信息科学技术学院实验报告

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **课程名称** | | **高级网页设计** | | | | **班级** | |  |
| **实验名称** | | **高级网页设计大作业** | | | | **教导教师** | | **曾少宁** |
| **姓名** | **陈晓佳** | | | **学号** | **1714080903117** | | **日期** | **2019.5.11** |
|  |  | |  | | |  | |  | |

**引言**

动机：小游戏

作用：娱乐

新意：玩

功能：鲨鱼吃金币、倒计时、计分

页面：鲨鱼吃金币

**一、代码实现**

1. **页面一：**

// main.js

import Player from './player/index'

import Enemy from './npc/enemy'

import BackGround from './runtime/background'

import GameInfo from './runtime/gameinfo'

import Music from './runtime/music'

import DataBus from './databus'

let ctx = canvas.getContext('2d')

let databus = new DataBus()

/\*\*

\* 游戏主函数

\*/

export default class Main {

constructor() {

// 维护当前requestAnimationFrame的id

this.aniId = 0

this.restart()

// this.time = setTimeout(function () { databus.gameOver = true; }, 5000)

}

restart() {

databus.reset()

canvas.removeEventListener(

'touchstart',

this.touchHandler

)

this.time = setTimeout(function () { databus.gameOver = true; }, 13000)

this.bg = new BackGround(ctx)

this.player = new Player(ctx)

this.gameinfo = new GameInfo()

this.music = new Music()

this.bindLoop = this.loop.bind(this)

this.hasEventBind = false

// 清除上一局的动画

window.cancelAnimationFrame(this.aniId);

this.aniId = window.requestAnimationFrame(

this.bindLoop,

canvas

)

}

/\*\*

\* 随着帧数变化的敌机生成逻辑

\* 帧数取模定义成生成的频率

\*/

enemyGenerate() {

if ( databus.frame % 30 === 0 ) {

let enemy = databus.pool.getItemByClass('enemy', Enemy)

enemy.init(6)

databus.enemys.push(enemy)

}

}

// 全局碰撞检测

collisionDetection() {

let that = this

databus.bullets.forEach((bullet) => {

for ( let i = 0, il = databus.enemys.length; i < il;i++ ) {

let enemy = databus.enemys[i]

if ( !enemy.isPlaying && enemy.isCollideWith(bullet) ) {

// enemy.playAnimation()

// that.music.playExplosion()

bullet.visible = false

// databus.score += 1

break

}

}

})

for ( let i = 0, il = databus.enemys.length; i < il;i++ ) {

let enemy = databus.enemys[i]

if ( this.player.isCollideWith(enemy) ) {

enemy.playAnimation()

that.music.playExplosion()

databus.score += 1

break

}

}

}

touchEventHandler(e) {

e.preventDefault()

let x = e.touches[0].clientX

let y = e.touches[0].clientY

let area = this.gameinfo.btnArea

if ( x >= area.startX

&& x <= area.endX

&& y >= area.startY

&& y <= area.endY )

this.restart()

}

render() {

ctx.clearRect(0, 0, canvas.width, canvas.height)

this.bg.render(ctx)

databus.bullets

.concat(databus.enemys)

.forEach((item) => {

item.drawToCanvas(ctx)

})

this.player.drawToCanvas(ctx)

databus.animations.forEach((ani) => {

if ( ani.isPlaying ) {

ani.aniRender(ctx)

}

})

this.gameinfo.renderGameScore(ctx, databus.score)

if ( databus.gameOver ) {

this.gameinfo.renderGameOver(ctx, databus.score)

if ( !this.hasEventBind ) {

this.hasEventBind = true

this.touchHandler = this.touchEventHandler.bind(this)

canvas.addEventListener('touchstart', this.touchHandler)

}

}

}

update() {

if ( databus.gameOver )

return;

this.bg.update()

databus.bullets

.concat(databus.enemys)

.forEach((item) => {

item.update()

})

this.enemyGenerate()

this.collisionDetection()

if ( databus.frame % 20 === 0 ) {

this.player.shoot()

this.music.playShoot()

}

}

// 实现游戏帧循环

loop() {

databus.frame++

this.update()

this.render()

this.aniId = window.requestAnimationFrame(

this.bindLoop,

canvas

)

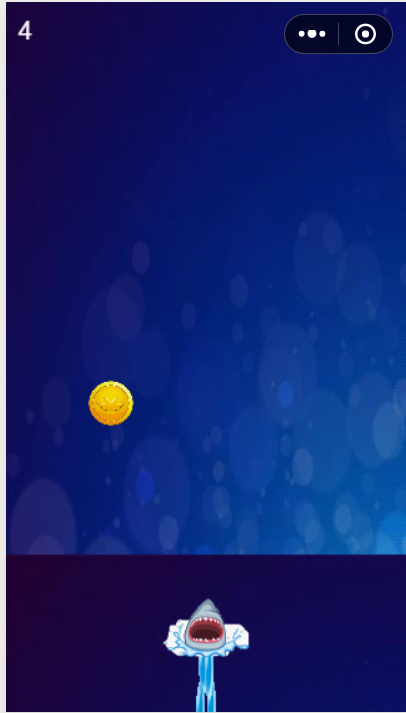
}

}

这是游戏的主界面

**二、运行结果截图**

1. **页面一：**



游戏运行

中

的界面，吃到快速掉落的金币即可得分。

1. ？？截图

用一段话描述这个截图，至少3行文字。

1. **页面二：**



13s时间倒计时结束后游戏结束

后

的得分界面

用一段话描述这个截图，至少3行文字。

**三、实验完成情况**

表1：实验完成情况汇总表

|  |  |  |  |
| --- | --- | --- | --- |
| 实验次数 | 实验名称 | 是否通过 | 备注 |
| 1 | Web服务器与调试工具 | Merged |  |
| 2 | HTML5语义标签 | Merged + Good | 唯一的Good |
| 3 | CSS3样式设计 | Merged |  |
| 4 | JavaScript编程 | Merged |  |
| 5 | HTML5移动端页面（选学） |  | 提交了但是没通过 |
| 6 | HTML5小程序页面 |  |  |
| 7 | 微信小程序API用法 |  |  |

**总结**

感觉还行吧。

最后贴上已发布小程序的二维码。

不会发布啊。。

临时的二维码

