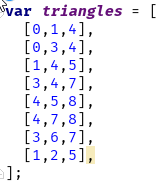
CG Project2说明文档

一、坐标、颜色转化与三角形组合

在config.js文件中，只为点提供了canvas坐标系的坐标和0-255的rgb颜色值，并且，也没有给出具体哪些点对应哪个三角形。因此，在绘图之前，应当对这些坐标和颜色值进行转化，并指明具体的三角形组合。

所以，首先先根据示例，得到了如下图所示的三角形数组。然后实现了transferTriangles()方法，将vertex\_pos数组，转化为glVertex和glVertex2数组，其中glVertex数组是为后期渲染边框准备的。glVertex数组中的元素为每个点的webGL坐标和0-1的rgb值。glVertex2数组中的元素为每个点的webGL坐标和颜色为红色的rgb值。



二、图形的绘制

因为要绘制的图形不属于三角形带和三角形扇，也没有找到合适的方法将其整体进行绘制，最终选择以三角形为单位进行绘制。据此实现了三个方法：draw(), drawFrame(), drawTriangle()。顾名思义，分别是画整体，画单个三角形，画三角形边框的方法。

在draw()方法中，对前面设定好的triangles进行遍历两遍。一遍是画三角形主体，第二遍是画三角形边框，这样使得边框能够像示例展示的，能够一直显现出来。

三、图形的缩放旋转

图形的缩放旋转主要设计到三个方法和一个事件。三个方法分别是getNewAngle(), transferMetrix(angle), action()。getNewAngle()方法能够得到随时间变化的angle，transferMetrix(angle)方法则是根据angle，对图形进行缩放和旋转。这里并没有用内置的矩阵库，因为使用内置的矩阵库进行动画时，并不会涉及到vertex\_pos的改变，而vertex\_pos对于拖拉功能的实现很重要，因此，选择用transferMetrix(angle)方法来实现自己的矩阵计算。

在action()方法中，通过一个名为flag的bool变量，来控制整个动画的开始于停止。实现了一个tick变量，并在该变量中通过调用requestAnimationFrame(tick, canvas)方法来实现动画效果，逐帧绘图。

最后是用document监听一个键盘事件“onKeyDown”。每当按下“T”键时，flag变为 !flag，并调用一次action，以此实现动画的开关效果。

四、图形的拖拽

拖拽效果的实现参考于上一个project，主要由两个方法和两个事件组成。canvas监听鼠标按下事件。得到按下鼠标时的坐标，通过findPoint()方法，判断是否是在某个顶点附件按下，若是，则调用drag()方法，drag()方法中监听鼠标移动事件。每移动一帧就重新绘制一次图形。

相比于project1，这里还要将vettex\_pos2同时进行更改。通过调用changeCoordinate()方法，按照currentAngle等比例在原图上还原拖拽效果。

