HTML5 实验报告

陶晓岚 17307110131

摘　要：本次居家实验，自主学习了有关 html5 及 canvas 相关方面的内容，对于 JavaScript 有了更多的了解。并应用相关内容设计编写了 3 个程序，第一个为小球斜抛的物理过程模拟，第二个为利用动态图模拟生命游戏，第三个程序在生命游戏的基础上实现了初步的传染病传播模拟。

一、 引言

HTML5 是构建 Web 内容的一种语言描述方式，是构建以及呈现互联网内容的一种语言方式，其被认为是互联网的核心技术之一。HTML 产生于 1990 年，1997 年 HTML4 成为互联网标准，并广泛应用于互联网应用的开发，HTML5 则是互联网的下一代标准。

HTML5 将 Web 带入一个成熟的应用平台，在这个平台上，视频、音频、图像、动画以及与设备的交互都进行了规范。其具有非常多样化的功能，包括 Canvas 画布，智能的属性表单及多媒体等。

本次实验的主要内容通过 HTML5 的画布功能实现物理功能模拟，并通过表单和鼠标控制改变物理过程的参数值，最后完成网页中物理过程的模拟。

二、 实验原理

1. HTML 语言[1]

HTML 文档由 HTML 元素定义，其语法为<开始标签>元素内容</结束标签>，部分元素为空元素，在开始标签中结束，即<开始标签/>。可以通过<title>,<body>等定义网页的标题以及主题内容。

2. CSS 层叠样式表

CSS 是一种用来表现 HTML 等文件样式的语言，可以静态或动态的修饰网页。其规则由选择器及一条或多条声明构成。语法为选择器{属性：值； 属性：值}，其常用选择器有#id，*，element（如 p，h1 等）。常用属性有 background，

margin，font，color 等，每一种颜色都有对应的编号或由 rgb 数组表示。

3. Canvas 画布

使用 JavaScript 进行编辑，其是一种轻量级的编程语言，插入 HTML 页面后由浏览器执行。使用 Canvas 的过程主要为

(1) 在 html 文件中添加<canvas>标签，并规定其 id，高度，宽度。

(2) 创建 JavaScript 文件并通过<script src=' id'></script>将其插入到 HTML 中

(3) 在 JS 文件中，利用 var canvas = document.getElementById("myCanvas"),context = canvas.getContext("2d")；链接至画布

(4) 利用 stroke，moveTo，lineTo，arc 函数画线，圆，矩形等图形。

4. HTML 表单

HTML 表单用于收集不同形式的用户的输入，如文本域，单选框，滑动条等。表单事件在 HTML 事件中被触发。其常用事件为 onchange。使用表单的主要过程为：

(1) 在 html 文件中添加<input></input>标签，标签属性通过 id，type，class 等控制。

(2) 在 js 文件中利用 document.getElementById 函数获取对应的表单

(3) 编辑表单函数。

5. 随时间变化的图案。

本次学习的过程中，接触到了 2 种使函数随时间变化的方法。

第一种方式为调用 util.js 文件，并使用 requestAnimationFrame(func)函数，其作用是能够在下一帧时调用 func 函数。通过在 func 函数中加入 clearRect() 函数擦除画布内容并重新绘画即可实现在 canvas 中实现动画。

第二种方式为使用 setInterval(func，time)函数[2]，其作用为每隔一定时间(time)就调用 func 函数。本次编写的程序中生命游戏部分的更新即以此为根据编写。

6. 鼠标事件

Canvas 中还可以通过鼠标的点击与移动影响图像的变化。常用的鼠标事件有 onclick,onmousedown, oumouseup, onmousemove 等。具体过程如下：

(1) 通过 WindowstoCanvas 函数将鼠标在整个页面中的位置转换为在画布内的坐标。

(2) 利用 onmousemove 函数确定鼠标在不同位置的形态

(3) 利用 onmouseup 与 onmousedown 函数规定鼠标点击与松开后对应的变化。

同时，也可以利用 addEventListener(event, function,useCapture)[3]函数，其可以将事件处理程序附加到指定的元素上。其第一个参数为事件的类型，如‘click’，第二个参数为在事件发生时调用的函数。通过这个函数也能实现鼠标点击控制 canvas 的数据。

三、 实验过程及结果

本次学习过程中主要完成了 2 个程序的编辑，第一个为模拟小球的斜抛运动，第二个为模拟了生命游戏的过程。以下分别介绍。

3.1 小球斜抛运动模拟

设计思路：想要实现一个小球在高台上斜抛的物理模拟过程，要求能随意改变抛出的速度以及角度，同时能够返回小球落地位置距离抛出点位置的水平距离。因此想到可以模拟一个类似弹弓的发射装置，将发射速度与拉伸长度挂钩，同时要求可以调整其弹性系数以及小球的质量以调整发射速度。

程序运行的截图见图 1，其主要能够实现如下功能：

1. 在静止状态下，可以用鼠标拖动小球以调整弹弓的拉伸长度以及角度（此处近似将弹弓当作弹簧处理，其拉力正比于长度）。同时在画布右上角显示相应

长度与角度。

2. 点击开始键，小球即被发射出去，在接触地面时停止，并在 Canvas 的右下角显示落地位置距离弹弓的水平距离（此处设置 Canvas 画布高度为 5m）。

3. 点击重置键，小球回复发射前位置。

4. 在停止状态下，可以拖动下方滑块调整弹弓的弹性系数以及小球质量。

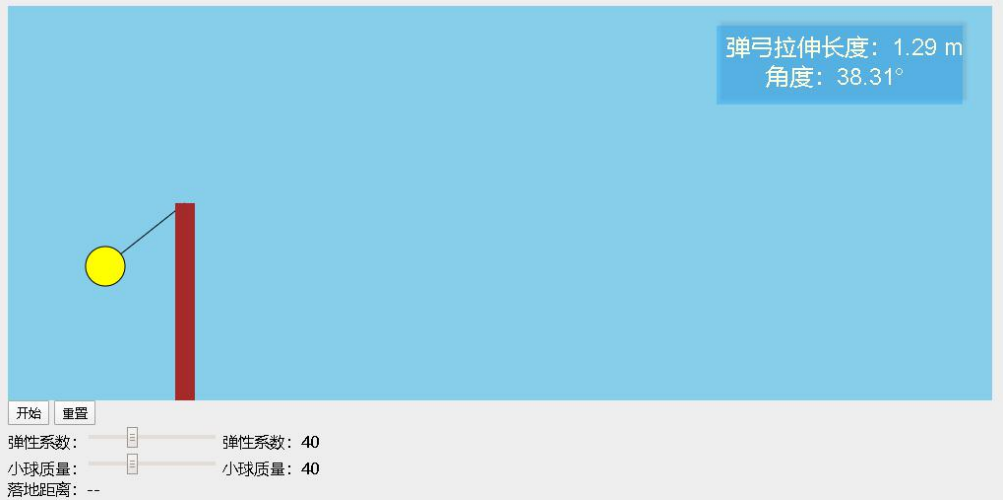

图 1

3.2 生命游戏的模拟

生命游戏规则：二维体系中，每个细胞有 2 种状态，分别为生或死。其下一刻的状态取决于周围 8 个细胞的状态：

（1）若细胞周围有 3 个细胞存活，无论其为何状态，下一刻都为生；

（2）若细胞周围有 2 个细胞存活，其下一刻状态保持不变；

（3）其余情况下，该细胞下一刻状态转变为死；

设计思路：

1. 首先需要构建一个二维网格，每一个格子代表一个细胞，用二维数组储存原胞的位置和状态，0 表示死，1 表示生。在画布上用黑色表示其状态为生，白色表示其状态为死。

2. 生命游戏需要的是对状态的更新而非连续的动画，因此选用 setInterval() 函数更加合适，其作用是每隔一定时间（此处为 500ms）更新一次画面，显示下一刻的细胞状态。同时可以通过 start/stop 按钮控制更新的运行和停止。

3. 利用 count_neighbors 函数获取每个细胞周围的存活细胞数，并以此为依据判断下一刻该细胞的状态。此处需注意不可以直接对原数组进行修改，否则在循环过程中，已判断状态的细胞会导致未判断状态细胞判断的错误。因此需要对原数组 cells 进行一次拷贝得到 cells2，通过 cells 进行判断，并将数组的更新应用于 cells2。此处不能用简单的相等，否则对于 cells2 的操作也会影响到 cells。因此通过使用 JSON.parse(JSON.stringify(obj))函数进行深拷贝[4]得到 2 个一模一样又完全独立的数组。

4. 添加一些辅助用功能：如 step 按钮控制系统进行步进操作，便于检验。Reset 键可以重新生成细胞的状态。同时设置一个进度条可以调整每次重置后随机生成细胞中存活的细胞数。并在运行过程中，显示存活细胞数。

5. 为更方便探索不同图样的演化过程，加入鼠标点击控制方式。利用 addEventListener('mousedown',func)函数，使得鼠标每次点击都可以改变相应格子里的细胞的状态。

程序结果：图 2、图 3 分别为运行和停止时的界面截图。其功能大致与思路中所述类似：点击 start 键开始运行；点击 stop 停止运行；step 步进操作；reset 重新生成细胞状态；滑动条改变生成生存的细胞数量；鼠标点击改变细胞状态；同时在右下角显示系统的运行状态和当前存活的细胞数。

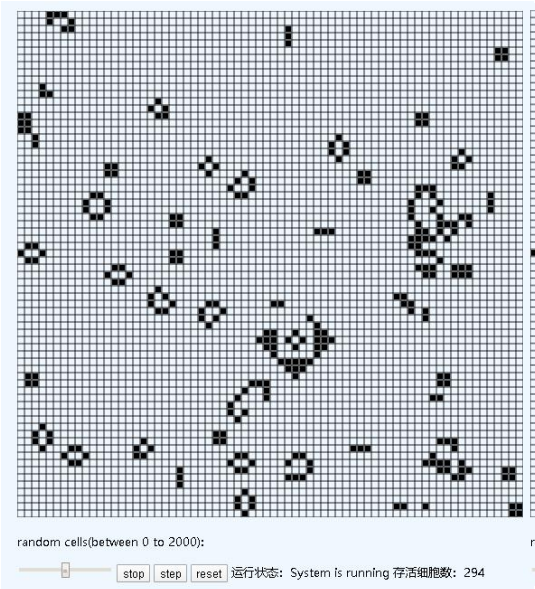

图 2

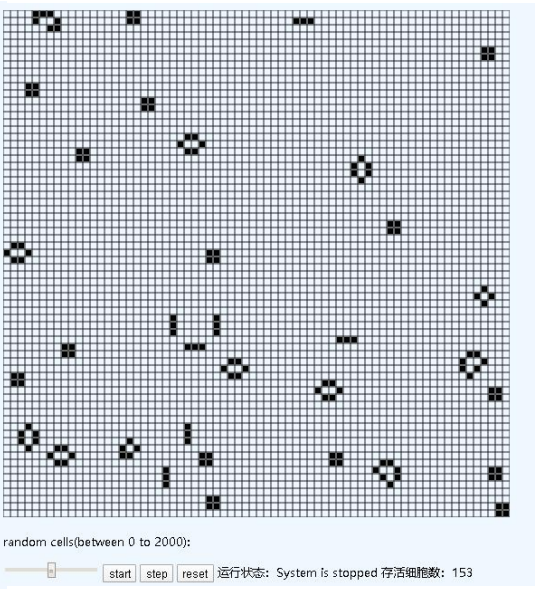

图 3

3.3 生命游戏的改进—传染病传播的初步模拟

设计思路：由于在上一个程序中已经设计了完备的网格模型及相应的操作，因此在此基础上通过改变格点变色规则可以来简单模拟疫情的传播。我们以黑色格子代表被传染人群，白色格子代表未传染人群。

首先需要设定一个传染概率 P，假设每位被感染者只能影响到周围 8 个人。同时假设对于每位未感染者，其在下一刻可能被感染的概率不仅与 P 有关，也与其周围的患者数有关：P 越大，周围患者数越多，则越容易被传染。因此将规则设计为：对于某格点，随机生成 0-1 的数 a,设其周围患者数为 n，假设当 a/sqrt(n)< P 时，这个格点下一刻被感染。

为使得疫情蔓延更易于观察，这里去除了上个程序中随机生成点的模式，仅采用用鼠标点击添加传染源的方式。设定 P=20%时，在画布中心放置一个传染源后感观察染人群随时间(迭代次数)的变化（见图 4）。可以看到黑点蔓延速度的显著增大，最后在 70 步左右全部人群被传染。

除此以外，我们还需要考虑死亡或者被治愈的群体，这一类人不应被计入在被传染人数内，也不会继续传染疾病。由于无法准确判断死亡或治愈，因此将 2 者概率一起计算，令为 R，并用灰色方块代表这一类人群。此处假设当一个患者已经处于生病状态 30 天（30 次迭代）以上时才有死亡或治愈的可能。图 5 的过程为单个传染源位于画布中心，P=20%，R=15%的情况。

需要注意的是，这个模型仅仅是对于传染病传播的最为初步和简单的模拟。其将所有人都限制到了一个格子内，忽略了其自由移动的可能性，并且认为每个人均能且仅能与周围 8 个格点发生接触。同时没有考虑医院，隔离措施，公共场所，小区等各类措施和地区对传播的影响。更精确的模拟应将每个人当作可以自由运动或被隔离在某个区域的粒子，其传染方式也不仅仅限于周围 8 个点，而是通过类似相互作用的方式计算被感染的几率[5]。

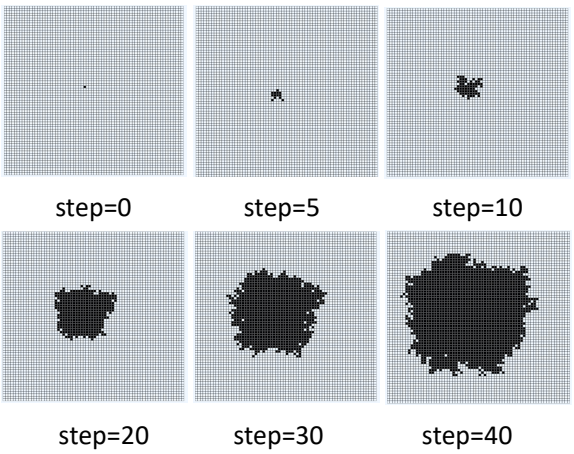

图 4

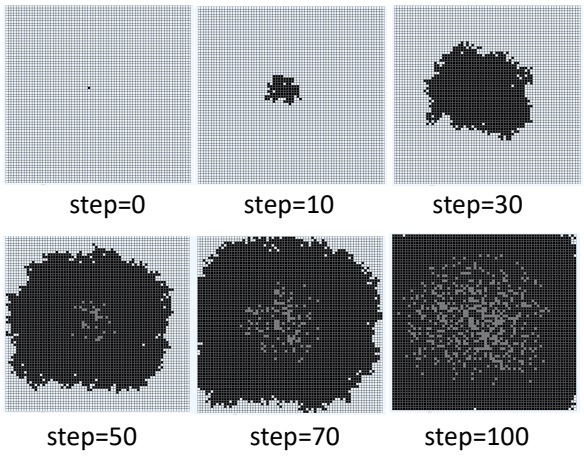

图 5

四、 实验结论

本次实验学习了 HTML5 及 Canvas 相关内容，设计并实现了 3 个内容的动态模拟。分别为小球斜抛运动，生命游戏以及传染病传播的简单模拟，均能较好实现期望的功能。在此过程中学会并应用了 Canvas 的表单，按钮，鼠标事件等的语法及运用，对于 html5 的编辑和使用有了更加深刻和全面的认识。

五、 参考文献：

[1] file:///C:/Users/mac/Desktop/html5/HTML5%E5%85%A5%E9%97%A8/main.html

[2] https://www.w3school.com.cn/jsref/met_win_setinterval.asp

[3] https://www.cnblogs.com/jc2182/p/11307165.html

[4] https://zhuanlan.zhihu.com/p/41058925

[5] https://www.bilibili.com/video/BV1dK411V7Xm?from=search&seid=8436961396151537554