实验五 HTML5画布API项目

说明

- 本次实验主要包含了两个基于HTML5画布API的应用设计实例，一是手绘时钟的设计与实现（必做），二是拼图游戏的设计与实现（选做）。至少选做其中一个实例；
- 在手绘时钟项目中，主要难点为时钟刻度与时钟的绘制以及时间实时更新的动画效果；
- 在拼图游戏项目中，主要难点为拼图画面初始化、鼠标点击事件、游戏计时功能、游戏成功与重新开始的判定。

目录

5.1 手绘时钟的设计与实现

5.1 手绘时钟的设计与实现

功能要求：

不依赖于任何图片素材，完全基于**HTML5**画布**API**绘制时钟，并实现每秒更新的动态效果。

5.1 手绘时钟的设计与实现

5.1.1 界面设计

1. 画布的创建

- 使用<canvas>标签
- 属性：style="border:1px solid"

5.1 手绘时钟的设计与实现

5.1.1 界面设计

- **2.** 绘制时钟刻度
  - 自定义**drawClock()**
  - （**1**）设置画笔样式和位置
  - （**2**）画**12**小时的刻度（使用**for**循环**12**次）
    - **rotate()**:旋转
    - **moveTo()**和**lineTo()**：移动和画线段
    - **stroke()**：描边路径

5.1 手绘时钟的设计与实现

5.1.1 界面设计

- **2. 绘制时钟刻度cont.**
  - **（3）画60分钟对应的刻度（使用for循环60次）**
    - **rotate():旋转**
    - **moveTo()和lineTo()：移动和画线段**
    - **stroke()：描边路径**

5.1 手绘时钟的设计与实现

5.1.1 界面设计

- **3.** 绘制时钟指针
  - （1）获取当前时间:
    - **var now = new Date();**
    - **getSeconds()**
    - **getMinutes()**
    - **getHours()**
    - 换算成12小时制（例如23点换算为11点）

5.1 手绘时钟的设计与实现

5.1.1 界面设计

- **3.** 绘制时钟指针
  - （2）绘制时针：

时针的角度 = 360/12 x小时 + 360/12/60 x分钟+ 360/12/60/60 x 秒

  - 换算成弧度：

时针的弧度 = π/6 x小时+π/360 x分钟+ π/21600 x秒

5.1 手绘时钟的设计与实现

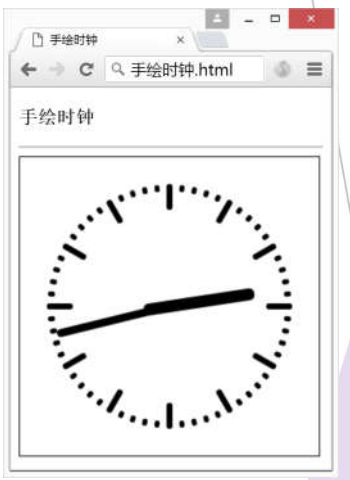
5.1.1 界面设计

- **3.** 绘制时钟指针
  - （3）绘制分针：

分针的角度 = 360/60 x分钟+ 360/60/60 x 秒

  - 换算成弧度：

分针的弧度 = π/30 x分钟+ π/1800 x秒



5.1 手绘时钟的设计与实现

5.1.1 界面设计

- **3.** 绘制时钟指针
  - （4）绘制秒针：

分针的角度 = 360/60 x 秒

  - 换算成弧度：

分针的弧度 = π/30 x秒

5.1 手绘时钟的设计与实现

5.1.1 界面设计

- **4.** 绘制时钟表盘
  - （1）设置样式
  - （2）绘制表盘：**arc()**
  - （3）描边路径：**stroke()**
  - （4）恢复初始状态：**restore()**

5.1 手绘时钟的设计与实现

5.1.2 实时更新效果

- **1.** 每秒刷新一次画面

```
setInterval("drawClock()", 1000);
```

- 注意：此时会变成错误的重叠效果
- 解决方案：每次刷新前需要清空画布

5.1 手绘时钟的设计与实现

5.1.2 实时更新效果

- **2.** 清空画布

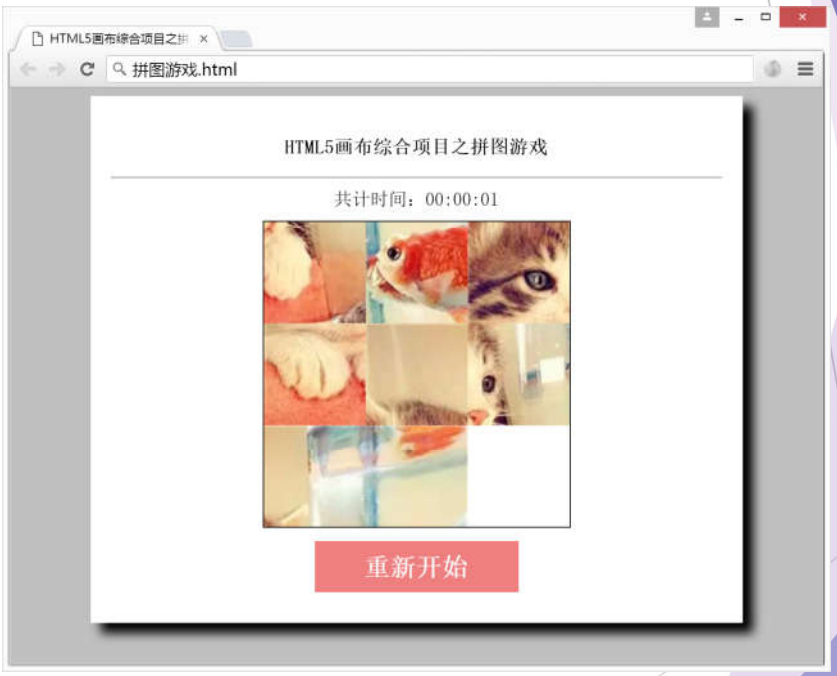
```
ctx.clearRect(0, 0, 300, 300);
```

  - 形成动态效果

5.2 拼图游戏的设计与实现

5.2 拼图游戏的设计与实现

- 功能要求：
  - 使用HTML5画布技术制作一款拼图小游戏，要求将图像划分为3*3的9块方块并打乱排序，用户可以移动方块拼成完整图片。



5.2 拼图游戏的设计与实现

- 5.2.1 界面设计
  - 1.整体布局设计
    - 使用 划分区域
    - CSS外部样式表：pintu.css

5.2 拼图游戏的设计与实现

- 5.2.1 界面设计
  - 2.游戏布局设计
    - 游戏计时
    - 游戏画布
    - 重新开始按钮

5.2 拼图游戏的设计与实现

- 5.2.2 实现游戏逻辑
  - 1.准备工作
    - 声明画布对象：ctx
    - 图片预加载：pintu.jpg

```
1.  //声明拼图的图片素材来源
2.  var img = new Image();
3.  img.src = "image/pintu.jpg";
4.  //当图片加载完毕时
5.  img.onload = function() {
6.      //游戏相关代码
7.  }
```

5.2 拼图游戏的设计与实现

00	01	02
10	11	12
20	21	22

- 5.2.2 实现游戏逻辑
  - 2.初始化拼图
    - 需要将素材图片分割成3行3列的9个小方块，并打乱顺序放置在画布上。
    - 为了在游戏过程中便于查找当前的区域该显示图片中的哪一个方块，首先为原图片上的9个小方块区域进行编号。

```
1. //定义初始方块位置
2. var num = [[00, 01, 02], [10, 11, 12], [20, 21, 22]];
```

5.2 拼图游戏的设计与实现

- 5.2.2 实现游戏逻辑
  - 2.初始化拼图
    - 自定义名称的generateNum()方法用于打乱拼图顺序。

```
1.  //打乱拼图的位置
2.    function generateNum() {
3.      //循环50次进行拼图打乱
4.      for (var i = 0; i < 50; i++) {
5.        //随机抽取其中一个数据
6.        //再随机抽取其中一个数据
7.        //对调它们的位置
8.      }
9.    }
```

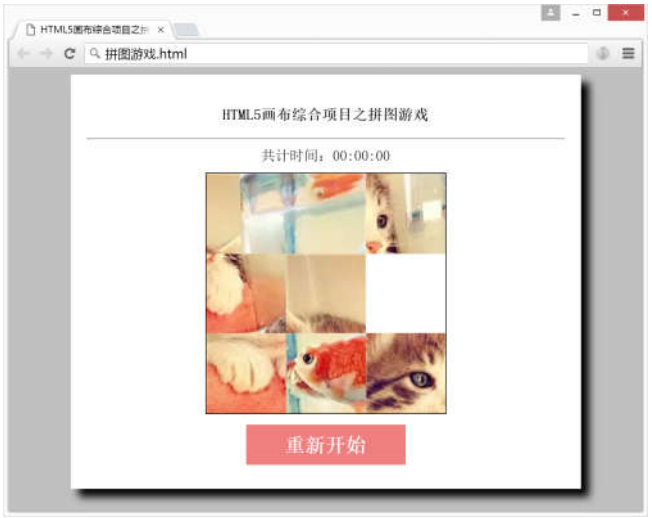
5.2 拼图游戏的设计与实现

- 5.2.2 实现游戏逻辑
  - 2.初始化拼图
    - 自定义名称的drawCanvas()方法用于在画布上绘制乱序后的图片。

```
1.  //绘制图片
2.  function drawCanvas() {
3.      //清空画布
4.      //使用双重for循环绘制3x3的拼图
5.          if (num[i][j] != 22) {
6.           //获取数值的十位数，即第几行
7.           //获取数组的个位数，即第几列
8.           //在画布的相关位置上绘图
9.      //for循环结束
10. }
```

5.2 拼图游戏的设计与实现

- 5.2.2 实现游戏逻辑
  - 2. 初始化拼图
    - 绘制完成的拼图效果。



5.2 拼图游戏的设计与实现

- 5.2.2 实现游戏逻辑
  - 3. 通过鼠标点击移动拼图
    - onmousedown事件

```
1.  //监听鼠标点击事件
2.      c.onmousedown = function(e) {
3.          //获取画布边界
4.          //获取鼠标在画布上的坐标位置(x,y)
5.          //将x和y换算成几行几列
6.          //如果当前点击的不是空白区域
7.          if (num[row][col] != 22) {
8.              //移动点击的方块
9.              //重新绘制画布
10.         }
11. }
```

5.2 拼图游戏的设计与实现

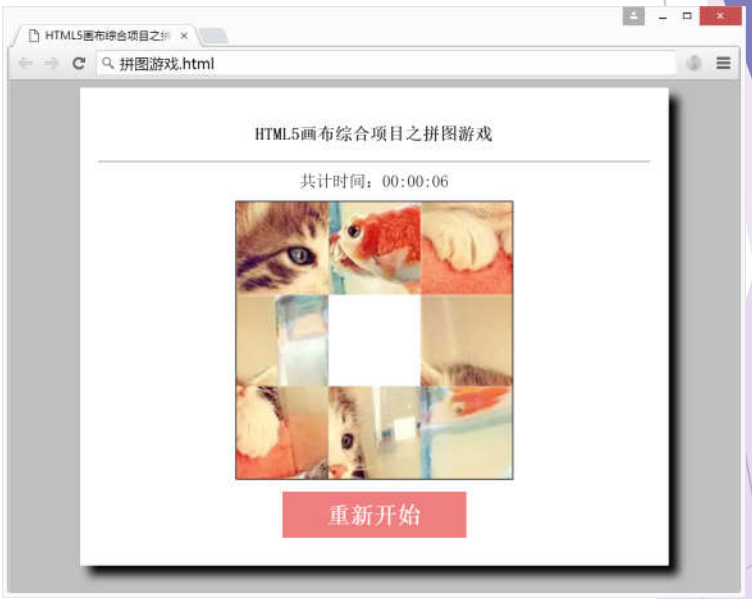
- 5.2.2 实现游戏逻辑
  - 3. 通过鼠标点击移动拼图
    - onmousedown事件

图(a) 拼图移动前的效果图

图(b) 拼图移动后的效果图

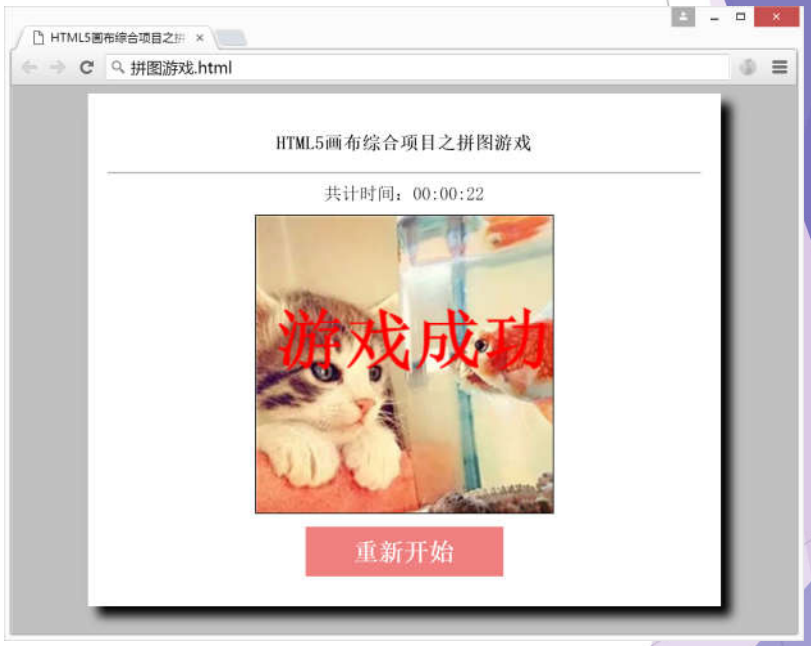
5.2 拼图游戏的设计与实现

- 5.2.2 实现游戏逻辑
  - 4. 游戏计时功能的实现
    - 自定义函数getCurrentTime()用于进行游戏计时；
    - 在JavaScript中使用setInterval()方法每隔1秒钟调用getCurrentTime()方法一次，以实现更新效果。



5.2 拼图游戏的设计与实现

- 5.2.3 游戏成功与重新开始
  - 1. 游戏成功判定与显示效果的实现
    - 自定义函数checkWin()用于进行游戏成功判断。
    - 如果成功则使用clearInterval()方法清除计时器。然后在画布上绘制完整图片，并使用fillText()方法绘制出“游戏成功”的文字图样。



5.2 拼图游戏的设计与实现

- 5.2.3 游戏成功与重新开始
  - 2.重新开始功能的实现
    - 为“重新开始”按钮提供点击事件onclick=“restartGame()”。
    - 声明restartGame()方法，用于重新开始游戏，包括三个部分：
      - 计时器重启clearInterval(timer)
      - 重新打乱拼图顺序generateNum()
      - 重新绘制画布内容drawCanvas()

图(a) 游戏成功画面

图(b) 游戏重新开始画面