1月18号到1月24号：

1. 使用G6绘制不同形状的图元（三角形，椭圆形，平行四边形）
2. 添加调整图元颜色、大小功能
3. 将图元的关键信息保存并下载（json格式，前端实现）
4. 读取下载的json文件，再将图元绘制到画布上
5. 将信息保存到后端，再传送到前端下载（后端使用java） **（难）**

1月25号到1月31号：

1. 了解canvas画布
2. 查阅canvas画布的基本知识；
3. 初步了解canvas画布的相关方法；
4. 查阅网上现有的canvas画布实例（至少三个），并过一遍代码。
5. 用html和js创建canvas画板
6. 可以选择画笔颜色，至少三个颜色；
7. 可以通过鼠标在画布上进行绘制。
8. 画布添加清屏和撤销功能
9. 可以通过点击按钮进行清屏和画笔撤销功能；
10. 撤销功能可以允许只后退一步操作，有余力可以尝试无限步撤销。
11. 在画布中绘制图形**（难）**
12. 通过点击“绘制直线”按钮，在画布上绘制出直线来；
13. 绘制直线的过程中直线要有拖动效果；
14. 除了直线外，可以尝试为画布添加绘制矩形和圆形等几何图形的功能。
15. 保存canvas画布为图片格式**（难）**
16. 将canvas画布的内容保存为图片；
17. 图片格式要用户自选；
18. 思考可不可以在浏览器弹出的“另存为”窗口的“保存类型”下拉框中选择图片类型。

2月1号到2月7号：

在html页面中实现两个小游戏，包含以下几个步骤：

1. 在html页面中设置能变色的小球

（1）程序界面占满整个用户区，且不允许出现滚动条；

（2）界面的正中央要有一个具有立体感的球；

（3）用鼠标点击球，球会发生变色，且如果鼠标长按，球也会按照恒定的速度不断改变颜色；

1. 设置两种背景色，白色和灰色，当点击球以外的区域时，背景色在白色和灰色之间切换。

提示：

1. CSS盒子模型：padding、margin、border
2. 边框圆角：border-radius
3. 鼠标点击事件：mousedown、mouseup
4. JS循环计时器：setInterval
5. 颜色：rgb、rgba
6. 在html页面中设置能够被拖动的小球
7. 程序界面占满全屏，不允许出现滚动条；
8. 定义一个可以被鼠标自由拖动的球；
9. 球可以借助鼠标的拖动获得速度，当用户松开鼠标时，球将开始进行惯性运动，且可以与边界发生弹性碰撞；
10. 设定球的运动为无摩擦运动，球与边界的碰撞为完全弹性碰撞。

提示：

1. JS可以操作DOM元素的css样式，通过style属性
2. 注意坐标系的换算
3. JS动画帧：requestAnimationFrame
4. 为小球设置障碍物
5. 随机在舞台中添加多个长方形障碍（每次刷新页面后障碍都应该不完全相同），障碍之间不能存在重叠，且不能跑出舞台；
6. 障碍会延其长度或宽度方向做直线往复运动；
7. 障碍上应该有简单的重复性花纹（考虑使用svg或者canvas）
8. 随机在舞台中添加多个小球（位置随机），且不能和障碍以及其他小球出现重叠；
9. 为每一个小球设置一个随机方向，随机大小的速度，让他们在舞台中自由运动。小球可与障碍之间发生弹性碰撞（暂不考虑小球之间的碰撞）；
10. 实现第一个小游戏
11. 做一个简单的小游戏，用户可以控制一个底部的障碍左右移动，来防止球与底部边界接触，在游戏开始时有一个在自由运动的球，舞台的上半部分是密集排列的静止的球，当运动的球与静止的球发生碰撞时，静止的球也将会获得速度开始运动；
12. 要求使用vue来构建整个程序，注意暂不要使用vue-cli，就是用原生HTML/JS/CSS来构建程序；
13. 要求程序有菜单界面，简单的计分系统。
14. 实现第二个小游戏

延续上一个游戏的任务，不过这次要求加入G6，使用G6库来构建舞台。