

计算机图形学作业8

Basic

1. 添加消除控制点

在绘制Bezier曲线之前，首先要添加消除控制点，必须通过监听鼠标按键以及光标位置来实现。注册回调函数如下

```
glfwSetCursorPosCallback(window, mouse_callback);
glfwSetMouseButtonCallback(window, mouse_button_callback);
```

由于鼠标的两种信息分别用不同函数来获得，需要记录一种信息并在另一个回调函数中处理。可以选择实时更新光标位置，记录如下

```
void mouse_callback(GLFWwindow* window, double posX, double posY) {
    mouseX = posX;
    mouseY = posY;
}
```

然后按下鼠标左键时，添加一个新的控制点；按下鼠标右键时，消除最后一个控制点。功能实现如下

```
void mouse_button_callback(GLFWwindow* window, int button, int action, int mods) {
    if (pointRB == POINT_ENABLE && action == GLFW_PRESS && isValid(mouseX, mouseY)) {
        if (button == GLFW_MOUSE_BUTTON_LEFT) {
            double viewportX = mouseX / WINDOW_WIDTH * 2.0f - 1.0f;
            double viewportY = -mouseY / WINDOW_HEIGHT * 2.0f + 1.0f;
            if (points.size() < 50) {
                points.push_back(Point(viewportX, viewportY));
            }
        }
        else if (button == GLFW_MOUSE_BUTTON_RIGHT) {
            if (!points.empty()) {
                points.pop_back();
            }
        }
    }
}
```

需要注意的是，通过 `isValid` 来判断光标位置是否位于窗口内，只有当光标位于窗口内才处理鼠标点击事件。当控制点的数量超过 50 时，不再添加控制点；当没有控制点时，无法消除控制点。由于光标的坐标系与窗口的坐标系不一致，需要进行转换，光标的坐标系是从左上角 (0,0) 到右下角的窗口大小，窗口的坐标系是以中心点为 (0,0) 的平面直角坐标系，四角的坐标为 (± 2 , ± 2)。

在ImGui的Point菜单栏中选中 Disable 和 Enable 来控制能否添加消除控制点，可以通过键盘按键【1】和【2】来切换。实现添加消除控制点的完整代码见 [P1.cpp](#)，添加消除控制点的视频见 [P1.mp4](#)。

2. 实时更新Bezier曲线

记录控制点信息之后，就可以绘制Bezier曲线了。Bezier曲线上的每一个点都可以用控制点的线性组合来表示

$$Q(t) = \sum_i^n P_i B_{i,n}(t), t \in [0, 1]$$

其中 P_i 为控制点， $B_{i,n}(t)$ 为Bernstein函数，参数 t 决定了每个控制点的权重。Bernstein函数展开如下

$$B_{i,n}(t) = \frac{n!}{i!(n-i)!} t^i (1-t)^{n-i}, i = 0, 1, \dots, n$$

由于Bernstein函数涉及阶乘运算，需要一些小技巧来避免溢出。尽管阶乘很快就超过了long long的表示范围，但是long long可以表示所有 $n \leq 50$ 的组合数。在初始化的时候先计算组合数并保存在数组中，这样可以降低运算的时间复杂度。计算组合数的公式如下

$$\begin{aligned} C_n^0 &= C_n^n = 1 \\ C_n^i &= C_{n-1}^i + C_{n-1}^{i-1} \end{aligned}$$

虽然Bezier曲线是连续的，但我们只能计算离散的像素点，因此可能会造成曲线密度不均。按相同步长取1001个不同的 t 来表示曲线，在大部分情况下都足够表示了。

在ImGui的Bezier菜单栏中选中 `Invisible` 和 `Visible` 来控制是否绘制Bezier曲线，可以通过键盘按键【3】和【4】来切换。绘制Bezier曲线的相关函数见 `P2.cpp`，实时更新Bezier曲线的视频见 `P2.mp4`。

Bonus

1. 动态生成Bezier曲线

动态生成Bezier曲线就是将直线上移动的点以及对应的轨迹表示出来，第一层的点在控制点连接的直线上根据参数 t 移动，第二层的点在第一层点连接的直线上根据参数 τ 移动，以此类推。用时间作为参数 t ，以10秒为一个周期，移动8秒，停止2秒。

在ImGui的Animation菜单栏中选中 `Freeze` 和 `Move` 来控制是否动态生成Bezier曲线，可以通过键盘按键【5】和【6】来切换。动态生成Bezier曲线的相关函数见 `B1.cpp`，视频见 `B1.mp4`。