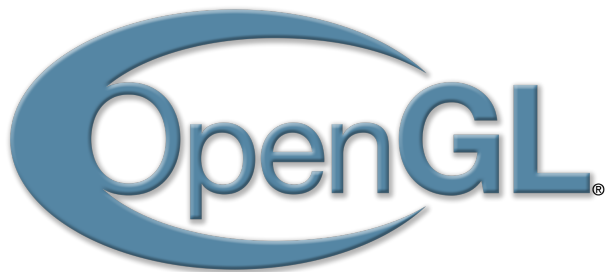


# OpenGL

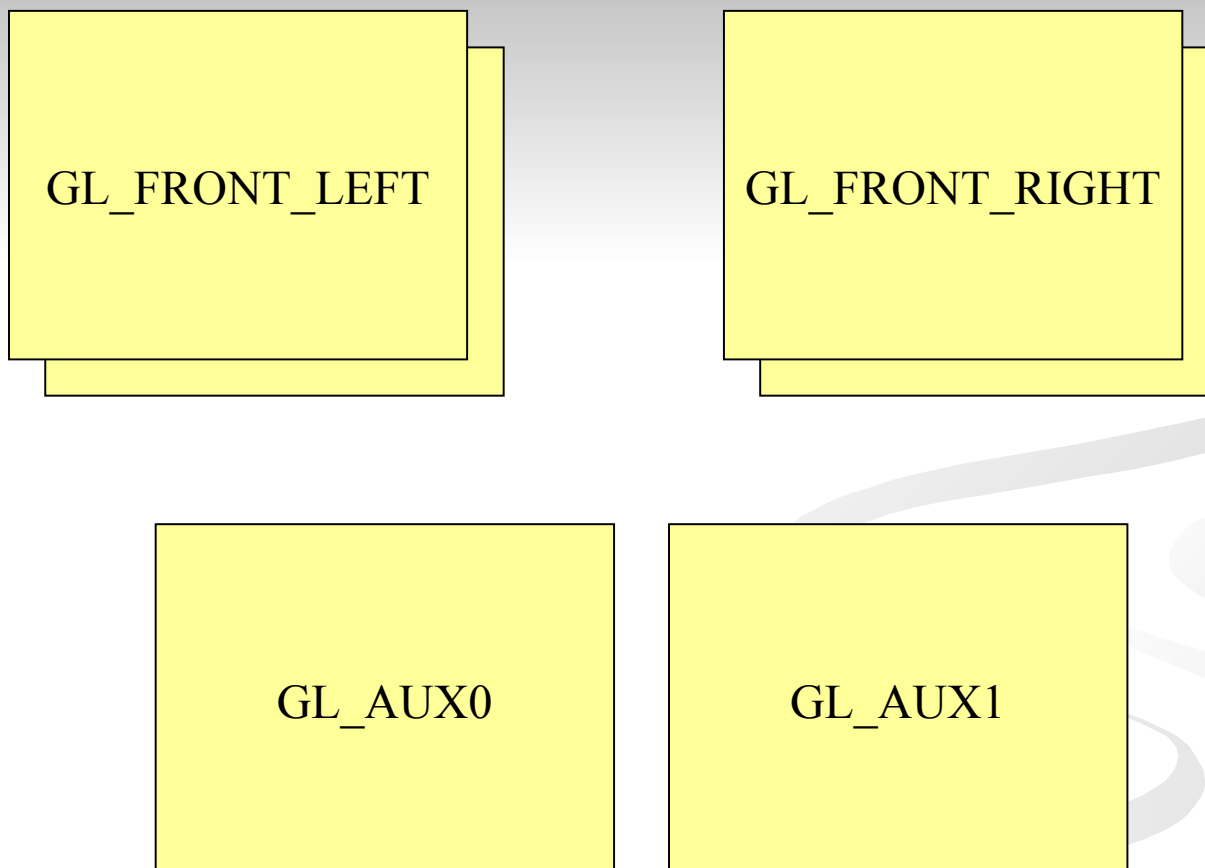
Буфери



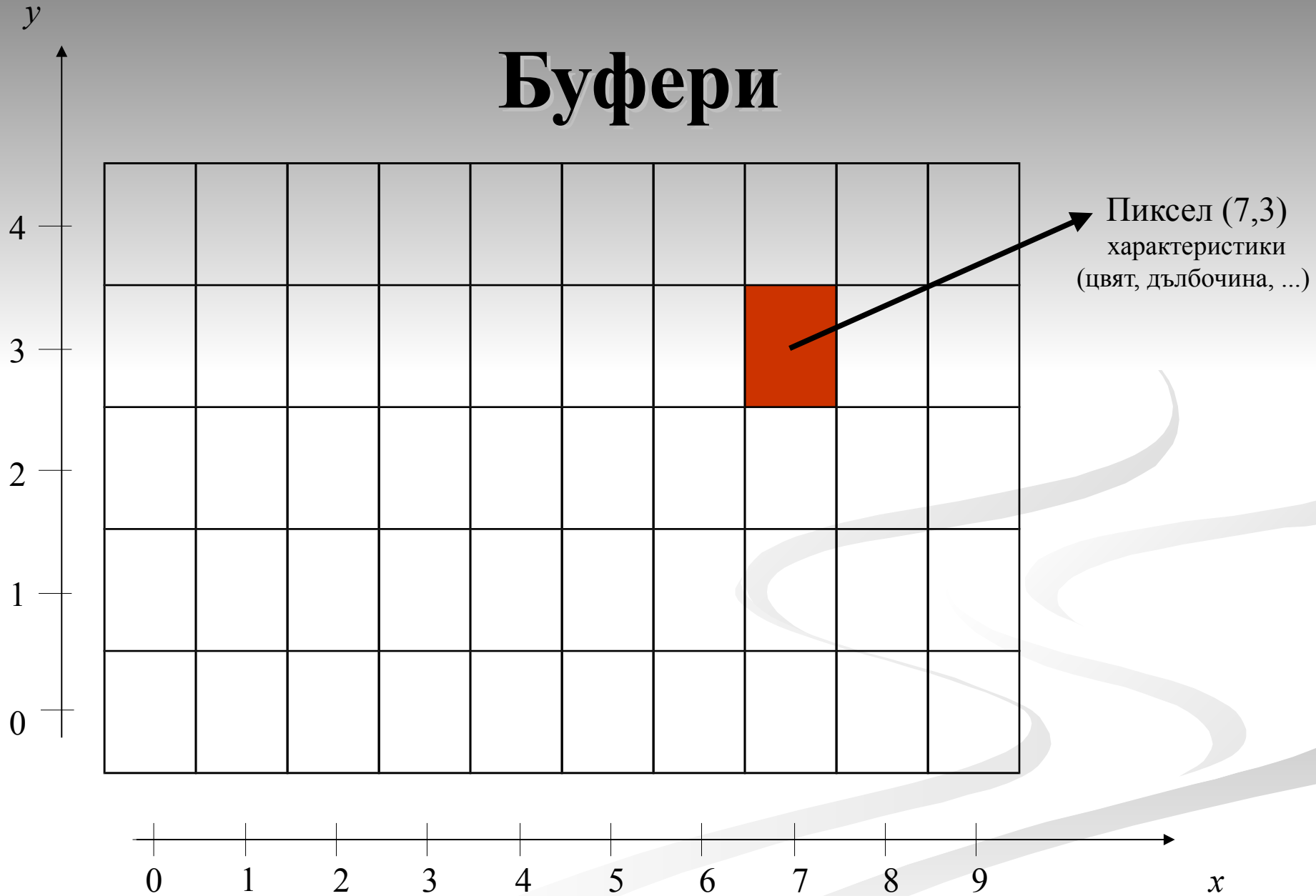
# Буфери в OpenGL

- Буфер на цвета (Color buffer):
  - Преден ляв (Front-left);
  - Преден десен (Front-right);
  - Заден ляв (Back-left);
  - Заден десен (Back-right);
  - Допълнителни (Auxiliary);
- Буфер на дълбочината (Depth buffer);
- Буфер на шаблона (Stencil buffer);
- Буфер акумулатор (Accumulation buffer).

# Буфери



# Буфери



# Избор на активен буфер

```
void glDrawBuffer(GLenum mode)  
void glReadBuffer(GLenum mode)
```

mode :

```
GL_NONE  
GL_FRONT  
GL_BACK  
GL_FRONT_AND_BACK  
GL_LEFT  
GL_RIGHT  
GL_FRONT_LEFT  
GL_FRONT_RIGHT  
GL_BACK_LEFT  
GL_BACK_RIGHT  
GL_AUXi
```

# Изчистване на буферите

```
void glClear(GLbitfield mask)
```

mask :

<code>GL_COLOR_BUFFER_BIT</code>	ЦВЯТ
<code>GL_DEPTH_BUFFER_BIT</code>	ДЪЛБОЧИНА
<code>GL_STENCIL_BUFFER_BIT</code>	ШАБЛОН
<code>GL_ACCUM_BUFFER_BIT</code>	АКУМУЛАТОР

# Буфер на цвета – изчистване

```
void glClearColor(  
    GLfloat r, GLfloat g,  
    GLfloat b, GLfloat a)
```

```
void glClearIndex(GLfloat c)
```

Пример:

```
glClearColor(0,0,0,0);
```

```
glClear(GL_COLOR_BUFFER_BIT);
```

# Буфер на цвета – маски

```
void glColorMask(  
    GLboolean red, GLboolean green,  
    GLboolean blue, GLboolean alpha)  
void glIndexMask(GLuint mask)
```

Пример:

```
glColorMask(GL_TRUE, GL_FALSE,  
    GL_FALSE, GL_FALSE);  
glColor(1,1,1,0);  
glSolidSphere(1,32,32);
```



# Буфер на цвета – смесване при прозрачност

```
void glBlendFunc(GLenum sfactor, GLenum dfactor)  
glEnable(GL_BLEND)
```

**sfactor:**

```
GL_ZERO, GL_ONE, GL_SRC_COLOR, GL_ONE_MINUS_SRC_COLOR,  
GL_DST_COLOR, GL_ONE_MINUS_DST_COLOR, GL_SRC_ALPHA,  
GL_ONE_MINUS_SRC_ALPHA, GL_DST_ALPHA, GL_ONE_MINUS_DST_ALPHA,  
GL_CONSTANT_COLOR, GL_ONE_MINUS_CONSTANT_COLOR,  
GL_CONSTANT_ALPHA, GL_ONE_MINUS_CONSTANT_ALPHA,  
GL_SRC_ALPHA_SATURATE
```

**dfactor:**

```
GL_ZERO, GL_ONE, GL_SRC_COLOR, GL_ONE_MINUS_SRC_COLOR,  
GL_DST_COLOR, GL_ONE_MINUS_DST_COLOR, GL_SRC_ALPHA,  
GL_ONE_MINUS_SRC_ALPHA, GL_DST_ALPHA, GL_ONE_MINUS_DST_ALPHA.  
GL_CONSTANT_COLOR, GL_ONE_MINUS_CONSTANT_COLOR,  
GL_CONSTANT_ALPHA, GL_ONE_MINUS_CONSTANT_ALPHA
```

# Буфер на цвета – смесване при прозрачност (Пример)

```
glEnable(GL_BLEND);  
glBlendFunc(GL_SRC_ALPHA,  
            GL_ONE_MINUS_SRC_ALPHA);  
  
glColor4d(1, 0, 0, 0.5);  
glutSolidSphere(1, 32, 32);  
glTranslated(0.5, 0, 0);  
glColor4d(0, 1, 0, 0.5);  
glutSolidSphere(1, 32, 32);
```

# Операции прилагани върху фрагментите

- Scissor тест;
- Alpha тест;
- Stencil тест;
- Depth тест;
- Blending;
- Dithering;
- Logical операции.

# Буфер на цвета – маска по прозрачност

```
void glAlphaFunc(GLenum func, GLclampf ref)  
glEnable(GL_ALPHA_TEST)
```

**func:**

**GL\_NEVER**

**GL\_LESS**

**GL\_EQUAL**

**GL\_LEQUAL**

**GL\_GREATER**

**GL\_NOTEQUAL**

**GL\_GEQUAL**

**GL\_ALWAYS**

**ref:**

**Стойност за сравнение [0..1]**

# Буфер на цвета – логически операции

```
void glLogicOp(GLenum opcode)
```

opcode :

```
GL_CLEAR, GL_SET, GL_COPY,  
GL_COPY_INVERTED, GL_NOOP, GL_INVERT,  
GL_AND, GL_NAND, GL_OR, GL_NOR,  
GL_XOR, GL_EQUIV, GL_AND_REVERSE,  
GL_AND_INVERTED, GL_OR_REVERSE,  
GL_OR_INVERTED
```

# Буфер на дълбочината

```
void glClearDepth(GLclampd depth)
```

```
void glDepthMask(GLboolean flag)
```

```
void glDepthRange(  
    GLclampd zNear,  
    GLclampd zFar)
```

# Буфер на дълбочината

```
void glDepthFunc (GLenum func)  
glEnable (GL_DEPTH_TEST)
```

func:

```
GL_NEVER, GL_LESS, GL_EQUAL,  
GL_LEQUAL, GL_GREATER,  
GL_NOTEQUAL, GL_GEQUAL,  
GL_ALWAYS
```

# Буфер на шаблона

```
void glClearStencil (GLint s)  
void glStencilMask (GLuint mask)
```



# Буфер на шаблона - функция

```
void glStencilFunc(  
    GLenum func, GLint ref, GLuint mask)  
glEnable(GL_STENCIL_TEST)
```

**func:**

```
GL_NEVER, GL_ALWAYS, GL_LESS, GL_LEQUAL,  
GL_EQUAL, GL_GEQUAL, GL_GREATER, GL_NOTEQUAL
```

**ref:**

Стойност за сравнение.

**mask:**

Побитова маска преди сравнение.

# Буфер на шаблона – операции

```
void glStencilOp(GLenum fail,  
                GLenum zfail, GLenum zpass)
```

**fail, zfail и zpass:**

```
GL_KEEP, GL_ZERO, GL_REPLACE,  
GL_INCR, GL_DECR, GL_INVERT
```

<b>fail</b>	ако stencil test пропадне
<b>zfail</b>	ако stencil test успее, но depth test пропадне
<b>zpass</b>	ако stencil test успее, но depth test успее

# Буфер акумулатор – изчистване

```
void glClearAccum(  
    GLfloat red, GLfloat green,  
    GLfloat blue, GLfloat alpha)
```

# Буфер акумулатор

```
void glAccum(GLenum op, GLfloat value)
```

op:

**GL\_ACCUM** чете от буфера активиран за текущ с `glReadBuffer()`, умножава стойностите по `value` и добавя резултата към акумулиращия буфер;

**GL\_LOAD** чете от буфера активиран за текущ с `glReadBuffer()`, умножава стойностите по `value` и записва резултата в акумулиращия буфер;

**GL\_ADD** добавя `value` към всеки пиксел в акумулиращия буфер;

**GL\_MULT** умножава всеки пиксел в акумулиращия буфер по `value`;

**GL\_RETURN** чете стойностите на акумулиращия буфер, умножава ги по `value` и записва резултата в цветовия буфер активиран за запис.

# Работа с растрови изображения

```
void glCopyPixels(GLint x, GLint y,  
    GLsizei width, GLsizei height,  
    GLenum type)
```

type:

GL\_COLOR, GL\_DEPTH или GL\_STENCIL

```
void glDrawPixels(GLsizei width,  
    GLsizei height, GLenum format, GLenum  
    type, const GLvoid *pixels)
```

```
void glPixelZoom(GLfloat xfactor,  
    GLfloat yfactor)
```

# Работа с растерными изображениями

```
void glReadPixels(  
    GLint x, GLint y,  
    GLsizei width, GLsizei height,  
    GLenum format, GLenum type,  
    GLvoid *pixels)
```

# Работа с растрови изображения

```
void glPixelStore...(  
    GLenum pname, GLfloat param)
```

```
void glPixelTransfer...(  
    GLenum pname, GLfloat param)
```

# OpenGL - Буфери

**Въпроси?**