第11讲

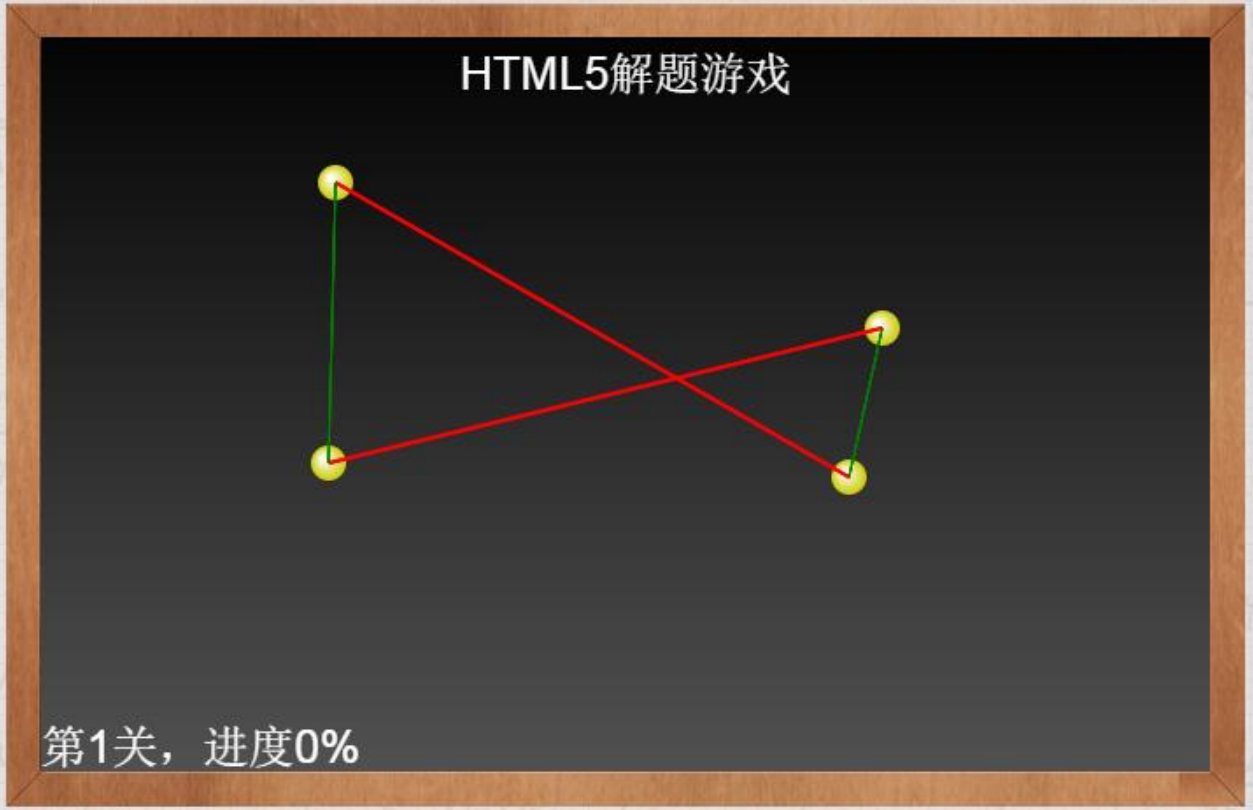
Untangle Puzzle Game

信息学院 孙辉

基于canvas的解题游戏

关于HTML5解题游戏

1. 鼠标点击小球可以随意拖动
2. 不相交的线，比较细，成绿色；相交的线，比较粗，成红色
3. 进度计算方法：绿色线数/总线数*100，100%时进入下一关
4. 共3关，比较简单，初学HTML5做的练习



问题分析

- 过关条件：拖动小球使得所有的线条都不相交
- 游戏中的元素
  - 小球
  - 线
  - 鼠标事件
    - 获取鼠标位置
    - 检测鼠标事件是否发生在小球上
  - 检测线的交点

1、Canvas上画圆

```
$(function(){
        var canvas = document.getElementById("game");
        var ctx = canvas.getContext("2d");
        ctx.fillSyle = "rgba(200, 200, 100, .6)";
        ctx.beginPath();
        ctx.arc(100,100, 50, 0, Math.PI*2, true);
        ctx.closePath();
        ctx.fill();
});
```

2、随机画出5个圆，并保存圆的位置

```
$(function(){

        var canvas = document.getElementById("game");
        var ctx = canvas.getContext("2d");

        var circleRadius = 10;

        var width = canvas.width;
        var height = canvas.height;

        // draw 5 circles randomly;
        var circleCount = 5;
        for (var i = 0; i < circleCount; i++) {
                var x = Math.random()*width;
                var y = Math.random()*height;
                drawCircles(ctx, x, y, circleRadius);
                untangleGame.circles.push(new Circle(x, y, circleRadius));
        }
});
```

3、用直线连接圆

```
function drawLine(ctx, x, y, x2, y2, thickness){
        ctx.beginPath();
        ctx.moveTo(x, y);
        ctx.lineTo(x2, y2);
        ctx.lineWidth = thickness;
        ctx.strokeStyle = "#cfc";
        ctx.stroke();
}
```

```
function Line(start, end, thick){
        this.startPoint = start;
        this.endPoint = end;
        this.thickness = thick;
}
```

```
for (var i = 0; i < untangleGame.circles.length; i++) {

        var startPoint = untangleGame.circles[i];

        for (var j=0; j < i; j++){
                var endPoint = untangleGame.circles[j];
                drawLine(ctx, startPoint.x, startPoint.y, endPoint.x, endPoint.y,
1);
                untangleGame.lines.push(new Line(startPoint, endPoint,
                        untangleGame.thinLineThickness));
                }
```

4、鼠标事件——鼠标按下

```
$("#game").mousedown(function(e) {
        var mouseX = e.pageX-this.offsetLeft;
        var mouseY = e.pageY-this.offsetTop;

        for(var i=0;i<untangleGame.circles.length;i++)
        {
                var circleX = untangleGame.circles[i].x;
                var circleY = untangleGame.circles[i].y;
                var radius = untangleGame.circles[i].radius;
                if (Math.pow(mouseX-circleX,2)
                    + Math.pow(mouseY-circleY,2) < Math.pow(radius,2))
                {
                        untangleGame.targetCircle = i;
                        break;
                }
        }
});
```

鼠标事件——move&up

```
$("#game").mousemove(function(e) {
        if (untangleGame.targetCircle != undefined) {
                var mouseX = e.pageX-this.offsetLeft;
                var mouseY = e.pageY-this.offsetTop;
                var radius = untangleGame.circles[untangleGame.targetCircle].radius;
                untangleGame.circles[untangleGame.targetCircle]
                                = new Circle(mouseX, mouseY, radius);

                connectCircles();
        }
});
```

```
$("#game").mouseup(function(e) {
        untangleGame.targetCircle = undefined;
});
```

游戏主循环

```
function gameloop(){
    var canvas = document.getElementById("game");
    var ctx = canvas.getContext("2d");

    clear(ctx);


    for(var i=0;i<untangleGame.lines.length;i++) {
        var line = untangleGame.lines[i];
        var startPoint = line.startPoint;
        var endPoint = line.endPoint;
        var thickness = line.thickness;
        drawLine(ctx, startPoint.x, startPoint.y, endPoint.x, endPoint.y, thickness);
    }

    for(var i=0;i<untangleGame.circles.length;i++) {
        var circle = untangleGame.circles[i];
        drawCircle(ctx, circle.x, circle.y, circle.radius);
    }
}
```

判断2条线段是否相交

- 两条直线是否相交?
  - 平行
  - 相交——交点
- 交点是否在线段内部?