

## 综合测试题 2 参考答案

### 一、 选择题（3 分×8 题=24 分）

题号	1	2	3	4	5	6	7	8
答案	CD	A	BD	B	CD	ABD	B	ACD

### 二、 判断题（2 分×6 题=12 分）

题号	1	2	3	4	5	6
答案	√	√	×	×	√	√

### 三、 填空题（2 分×8 空=16 分）

- 1、齐次向量（齐次坐标）
- 2、DXGI\_FORMAT\_R8G8B8A8\_SINT
- 3、Wrap, Border Color, Clamp, Mirror（写出任意三个即可得分）
- 4、8
- 5、MIN\_LINEAR\_MAG\_MIP\_POINT
- 6、材质

### 四、 名词解释（5 分×2=10 分）

纹理过滤器：当纹理图片放大或者缩小时，会发生形变。纹理过滤器通过插值的方式使这种形变变得平滑。

顶点输入布局：一个顶点可以包含多种信息，顶点输入布局用于描述一个顶点中具体包括哪些信息，从而确定一个顶点以及顶点的不同信息在缓存中的大小和位置。

### 五、 简答题（答中要点即可）（7 分×3 题=21 分）

- 1、点光源：该光源在世界坐标系中有固定的位置，并向所有的方向发射射线  
方向光：该光源没有位置信息，所发射的光线相互平行地沿着某一特定的方向传播  
聚光灯：与手电筒类似，该光源有位置信息，其发射的光线呈锥形沿着特定方向传播  
不同之处：三种光线在有无光源位置信息，所发出的光的传播方向上的不同
- 2、用户或者应用程序的某些行为会产生一些事件，操作系统找到事件所属的应用程序发送一条相应的消息，该消息被加入到该应用程序的消息队列中，之后应用程序不断地检查消息队列，每当接收到一条消息时，应用程序就将该消息分发给所属该窗口的相关的窗口过程，最后，窗口过程的执行与之前的消息对应的指令。

- 3、输入装配阶段
  - 顶点着色器阶段
  - 外壳着色器阶段
  - 曲面细分阶段
  - 域着色阶段
  - 几何着色阶段
  - 光栅化阶段
  - 像素着色阶段
  - 输出合并阶段

## 六、 综合题（共 17 分）

(1)

//标记镜子区域

D3D11\_DEPTH\_STENCIL\_DESC DSSDesc; //深度模板描述

DSSDesc.DepthEnable = true; //开启深度测试

DSSDesc.DepthFunc = D3D11\_COMPARISON\_LESS; //深度测试比较为

“小于”

DSSDesc.DepthWriteMask = D3D11\_DEPTH\_WRITE\_MASK\_ZERO; //禁止将深度值写入  
深度模板缓存

DSSDesc.StencilEnable = true; //开启模板测试

DSSDesc.StencilReadMask = D3D11\_DEFAULT\_STENCIL\_READ\_MASK; //设置默认读  
掩码

DSSDesc.StencilWriteMask = D3D11\_DEFAULT\_STENCIL\_WRITE\_MASK; //设置默认写  
掩码

DSSDesc.FrontFace.StencilFunc = D3D11\_COMPARISON\_ALWAYS; //通过模板测试的  
条件为“总是”

DSSDesc.FrontFace.StencilPassOp = D3D11\_STENCIL\_OP\_REPLACE; //通过测试进行替  
换操作

DSSDesc.FrontFace.StencilDepthFailOp = D3D11\_STENCIL\_OP\_KEEP; //模板通过，深度失  
败则保持不变

DSSDesc.FrontFace.StencilFailOp = D3D11\_STENCIL\_OP\_KEEP; //模板测试失败则  
保持不变

DSSDesc.BackFace.StencilFunc = D3D11\_COMPARISON\_ALWAYS; //通过模板测  
试的条件为“总是”

DSSDesc.BackFace.StencilPassOp = D3D11\_STENCIL\_OP\_REPLACE; //通过测试进行  
替换操作

DSSDesc.BackFace.StencilDepthFailOp = D3D11\_STENCIL\_OP\_KEEP; //测试通过，深度失败则保持不变

DSSDesc.BackFace.StencilFailOp = D3D11\_STENCIL\_OP\_KEEP; //模板测试失败则保持不变

```
if(FAILED(device->CreateDepthStencilState(&markMirrorDSSDesc,& markDSS)))
```

```
{
```

```
    MessageBox(NULL,L"Create 'MarkMirror' depth stencil state failed!",L"Error",MB_OK);
```

```
    return false;
```

```
}
```

(2) 需要写入的代码如下：

绘制之前：immediateContext->OMSetDepthStencilState(markMirrorDSS,0x1); //设置好相应的模板缓冲区状态

绘制之后：immediateContext->OMSetDepthStencilState(NULL,0x1);

(3) 代码生效后，会将物体像素点对应的所有深度模板缓存标记为 1。