## 使用缓冲区对象-绘制多个点

#### 1.0 什么是缓冲区对象

◆ 缓冲区对象是WebGL系统中的一块内存区域,可以一次性地向缓冲区对象中填充大量的顶点数据,然后将这些数据保存在其中,供顶点着色器使用。

#### 1.1 创建顶点数据

```
const points = new Float32Array([
  -0.5, -0.5,
  0.5, -0.5,
  0.0, 0.5,
1)
```

### 1.1 类型化数组 - Float32Array

◆ 在 webgl 中,需要处理大量的相同类型数据,所以引入类型化数组,这样程序就可以预知到数组中的数据类型, 提高性能。

### 1.1 类型化数组 - Float32Array

Int8Array: 8位整型

UInt8Array: 8位无符号整型

Int16Array: 16位整型

UInt16Array:16位无符号整型

类型化数组类型

#### 1.1 类型化数组 - Float32Array

Int32Array:32位整型

UInt32Array:32位无符号整型

Float32Array:单精度32位浮点型

Float64Array:双精度64位浮点型

类型化数组类型

#### 1.2 创建缓冲区对象

```
const buffer = gl.createBuffer();
```

### 1.3 gl.bindBuffer(target, buffer)

◆ buffer: 已经创建好的缓冲区对象

◆ target: 可以是如下两种

#### 1.3 gl.bindBuffer(target, buffer)

◆ gl.ARRAY BUFFER: 表示缓冲区存储的是顶点的数据

◆ gl.ELEMENT\_ARRAY\_BUFFER: 表示缓冲区存储的是顶点的索引值

### 1.4 gl.bufferData(target, data, type)

◆ target: 类型同 gl.bindBuffer 中的 target

◆ data: 写入缓冲区的顶点数据,如程序中的 points

### 1.4 gl.bufferData(target, data, type)

◆ type: 表示如何使用缓冲区对象中的数据,分为以下几类

gl.STATIC\_DRAW: 写入一次,多次绘制

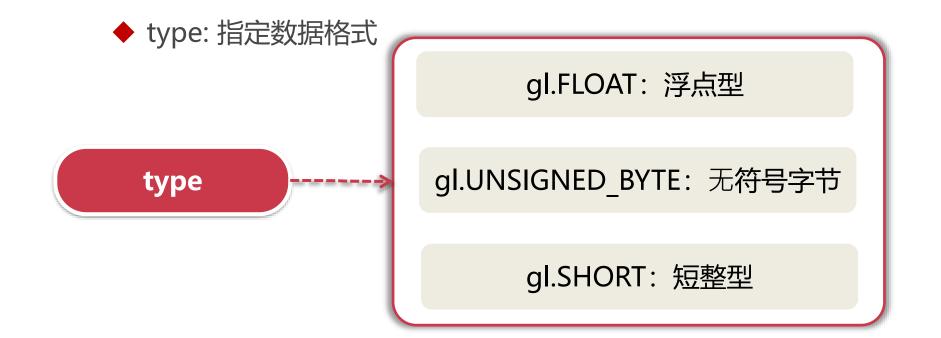
gl.STREAM\_DRAW:写入一次,绘制若干次

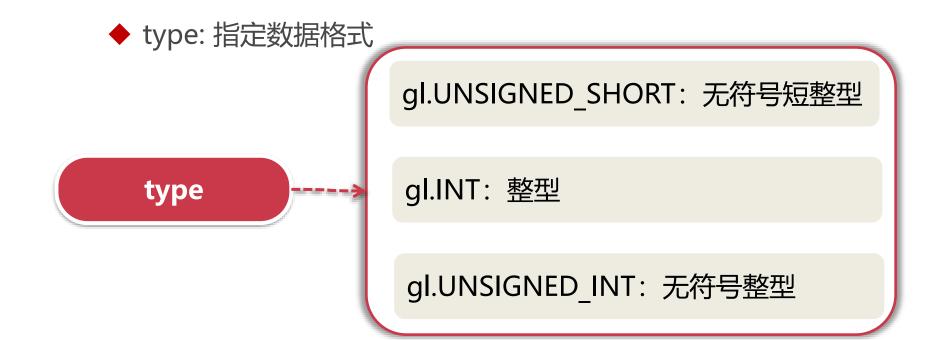
gl.DYNAMIC\_DRAW: 写入多次, 绘制多次

type -

◆ location: attribute 变量的存储位置

◆ size: 指定每个顶点所使用数据的个数





◆ normalized: 表示是否将数据归一化到 [0, 1] [-1, 1] 这个区间

◆ stride: 两个相邻顶点之间的字节数

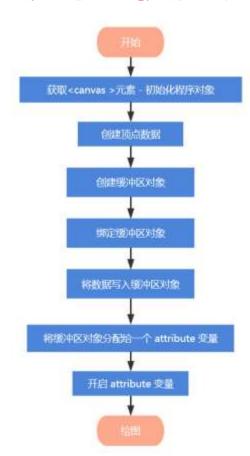
◆ offset: 数据偏移量

### 1.6 gl.enableVertexAttribArray(location)

◆ location: attribute 变量的存储地址

◆ gl.disableVertexAttribArray(aPosition); 使用此方法禁用

### 2. 缓冲区使用流程



#### 3. 缓冲区执行过程

