**“打鼹鼠”游戏实验报告**

小组成员： 宋斌（负责代码编写） 2160506043

杜炎（负责实验报告编写） 2160506060

王安邦（负责科技论文编写） 2160506044

指导老师： 杨琦

实验日期： 2018.10.15-2018.10.31

1. 实验目的

通过编写“打鼹鼠”游戏程序，分析了解“打鼹鼠”游戏的功能实现过程，熟悉Javascript的语法，掌握游戏设计的有关知识，提高编程能力。

二、打地鼠游戏的设计思路

1.游戏中的背景是一张图片，固定在指定位置显示，在游戏中没有任何动作，所以没有关于背景的Javacript程序需要编写。

2.设计打鼹鼠游戏的关键，在于使用vue.js框架实现鼹鼠的随机出现以及鼹鼠的消失。鼹鼠每3秒随机生成一个，鼹鼠随机出现后，若被玩家打中，就立即消失，同时打中的鼹鼠数加1；若鼹鼠未被打中，则在几秒钟延时后消失。

3.页面上会动态地显示玩家打了几只鼹鼠，方便玩家了解自己的成绩。

3.鼹鼠出现以及消失时有一个动画效果，通过css中的transition: background-image 0.5s;来实现。

4.玩家打中鼹鼠后会有提示音，给玩家较好的游戏体验。

5.当页面上的鼹鼠数超过6时，游戏结束，页面上会有提示框显示“Game Over!You have whaced n moles”(n为玩家在游戏结束时打中的地鼠总数)。

1. html代码

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<title>Whac a Mole</title>

<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="stylesheet" href="./css/main.css" />

<script src="./js/vue.js"></script>

<script defer src="./js/main.js"></script>

</head>

<body>

<main id="game-container">

<audio src="./audio/hit.wav" ref="hit"></audio> //插入打鼹鼠音频

<div class="start-action-container">

<h1 class="title">Welcome to play Whac-a-Mole!</h1>

<a v-show="gameProcess === null" class="button" @click="gameStart">Start Game</a>

<p v-show="gameProcess !== null" class="count">You whac {{ rank }} moles</p>

</div>

<div class="game-map">

<div class="game-mole"

v-for="i in 9"

:key="i"

:id="i"

:class='{"mole-coming":moleStatus[i]}'

@mousedown="whac">

</div>

</div>

</main>

</body>

</html>

四、Javascript代码

var whacAMole = new Vue({

el: '#game-container',

data: {

count: 0,

rank: 0,

moleStatus: [false, false, false, false, false, false, false, false, false, false],

gameProcess: null,

gameOver: false,

gameSpeed: 300

},

//定义属性

methods: {

whac: function(event) {

this.$refs.hit.pause();

this.$refs.hit.play();

//设置打中鼹鼠时的音效

if (this.gameOver) {

return;

}

//判断游戏结束是否结束，若游戏已结束，则点击鼹鼠无效

var index = event.target.id;

if (this.moleStatus[index]) {

this.rank++;

this.count--;

Vue.set(this.moleStatus, index, false);

}

},

//玩家打中鼹鼠时，打中的鼹鼠数加1，页面上的鼹鼠数减1

gameStart: function() {

for (let i in this.moleStatus) {

this.moleStatus[i] = false;

}

this.count = 0;

this.rank = 0;

this.gameOver = false;

this.gameSpeed = 300;

setTimeout(this.render, this.gameSpeed);

},

//开始时每0.3秒生成一只鼹鼠

render: function() {

let nextMole = this.getNextMole();

this.count++;

if (this.count >= 6) {

this.gameOver = true;

alert(`Game Over! You have whaced ${this.rank} moles`);

}

//判断页面上的鼹鼠数超过6.若超过6，游戏结束，页面上会有提示框显示“Game Over!You have whaced n moles”(n为玩家在游戏结束时打中的地鼠总数)

if (!this.gameOver) {

Vue.set(this.moleStatus, nextMole, true);

if (this.gameSpeed >= 250) {

this.gameSpeed--;

}

setTimeout(this.render, this.gameSpeed);

}

},

//随着时间推移，鼹鼠出现的速度变快，最快速度每0.25秒出现一只鼹鼠

getNextMole: function() {

while (true) {

let nextMole = parseInt(Math.random() \* 9) + 1;

if (!this.moleStatus[nextMole]) {

return nextMole;

}

}

}

//随机生成下一只地鼠

}

});

1. css代码

\* {

padding: 0;

margin: 0;

}

#game-container {

width: 100%;

height: 100%;

margin-top: 10vh;

}

.game-map {

width: 480px;

height: 480px;

margin: 0 auto;

display: flex;

flex-direction: row;

flex-wrap: wrap;

justify-content: space-around;

}

//设置背景的属性值

.game-mole {

box-sizing: border-box;

width: 160px;

height: 160px;

border: 1px solid grey;

cursor: pointer;

background-image: url(./../img/bg.png);

background-size: cover;

transition: background-image 0.5s;

-webkit-transition: background-image 0.5s

}

//设置鼹鼠图片的属性值，实现鼹鼠出现及消失时的动画效果

.mole-coming {

background-image: url(./../img/mole.png);

background-color: #883d0b;

}

//实现鼹鼠出现及消失时的动画效果

.start-action-container {

margin: 10px 0;

text-align: center;

}

.title {

margin: 10px;

}

.button {

display: inline-block;

padding: 10px;

border: 1px solid black;

border-radius: 10px;

cursor: pointer;

}

//设置Start Game按钮的效果

六、心得体会

在设计打鼹鼠游戏及编写代码的过程中,我们熟悉了Javascript的语法，对Vue.js框架的使用更加熟练了，更感受到了亲自编写程序、调试程序的重要性。同时在这个过程中我们也使用了html5和css3的新标准，使用了动画效果，编程能力得到了很大的提高，获益匪浅。