Лекция 5. Буфер кадра

Илья Макаров

ИТМО

12 октября 2022

Санкт-Петербург

Буфер кадра

Все операции требуют **frame buffer**. Этот буфер может содержать несколько компонент (**attachments**).

- **color** - цвет;
- **depth** - глубина;
- **stencil** - трафарет.

FBO

Создание и использование буфера:

```
int fbo = -1;
glGenFramebuffers(1, &fbo);

glBindFramebuffer(GL_FRAMEBUFFER, fbo);

// Do smth

glBindFramebuffer(GL_FRAMEBUFFER, 0);
glDeleteFramebuffers(1, &fbo);
```

Бывают другие типы FBO (read-only, write-only, read-write).

FBO

Для использования буфера:

- Должен быть подключен как минимум один буфер (цвета, глубины или трафарета).
- Должно присутствовать хотя бы одно прикрепление цвета (color attachment).
- Все подключения также должны быть завершенными (обеспечены выделенной памятью).
- Каждый буфер должен иметь одинаковое количество семплов.

Для проверки статуса FBO используют следующую конструкцию:

```
glCheckFramebufferStatus(GL_FRAMEBUFFER)
    == GL_FRAMEBUFFER_COMPLETE
```

Attachments

Типы прикреплений.

- **texture** - текстура;
- **renderbuffer** - рендер-буфер.

Texture attachment

Создаем текстуру:

```
int texture;
glGenTextures(1, &texture);
glBindTexture(GL_TEXTURE_2D, texture);
glTexImage2D(GL_TEXTURE_2D, 0, GL_RGB,
    800, 600, 0, GL_RGB, GL_UNSIGNED_BYTE, NULL);
glTexParameteri(GL_TEXTURE_2D,
    GL_TEXTURE_MIN_FILTER, GL_LINEAR);
glTexParameteri(GL_TEXTURE_2D,
    GL_TEXTURE_MAG_FILTER, GL_LINEAR);
```

Прикрепляем текстуру:

```
glFramebufferTexture2D(GL_FRAMEBUFFER,
    GL_COLOR_ATTACHMENT0, GL_TEXTURE_2D, texture, 0);
```

Attachments

Можно прикреплять несколько буферов цвета.

- **GL_COLOR_ATTACHMENT0**;
- **GL_COLOR_ATTACHMENT1**;
- **...**

Аналогичным образом можно прикреплять текстуры для буферов глубины и трафарета.

- **GL_DEPTH_ATTACHMENT**;
- **GL_STENCIL_ATTACHMENT**;
- **GL_DEPTH_STENCIL_ATTACHMENT**;

Texture attachment

Пример комбинированного буфера:

```
glTexImage2D(
    GL_TEXTURE_2D, 0, GL_DEPTH24_STENCIL8,
    800, 600, 0, GL_DEPTH_STENCIL,
    GL_UNSIGNED_INT_24_8, NULL);

glFramebufferTexture2D(GL_FRAMEBUFFER,
    GL_DEPTH_STENCIL_ATTACHMENT,
    GL_TEXTURE_2D, texture, 0);
```

Renderbuffer attachment

Создание и установка:

```
int rbo;
glGenRenderbuffers(1, &rbo);
glBindRenderbuffer(GL_RENDERBUFFER, rbo);
glRenderbufferStorage(GL_RENDERBUFFER,
    GL_DEPTH24_STENCIL8, 800, 600);
```

Подключение:

```
glFramebufferRenderbuffer(GL_FRAMEBUFFER,
    GL_DEPTH_STENCIL_ATTACHMENT, GL_RENDERBUFFER, rbo);
```