准备工作：整理出以下要必须知识：

1.手势动画html5的canvas系列函数。

2.手势触发操作touch函数。

3.鼠标触发操作mouse系列函数。

4.bootstrap和vue.js框架

此处有个需要注意的地方：一个画布上画多个独立图形得用到ctx.beginPath();ctx.closePath();来开关路径，否则会出现图形连体的状况。其次最主要要考虑实现：设计一个手势解锁的组件首先应该想到的应该就是先画出九宫格：

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | 9 |

九宫格的思路是这样的：



首先应该确定canvas中九宫格的9个点对应的圆心的坐标位置，但是圆心的位置仅是通过画布确定的，也就是说9个圆心的位置是相对于画布确定的。初始化initDrawCircles()中把9个点全都先放到数组中（这个函数中的算法是在草稿纸上根据规律推演来的）。

另外，每个圆圈都是按照坐标的形式存储，后面的密码校验就需要根据坐标来算出对应的密码（123456789），算法在transToNumber()中，至于base参数可以看成是左上角圆心到画布左边的距离，因为画布是正方形的，且圆心均匀分布，则base可看成是一个基础值，然后以次为基础逐个计算。如果画布位于左上角时正常（左上角位置（0,0）），画布位置变动后就会有问题，如何让画布居中且功能正常？在触发move操作时做了一个获取相对位置的函数getRelativePosition()，此函数分为鼠标滑动和手指移动两种事件，分别做了处理。鼠标移动事件中自动获取相对位置px = event.offsetX; py = event.offsetY;而手势移动需要变动下，简单的加减。

接下来，九宫格圆心的位置确定了，下面开始画圆。使用canvas函数，很简单，不赘述。

下面开始手势操作：

1.线要跟随手指（鼠标）移动；

2.当滑到圆圈中时，立马连接上，并且此时开始从第二个点开始。这里面就有一个思想：重绘。鼠标每移动一下就重绘一次，所以把initCanvas（）函数放到mousemove动作函数中去。所有的校验功能都放到手指（鼠标）抬起事件中，此处的校验分为两部分：setPasswordMode()和validPasswordMode()两种。对应两个radio，里面的校验细节对应360官方文档。