

计算机图形学 Project2 说明文档

20302010008 林奕铨

项目目录及文件说明

<code>readme.pdf</code>	说明文档
<code>config.js</code>	配置文件 (与要求相同, 未做改动)
<code>Project2.html</code>	项目 HTML 文件 (仅包含网页框架)
<code>Project2.js</code>	项目 JavaScript 文件 包括: 类、方法等全部项目逻辑
<code>/lib/*.js</code>	库文件

开发及运行环境

开发环境: macOS Ventura, WebStorm, Safari 16.3

运行环境: Safari 16.3

运行及使用方法

运行方法: 使用支持 `HTML5` 的浏览器打开 `Project2.html` 即可。

使用方法: 按照网页提示, 按下[B]键来显示/隐藏边框; 按下[E]键来开启/关闭编辑模式; 按下[T]开始/停止动画。在编辑模式下, 使用鼠标拖动多边形的顶点, 查看多边形网络的动态绘制情况。

项目亮点

▼ 架构清晰、注释完整

本项目使用了 `class` 关键字, 以提供更直观的项目架构。

项目共有七个类, 分别为 `Point`, `Triangle`, `Shapes`, `Transform`, `Renderer`, `Controller`, `InteractionManager`。

类与类之间职责划分与主要成员和功能如下所示:

▼ Class Point

- ▶ 职责: 存储顶点坐标 `[x, y, z]` 与颜色 `[r, g, b]`

▼ Class Triangle

- ▶ 职责: 存储由三个顶点构成的三角形
- ▶ 成员: `point1, point2, point3`

▼ Class Shapes

- ▶ 职责: 存储顶点与三角形, 在渲染时提供坐标与颜色数据。
- ▶ 成员: `points` - Point对象构成的数组, 存储所有顶点。
- ▶ 成员: `triangles` - Triangle对象构成的数组, 引用Point对象。
- ▶ 方法: `addPolygon()` - 将四边形作为两个三角形添加到 `triangles` 中
- ▶ 方法: `getVerticesWithColors()` - 返回所有顶点的坐标及颜色, 以供渲染器使用。

▼ Class Transform

- ▶ 职责: 存储与更新变换矩阵, 在渲染时提供变换矩阵。
- ▶ 成员: `modelMatrix` - 变换矩阵
- ▶ 方法: `startTransform()` 和 `updateTransform()` - 根据时间戳, 更新变换矩阵。
- ▶ 方法: `get modelMatrixElements()` - 返回变换矩阵, 以供渲染器使用。

▼ Class Renderer

- ▶ 职责: 从Shapes类和Transform类获取数据进行渲染。
- ▶ 成员: `WebGLContext`, `Shaders`, `vertexColorBuffer` 等渲染相关环境、着色器、缓冲区、变量。
- ▶ 方法: `render()` - 从Shapes类中获取坐标与颜色数据、从Transform类中获取变换矩阵, 进行渲染。

▼ Class Controller

- ▶ 职责: 控制边框显示/隐藏、顶点拖动和变换动画。
- ▶ 成员: `isShowBorder` - 是否显示边框
- ▶ 方法: `render()` - 根据 `isShowBorder` 来调用渲染器
- ▶ 方法: `startAnimation`, `updateAnimation`, `stopAnimation` - 控制开始/更新/结束变换。
- ▶ 方法: `mouseDown`, `mousemove`, `mouseup` - 处理鼠标拖动事件

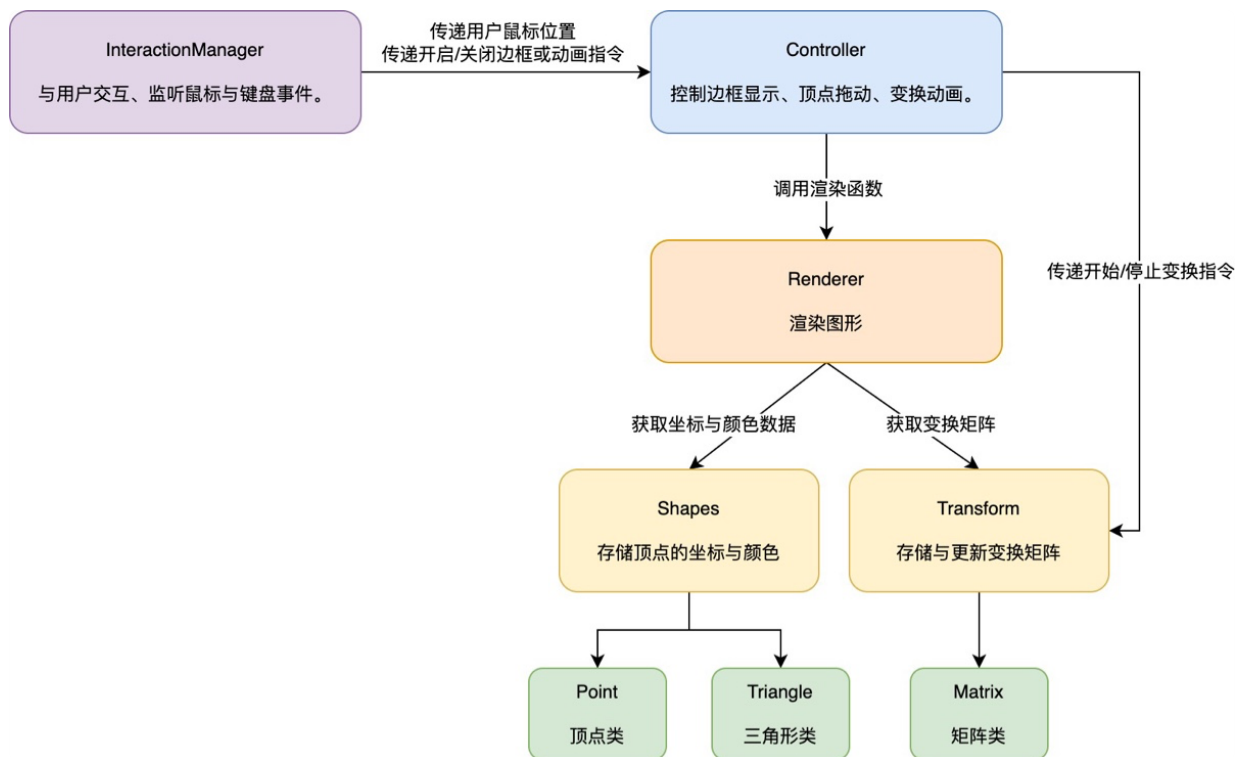
▼ Class InteractionManager

- ▶ 职责: 负责与用户交互, 监听键盘与鼠标事件。
- ▶ 通过调用Controller函数, 处理用户事件。

为了使源代码文件更清晰，项目使用了 `JSDoc` 为类和函数添加了注释，使阅读更为容易。

▼ 合理分配职责、避免超大类的出现

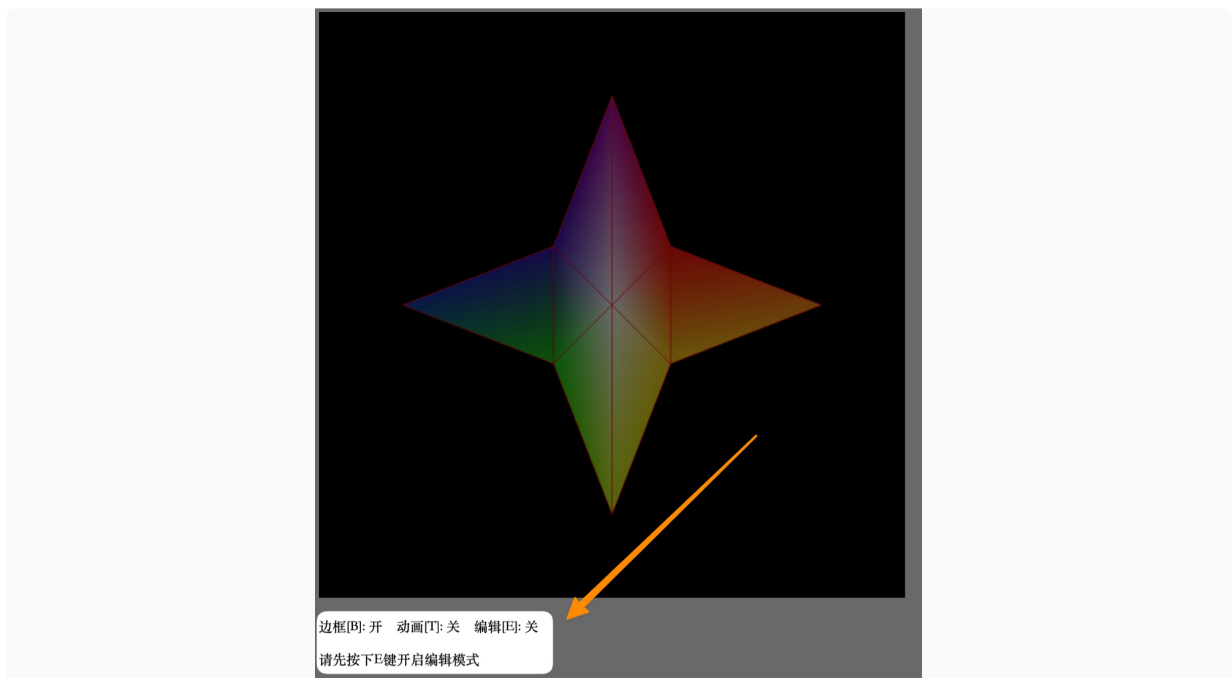
本次项目的大致结构如下：



将类按照职责与功能划分，遵循了“单一职责原则”，从而避免了超大类的出现。这样做的好处是，即使需要添加新功能也不会影响到其他部分的代码。例如，当需要添加新的变换效果时，只需要在 Transform 类中添加相应的方法即可，无需修改其他类。这也使得代码的维护更加便利。

▼ 用户界面指示当前状态

在 HTML 页面的 `canvas` 元素下方会显示当前的状态，包括是否显示边框、是否处在动画模式、是否处在编辑模式。用户可以直观地看到当前页面的状态，避免了因为不了解页面状态而产生的误操作。同时，若用户在动画模式或非编辑模式下尝试鼠标拖动操作，页面中也会产生提醒文字。



开发中遇到的问题及解决方法

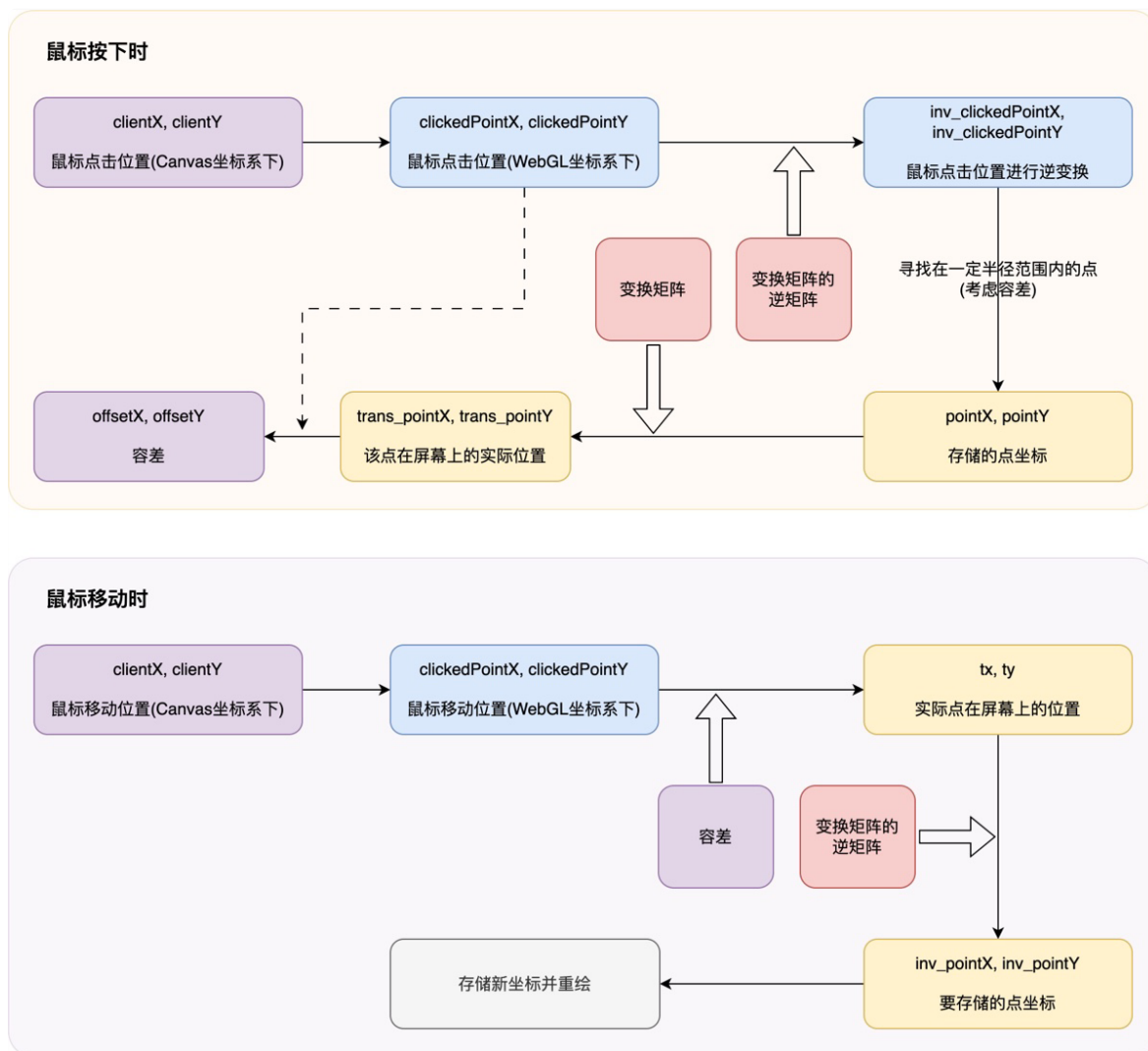
▼ 图形经变换后编辑的处理

▼ 问题描述

存储在 Shapes 中的顶点坐标向量与 Transform 变换矩阵相乘后，得到结果渲染在屏幕上。此时，显示在屏幕上的坐标与存储的坐标不一致。若要在屏幕上拖动顶点改变坐标，不仅要考虑鼠标点击位置与实际顶点位置之间的容差，还需要考虑坐标系的转换。该过程比较复杂。

▼ 问题解决

通过如下图所示的坐标变换过程，即可实现图形经变换后的编辑处理。



项目中可能存在的缺陷

在本项目的编辑模式中，拖动顶点的判定容差为固定值 0.1 (WebGL 坐标系下)。在图形未缩放时，判定范围较大，虽然对于用户的操作容错度较高，但当两个顶点很接近时可能会出现误判情况；在图形缩小后，判定范围过小，可能出现用户较难操作的情况。其中一个解决方案为动态调整判定容差的大小，使其与 `scale` 缩放比例成反相关。