

计算机图形学PJ报告

16307130307 于泳欣

目录

- 音乐可视化：灯光效果
- 绘制静物：机器人
- flash制作：吃西瓜

音乐可视化

使用方法：HTML5 AudioContext

参考：

<https://developer.mozilla.org/en-US/docs/Web/API/WebAudioAPI>

1. AudioContext

AudioContext是用于音频处理的接口，工作原理是将AudioContext创建出的各种节点(AudioNode)相互连接，音频数据流经这些节点并作出相应处理。

- 创建AudioContext对象

```
window.AudioContext = window.AudioContext || window.webkitAudioContext || window.mozAudioContext || window.msAudioContext;  
  
var audioContext = new AudioContext();
```

- 创建AnalyserNode

```
var analyser = audioContext.createAnalyser();
```

- 截取音频信号

```
source = audioContext.createMediaElementSource(audio);  
source.connect(analyser);
```

- 声音连接到扬声器

```
analyser.connect(audioContext.destination);
```

2. 绘制可视化图像

- 得到音频能量值

```
var data = new Uint8Array(analyser.frequencyBinCount);  
analyser.getByteFrequencyData(data);
```

- 得到频谱

```
var playerTimeDomainData = new Uint8Array(analyser.fftSize);  
analyser.getByteTimeDomainData(playerTimeDomainData);
```

具体的canvas绘制逻辑参考网上已有代码，实现灯光效果。

绘制静物

使用方法：OpenGL

环境准备：Xcode中引用GLUT库

具体函数解析：

1. main函数

`glutInit(&argc, argv)`：对GLUT进行初始化。

`glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH)`：初始化显示模式

`glutCreateWindow()`：创建窗口

`glClearColor()`：设置背景色

`glEnable(GL_DEPTH_TEST)`：开启更新深度缓冲区的功能，也就是，如果通过比较后深度值发生了变化，会进行更新深度缓冲区的操作

`glShadeModel(GL_SMOOTH)`：平滑模式填充线段

`glMatrixMode(GL_PROJECTION)`：投影矩阵

`glLoadIdentity()`：变换前将矩阵设置为单位矩阵

`glOrtho(-100, 100, -100, 100, -100, 100)`：平行投影

`glutSpecialFunc(SpecialKeys)`：注册功能键回调函数

`glutDisplayFunc(drawSnowMan)` : 绘制图形

`glutReshapeFunc(changeSize)` : 改变大小

`glutMainLoop()` : 循环

2. void drawSnowMan()

从头开始画机器人，包括头，大臂，小臂，手，大腿，小腿，都是立方体

`glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)` : 清除颜色缓冲以及深度缓冲

`glColor3f()` : 设置颜色

`glTranslatef()` : 改变位置

`glutSolidSphere(0.75f, 20, 20)` : GLUT库中内置的画球的函数，三个参数分别代表

`glPushMatrix()` : 对于矩阵的操作都是对于矩阵栈的栈顶来操作的。当前矩阵即为矩阵栈的栈顶元素，而对当前矩阵进行平移、旋转等的变换操作也同样是对栈顶矩阵的修改。所以我们在变换之前调用`glPushMatrix()`的话，就会把当前状态压入第二层。

`glPopMatrix()` : 当经过一系列的变换后，栈顶矩阵被修改，此时调用`glPopMatrix()`时，栈顶矩阵被弹出，且又会恢复为原来的状态。

`glutSwapBuffers()` : 实现双缓冲技术，当绘制完成时，把绘制的最终结果复制到屏幕上。

所谓双缓冲技术，是指使用两个缓冲区：前台缓冲和后台缓冲。前台缓冲即我们看到的屏幕，后台缓冲则在内存当中，对我们来说是不可见的。每次的所有绘图操作都在后台缓冲中进行，当绘制完成时，把绘制的最终结果复制到屏幕上，这样，我们看到所有GDI元素同时出现在屏幕上，从而解决了频繁刷新导致的画面闪烁问题。

3. void changeSize(int w, int h)

`changeSize`函数的作用是作为`glutReshapeFunc`函数的参数改变图形大小。

`glViewport(0, 0, w, h)` : 把视景体截取的图像按照怎样的高和宽显示到屏幕上。

`void gluPerspective (GLdouble fovy, GLdouble aspect, GLdouble zNear, GLdouble zFar)` : `fovy`是眼睛上下睁开的幅度，角度值，值越小，视野范围越狭小，值越大，视野范围越宽阔（；`zNear`表示近裁剪面到眼睛的距离，`zFar`表示远裁剪面到眼睛的距离，注意`zNear`和`zFar`不能设置为负值。`aspect`表示裁剪面的宽`w`高`h`比，这个影响到视野的截面有多大。

`gluLookAt(x, y, z, x + lx, y + ly, z + lz, 0.0f, 1.0f, 0.0f)` : 从哪里看

Flash动画制作

环境：Adobe Animate 2019

学习过程

1. 了解使用Animate制作flash的过程

2. 熟悉Animate中各工具的用法

基本概念

图层：可以在一个图层上绘制和编辑对象，而不会影响其他图层上的对象。

帧：在时间轴上用帧来组织和控制文档的内容，帧的顺序决定了显示顺序。

关键帧：角色或者物体运动或变化中的关键动作所处的那一帧

补间动画：两个关键帧之间使用补间动画实现动画效果

制作步骤

- 引入图片，注意图层的分布
- 插入关键帧，在关键帧上对图片进行修改
- 在关键帧之间插入普通帧，实现延时效果
- 调试影片，测试播放效果

参考资料

<https://helpx.adobe.com/cn/animate/using/time.html> <https://helpx.adobe.com/cn/animate/using/visual-glossary.html>