏幕，它的半径大小为1cm的圆，颜色为灰色。

小宝石（半径0.2cm）的可得分数为每个2分，中宝石（0.5cm）的可得分数为每个3分，大宝石（1cm）的可得分数为每个5分。

聚宝盆无法离开屏幕游戏范围，聚宝盆跟随鼠标移动，以鼠标为圆心绘制聚宝盆，且隐藏鼠标指针。

当积分达到100、200时，聚宝盆的半径增加1cm。每一关限定时间为15s。聚宝盆有5格耐久度，耐久度为0时游戏结束，统计得分。通关第100关后游戏通关，统计得分。

每关前10s宝石和石头速度为每秒300px，后5s的宝石和石头速度为每秒500px。

每一关结束后统计得分和耐久度，并统计最终得分结果。

耐久度在屏幕左上角现实，屏幕右上角有开始和停止按钮。每次开始时倒计时三秒，在屏幕中央显示，三秒后游戏开始进行。键盘的S键等同于开始按钮，键盘的P键等同于结束按钮。

4.游戏目标用户：大学生

# 二、游戏

## （一）游戏玩法

玩家看到窗口提示确认后点击开始游戏，将鼠标移到游戏区域，接住宝石加分，接到石头扣分。

游戏在接住宝石后会有音效，当耐久度降低会有不同提示。

游戏在耐久度降为0或者停止游戏后在开始，会有不同的速度模式，体验更刺激快速的接元宝！

## （二）游戏截图

山上的风景

中度可信度描述已自动生成

电脑游戏的截图

低可信度描述已自动生成

# 三、游戏代码详细

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>聚宝盆</title>

    <style>

       #game-area {

            position: relative;

            background-image: url('8472.jpg');

            margin: auto;

            width: 1000px;

            height: 800px;

            background-color: #f2f2f2;

*/\* 添加其他样式以设计游戏区域 \*/*

        }

        #start-button, #stop-button {

            position: relative;

            top: 50px;

            left: 10px;

            padding: 5px;

            border-radius: 5px;

            font-size: 28px;

            color: white;

            background-color: #77D9F0;

            box-shadow: 2px 2px 4px rgba(0, 0, 0, 0.5);

            border: none;

*/\* 添加其他样式以满足需求 \*/*

       }

       #pot {

            position: relative;

            width: 2cm;

            height: 2cm;

            border-radius: 50%;

        }

        .game-score{

            color:#77D9F0;

            font-weight: bold;

            font-size: 24px;

            margin: 10px;

        }

        .game-durability{

            margin: 10px;

            color:#77D9F0;

            font-weight: bold;

            font-size: 24px;

        }

        #rank{

            background-color: white;

            width: 150px;

            height: 100px;

            margin: 20px;

            position: fixed;

            top: 240px;

            border-radius: 5px;

        }

        body{

            background-image: url('body.jpg');

            background-size: cover;

        }

    </style>

</head>

<body>

    <button id="start-button">开始游戏</button>

    <button id="stop-button">停止游戏</button>

    <div id="game-area"></div>

    <div id="rank">

        <div class="game-score">得分：0</div>

        <div class="game-durability">耐久度：5</div>

    </div>

      <script>

        var gameScore = document.querySelector('.game-score')

        var gameDurability = document.querySelector('.game-durability')

        var createNum = 500;

        var sp = 20;

        var gameArea = document.getElementById("game-area");

        var startButton = document.getElementById("start-button");

        var stopButton = document.getElementById("stop-button");

        var score = 0;

        var durability = 5;

        var createGemOrStoneInterval = null;

        var timer;

        function **playSoundMusic**(){

                var sound = new Audio("");

                sound.play();

            }

        startButton.addEventListener("click", function() {

            playSoundMusic();

            timer = setTimeout(function() {

*// 运行游戏区域的函数*

                runGameArea();

            }, 3000);

            var countdownArea = document.createElement('div');

            countdownArea.style.zIndex = '100';

            countdownArea.classList.add('countdown');

            countdownArea.style.position = 'fixed';

            countdownArea.style.top = '50%';

            countdownArea.style.left = '50%';

            countdownArea.style.transform = 'translate(-50%, -50%)';

            countdownArea.style.fontSize = '24px';

            countdownArea.style.fontWeight = 'bold';

            countdownArea.style.textAlign = 'center';

            document.body.appendChild(countdownArea);

*// 倒计时3秒后运行游戏*

            var countdown = 3;

            var countdownInterval = setInterval(function() {

                countdown--;

                countdownArea.innerText = countdown;

                if (countdown === 0) {

                    clearInterval(countdownInterval);

                    document.body.removeChild(countdownArea);

                }

            }, 1000);

        });

        stopButton.addEventListener("click", function() {

            clearTimeout(timer);

*// 停止游戏并清除游戏区域*

            stopGameArea();

        });

        function **runGameArea**() {

*// 创建聚宝盆*

            var pot = document.createElement("div");

            pot.id = "pot";

            gameArea.appendChild(pot);

            var element = document.getElementById("pot");

            element.style.backgroundImage = "url('pot.png')";

            element.style.backgroundSize = "cover";

*// 限制聚宝盆在游戏区域内*

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

                pot.style.left = *e*.pageX - gameArea.offsetLeft  + 'px';

                pot.style.top = *e*.pageY - gameArea.offsetTop + 'px';

            });

*// 运行游戏区域的函数的代码*

            createGemOrStoneInterval = setInterval(createGemOrStone, createNum);

            function **createGemOrStone**() {

                var gemOrStone = document.createElement("div");

                var size, color, score;

*// 随机决定是创建宝石还是石头*

                if (Math.random() < 0.75) {

*// 创建宝石*

                    size = Math.random() < 0.33 ? 0.2 : (Math.random() < 0.5 ? 0.5 : 1);

                    color = ["red", "yellow", "blue"][Math.floor(Math.random() \* 3)];

                    gemOrStone.score = size === 0.2 ? 2 : (size === 0.5 ? 3 : 5);

                    gemOrStone.className = "gem";

                } else {

*// 创建石头*

                    size = 3;

                    color = "gray";

                    gemOrStone.className = "stone";

                }

*// 设置宝石或石头的样式*

                gemOrStone.style.width = size + "cm";

                gemOrStone.style.height = size + "cm";

                gemOrStone.style.backgroundColor = color;

                gemOrStone.style.position = "absolute";

                gemOrStone.style.borderRadius = "50%";

                gemOrStone.style.left = Math.random() \* (gameArea.offsetWidth - size \* 10) + "px";

                gemOrStone.style.top = "0px";

                gameLoop()

                moveGemOrStone(gemOrStone)

*// 将宝石或石头的元素添加到游戏区域*

                gameArea.appendChild(gemOrStone);

*// 返回创建的宝石或石头的信息*

                return {

                    element: gemOrStone,

                    score: score

                };

            }

            gameArea.style.cursor = 'none'; *// 隐藏鼠标指针*

            function **playSoundLose**() {

                var sound = new Audio("split.aac");

                sound.play();

            }

            function **playSoundF**(){

                var sound = new Audio("F.mp3");

                sound.play();

            }

            function **playSoundGem**(){

                var sound = new Audio("苍蝇.mp3");

                sound.play();

            }

*// 在游戏的主循环中检查碰撞*

            function **checkCollision**(*gemOrStone*) {

                var potRect = pot.getBoundingClientRect();

                var gemOrStoneRect = *gemOrStone*.getBoundingClientRect();

*// console.log(gemOrStoneRect)*

                if (potRect.left < gemOrStoneRect.right &&

                    potRect.right > gemOrStoneRect.left &&

                    potRect.top < gemOrStoneRect.bottom &&

                    potRect.bottom > gemOrStoneRect.top) {

*// 发生碰撞*

                    if (durability === 1) {

*// 添加音效*

                        playSoundLose();

                    }

                    if(durability === 4){

                        playSoundF();

                    }

                    if (*gemOrStone*.score > 0) {

*// 如果是宝石，增加得分*

                        playSoundGem();

                        score += *gemOrStone*.score;

                        gameScore.innerHTML = "得分："+score;

                    } else {

*// 如果是石头，减少耐久度*

                        durability--;

                        gameDurability.innerHTML = "耐久度："+durability

                        if(durability === 0){

                            stopGameArea();

                        }

                    }

*// 移除宝石或石头*

                    if(*gemOrStone*.parentNode !== null){

                        gameArea.removeChild(*gemOrStone*);

                    }

                }

            }

            function **gameLoop**() {

                const gemsAndStones = document.querySelectorAll('.gem , .stone')

                for (var i = 0; i < gemsAndStones.length; i++) {

                    checkCollision(gemsAndStones[i]);

                }

                if (score >= 40) {  *// 当得分达到一定值时执行某些函数*

                    sp = 16;

                    createNum = 100 ;

                }else if(score >=80){

                    sp = 6;

                    createNum = 10;

                }

                if (durability <= 0) {

                    endGame();

                }

            }

        }

        function **stopGameArea**() {

            clearInterval(createGemOrStoneInterval)

*// 停止游戏并清除游戏区域的代码*

            while (gameArea.firstChild) {

                gameArea.removeChild(gameArea.firstChild);

            }

*// 停止游戏*

            score = 0;

            durability = 5;

            gameScore.innerHTML = "得分："+score;

            gameDurability.innerHTML = "耐久度："+durability

            clearTimeout(timer);

        }

*//宝石和石头运动轨迹和速度*

        function **moveGemOrStone**(*element*){

*// width: 1000px;*

*// height: 800px;*

            let targetX1 = Math.random()\*1000;

            let targetY1 = 800;

            let targetX2 = 1000;

            let targetY2 =  Math.random()\*800;

            let targetX3 = 0;

            let targetY3 = Math.random()\*800;

            let prop = Math.random();

            if(prop < 0.33){

                moveInterval(targetX1, targetY1, *element*)

            }else if(prop >= 0.33 && prop < 0.66){

                moveInterval(targetX2, targetY2, *element*)

            }else{

                moveInterval(targetX3, targetY3, *element*)

            }

        }

        function **moveInterval**(*targetX*, *targetY*, *element*){

            let currentX = parseInt(*element*.style.left);

            let currentY = parseInt(*element*.style.top);

            let speed = 100;

            let deltaX = (*targetX* - currentX)/speed;

            let deltaY = (*targetY* - currentY)/speed;

            var moveAction = setInterval(function(){

                if(speed < 0){

                    clearInterval(moveAction)

                    if(*element*.parentNode !== null){

                        gameArea.removeChild(*element*)

                    }

                }else{

                    currentX += deltaX;

                    currentY += deltaY;

*element*.style.left = currentX + "px";

*element*.style.top = currentY + "px";

                    speed --;

                }

            }, sp)

        }

        function **openPopup**() {

*// 创建文本窗口元素*

            var popup = document.createElement("div");

            popup.id = "popup";

            popup.style.position = "fixed";

            popup.style.top = "50%";

            popup.style.left = "50%";

            popup.style.transform = "translate(-50%, -50%)";

            popup.style.backgroundColor = "white";

            popup.style.padding = "50px";

            popup.style.borderRadius = "5px";

            popup.style.border = "1px solid black";

*// 创建文本元素*

            var text = document.createElement("p");

            text.textContent = "你将用一个聚宝盆接住宝石，并获得积分，当积分达到一定值时或再点击开始游戏，会有意想不到的效果。聚宝盆只有中心接触才能获得宝石，并且你将同时躲避灰色石头的到来。";

*// text.style.*

            text.style.width = "150px";

            text.style.margin = "0";

*// 创建确认按钮元素*

            var confirmButton = document.createElement("button");

            confirmButton.textContent = "确认";

            confirmButton.style.position = "absolute";

            confirmButton.style.bottom = "5px";

            confirmButton.style.right = "5px";

            confirmButton.addEventListener("click", function() {

*// 点击确认按钮后，将窗口移动到屏幕右边*

            popup.style.left = "90%";

            });

*// 将文本元素和确认按钮添加到文本窗口中*

            popup.appendChild(text);

            popup.appendChild(confirmButton);

*// 将文本窗口添加到页面中*

            document.body.appendChild(popup);

        }

*// 在页面加载完成后调用openPopup函数*

            window.addEventListener("load", openPopup);

    </script>

</body>

</html>

# 四、游戏开发日志

设计思路与实际实现功能有所差异

文本

描述已自动生成文本

描述已自动生成

1.首先我对cursor进行大致的框架建构请求，并开始对按钮等细节一一调整。

文本

描述已自动生成

2.这个是三秒倒计时的代码请求，但是其表达方式有所改变，因为clearTimeout和setInterval的终止区别,为了代码顺利实现，显示倒计时和倒计时是两个不同的地方，因此倒计时在第2s才出现，也因此这是一个不美好的**误点。**

文本

描述已自动生成电脑屏幕截图

描述已自动生成

3.这是对框架内每一个函数进行修正，因为有很多没有应用到的函数需要自己修改，所以很多功能有时候只是反复进行提问来求新。后来我一步步进入。

文本

描述已自动生成

4.有下面一些封装没有弄好时会出现这样的情况，我还不知道，但是当我一条条去看懂时我才慢慢了解，代码要么是没有全局变量，要么是少了括号，要么是定义未使用出现console的错误提示修正后才能有所有功能的正常运行。

文本

描述已自动生成

文本

描述已自动生成

文本

描述已自动生成

电脑屏幕截图

描述已自动生成

5.这些是我的重要代码，实现了核心的宝石和石头运动功能、达到一定条件的状态设置和游戏提示功能。