

# Several Attributes

## Lecture 5

Robb T. Koether

Hampden-Sydney College

Wed, Sep 4, 2019

# Outline

## 1 Color

## 2 Coloring a Rectangle

- One Array, Segregated Attributes
- Two Arrays, Segregated Attributes
- One Array, Integrated Attributes
- One Array, Structured Data

## 3 Assignment

# Outline

## 1 Color

## 2 Coloring a Rectangle

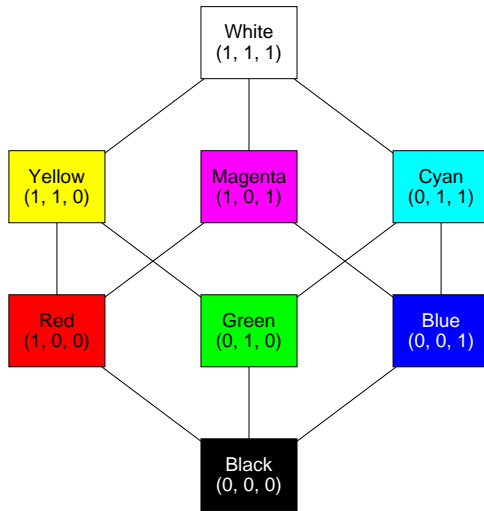
- One Array, Segregated Attributes
- Two Arrays, Segregated Attributes
- One Array, Integrated Attributes
- One Array, Structured Data

## 3 Assignment

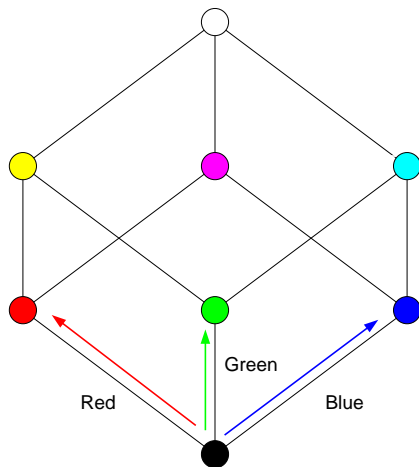
# Color

- In computer graphics, every color has three components.
  - Red
  - Green
  - Blue
- Any specific color is represented by a triple  $(r, g, b)$ , with each component between 0.0 and 1.0.
- The RGB values are **clamped** to the range  $[0, 1]$ .
- On the graphics card, they are stored as **unsigned ints**, from 0 to 255.

# Color



# Color



- What RGB triple would appear gray?

# Color

- What RGB triple would appear gray?
- Dark gray?



# Color

- What RGB triple would appear gray?
- Dark gray?
- Yellow?

# Color

- What RGB triple would appear gray?
- Dark gray?
- Yellow?
- Light yellow?

# Color

- What RGB triple would appear gray?
- Dark gray?
- Yellow?
- Light yellow?
- Pink?

# Color

- What RGB triple would appear gray?
- Dark gray?
- Yellow?
- Light yellow?
- Pink?
- Orange?

# Color

- What RGB triple would appear gray?
- Dark gray?
- Yellow?
- Light yellow?
- Pink?
- Orange?
- Brown?

# Outline

## 1 Color

## 2 Coloring a Rectangle

- One Array, Segregated Attributes
- Two Arrays, Segregated Attributes
- One Array, Integrated Attributes
- One Array, Structured Data

## 3 Assignment

# Coloring a Rectangle

- To color a rectangle, we need to include the color data in the buffer along with the vertex coordinates.
- There are several ways to do this.

# Outline

## 1 Color

## 2 Coloring a Rectangle

- One Array, Segregated Attributes
- Two Arrays, Segregated Attributes
- One Array, Integrated Attributes
- One Array, Structured Data

## 3 Assignment



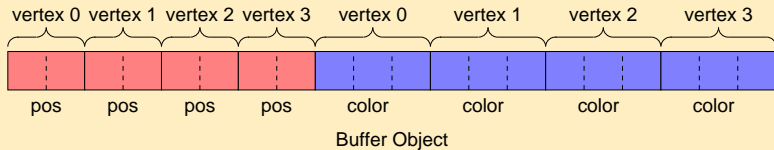
# One Array, Segregated Attributes

## One Array, Segregated Attributes

```
GLfloat rect_data[] =  
{  
    -0.5f, -0.5f,          // 1st vertex  
     0.5f, -0.5f,          // 2nd vertex  
     0.5f,  0.5f,          // 3rd vertex  
    -0.5f,  0.5f,          // 4th vertex  
     1.0f, 0.0f, 0.0f,     // Color of 1st  
     1.0f, 1.0f, 0.0f,     // Color of 2nd  
     0.0f, 1.0f, 0.0f,     // Color of 3rd  
     0.0f, 0.0f, 1.0f     // Color of 4th  
};
```

- We can pack all the data contiguously into one array, with the attributes segregated.

## One Array, Segregated Attributes



# Color a Rectangle

## Color a Rectangle

```
enum {vPosition = 0, vColor = 1};
```

- Create a symbolic name for the color attribute.
- This appears in the vertex shader as the “location.”

# Color a Rectangle

## Color a Rectangle

```
glNamedBufferStorage(VBO[RectBuffer], sizeof(rect_data),  
    rect_data, 0);
```

- Store the data in the buffer and bind the vertex array object, as before.

# Color a Rectangle

## Color a Rectangle

```
glBindVertexArray(VAOs[Rect]);  
glVertexAttribPointer(vPosition, 2, GL_FLOAT, GL_FALSE,  
    0, BUFFER_OFFSET(0));  
glVertexAttribPointer(vColor, 3, GL_FLOAT, GL_FALSE,  
    0, BUFFER_OFFSET(8*sizeof(GLfloat)));
```

- Set the position attribute as before.
- Give the color attribute an offset equal to the size of the position data.
- Both attributes have a stride of 0.

# Outline

## 1 Color

## 2 Coloring a Rectangle

- One Array, Segregated Attributes
- **Two Arrays, Segregated Attributes**
- One Array, Integrated Attributes
- One Array, Structured Data

## 3 Assignment

# Two Arrays, Segregated Attributes

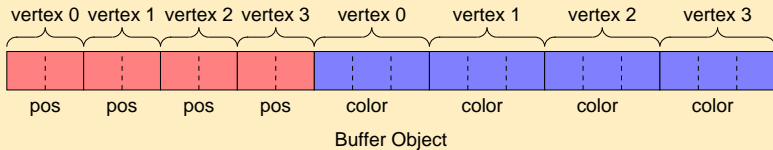
## Two Arrays, Segregated Attributes

```
GLfloat rect_pos[] =
{
    -0.5f, -0.5f,      // 1st vertex
     0.5f, -0.5f,      // 2nd vertex
     0.5f,  0.5f,      // 3rd vertex
    -0.5f,  0.5f      // 4th vertex
};

GLfloat rect_color[] =
{
    1.0f, 0.0f, 0.0f,  // Color of 1st
    1.0f, 1.0f, 0.0f,  // Color of 2nd
    0.0f, 1.0f, 0.0f,  // Color of 3rd
    0.0f, 0.0f, 1.0f   // Color of 4th
};
```

- We can create two separate arrays, with the attributes necessarily segregated.

## One Array, Segregated Attributes





# Two Arrays, Segregated Attributes

## Two Arrays, Segregated Attributes

```
glNamedBufferStorage(VBO[RectBuffer], sizeof(rect_pos)
    + sizeof(rect_color), NULL, 0);
glNamedBufferSubData(VBO[RectBuffer], 0, sizeof(rect_pos),
    rect_pos);
glNamedBufferSubData(VBO[RectBuffer], sizeof(rect_pos),
    sizeof(rect_color), rect_color);
```

- We must first reserve the memory and then separately store the two arrays using `glNamedBufferSubData()`.

# Two Arrays, Segregated Attributes

## Two Arrays, Segregated Attributes

```
glBindVertexArray(VAOs[Rect]);  
glVertexAttribPointer(vPosition, 2, GL_FLOAT, GL_FALSE,  
    0, BUFFER_OFFSET(0));  
glVertexAttribPointer(vColor, 3, GL_FLOAT, GL_FALSE,  
    0, BUFFER_OFFSET(sizeof(rect_pos)));
```

- Set the position attribute as before.
- Give the color attribute an offset equal to the size of the position data.

# Outline

## 1 Color

## 2 Coloring a Rectangle

- One Array, Segregated Attributes
- Two Arrays, Segregated Attributes
- **One Array, Integrated Attributes**
- One Array, Structured Data

## 3 Assignment

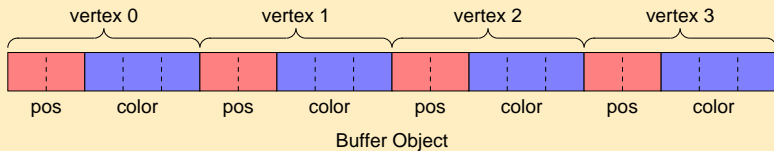
# One Array, Integrated Attributes

## One Array, Integrated Attributes

```
GLfloat rect_data[] =  
{  
    -0.5f, -0.5f, 1.0f, 0.0f, 0.0f, // 1st vertex  
     0.5f, -0.5f, 1.0f, 1.0f, 0.0f, // 2nd vertex  
     0.5f,  0.5f, 0.0f, 1.0f, 0.0f, // 3rd vertex  
    -0.5f,  0.5f, 0.0f, 0.0f, 1.0f  // 4th vertex  
};
```

- We can create one array, with the attributes integrated.

## One Array, Integrated Attributes



# One Array, Integrated Attributes

## One Array, Integrated Attributes

```
glNamedBufferStorage(VBO[RectBuffer], sizeof(rect_data),  
rect_data, 0);
```

- Store the data in the buffer and bind the vertex array object, as before.

# One Array, Integrated Attributes

## One Array, Integrated Attributes

```
glBindVertexArray(VAOs[Rect]);  
glVertexAttribPointer(vPosition, 2, GL_FLOAT, GL_FALSE,  
    5*sizeof(GL_FLOAT), BUFFER_OFFSET(0));  
glVertexAttribPointer(vColor, 3, GL_FLOAT, GL_FALSE,  
    5*sizeof(GL_FLOAT), BUFFER_OFFSET(2*sizeof(GLfloat)));
```

- Set the position attribute as before.
- Give the color attribute an offset equal to the size of a position.
- Give the position and color a stride equal to the size of the data for a vertex.

# Outline

## 1 Color

## 2 Coloring a Rectangle

- One Array, Segregated Attributes
- Two Arrays, Segregated Attributes
- One Array, Integrated Attributes
- **One Array, Structured Data**

## 3 Assignment



# One Array, Structured Data

## One Array, Structured Data

```
struct VertexData2D
{
    GL_FLOAT pos[2];
    GL_FLOAT color[3];
};
```

- Create a VertexData2D structure.

# One Array, Structured Data

## One Array, Structured Data

```
struct VertexData2D
{
    vec2 pos;
    vec3 color;
};
```

- Create a VertexData2D structure.

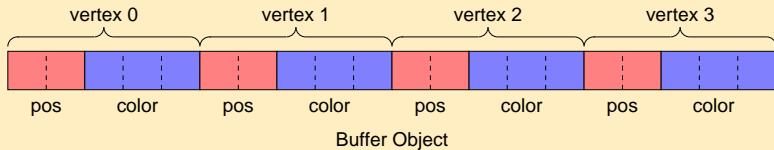
# One Array, Structured Data

## One Array, Structured Data

```
VertexData2D rect_data[] =  
{  
    {{-0.5f, -0.5f}, {1.0f, 0.0f, 0.0f}}, // 1st vertex  
    {{ 0.5f, -0.5f}, {1.0f, 1.0f, 0.0f}}, // 2nd vertex  
    {{ 0.5f,  0.5f}, {0.0f, 1.0f, 0.0f}}, // 3rd vertex  
    {{-0.5f,  0.5f}, {0.0f, 0.0f, 1.0f}} // 4th vertex  
};
```

- We can create one array of type `VertexData2D`.

## One Array, Structured Data



# One Array, Integrated Attributes

## One Array, Integrated Attributes

```
glNamedBufferStorage(GL_ARRAY_BUFFER, sizeof(rect_data),  
rect_data, 0);
```

- Store the data in the buffer and bind the vertex array object, as before.

# One Array, Integrated Attributes

## One Array, Integrated Attributes

```
glBindVertexArray(VAOs[Rect]);  
glVertexAttribPointer(vPosition, 2, GL_FLOAT, GL_FALSE,  
    sizeof(VertexData2D), BUFFER_OFFSET(0));  
glVertexAttribPointer(vColor, 3, GL_FLOAT, GL_FALSE,  
    sizeof(VertexData2D),  
    BUFFER_OFFSET(sizeof(vec2)));
```

- Set the position attribute as before.
- Give the color attribute an offset equal to the size of a position.
- Give the position a stride equal to the size of a color.
- Give the color a stride equal to the size of a position.

# Outline

## 1 Color

## 2 Coloring a Rectangle

- One Array, Segregated Attributes
- Two Arrays, Segregated Attributes
- One Array, Integrated Attributes
- One Array, Structured Data

## 3 Assignment

# Assignment

## Assignment

- Read pp. 16 - 22 in The Red Book.