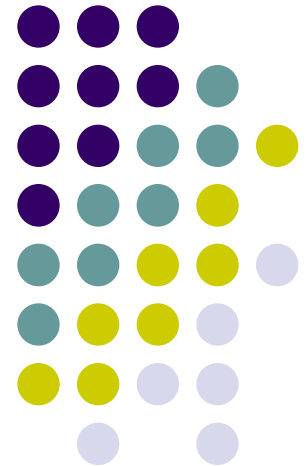
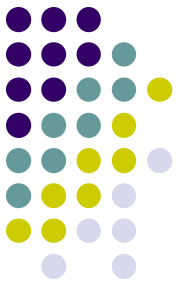


OpenGL

Корлякова М.О.
2019

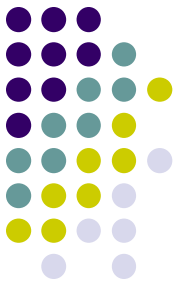




OpenGL

- **OpenGL** - Silicon Graphics, 1993
- цель OpenGL - задание механизма описания процесса. визуализации сложных объектов, а не способов их построения.

работа OpenGL



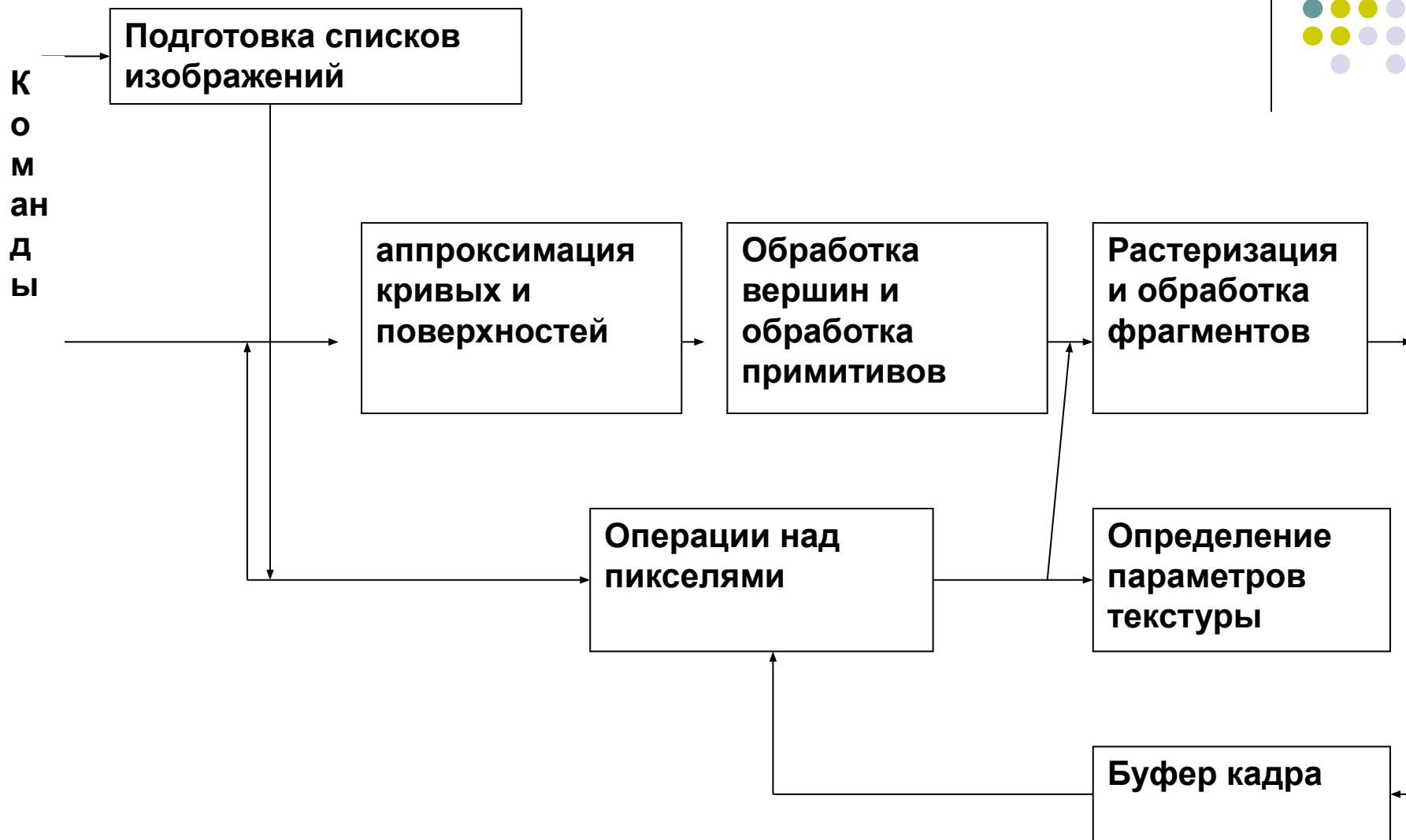
1. получить контекст устройства
2. создать контекст воспроизведения
3. сделать его текущим

Цикл обработки сообщений

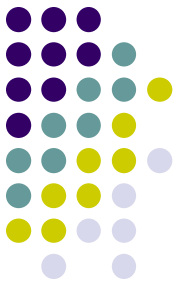
Вызовы команд OpenGL

1. Сделать контекст воспроизведения не текущим
2. Удалить контекст воспроизведения
3. Освободить контекст устройства, связанный с контекстом воспроизведения

Обработка команд в OpenGL

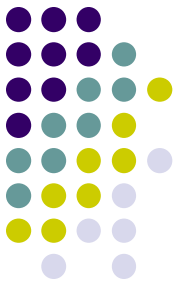


Функция библиотеки OpenGL



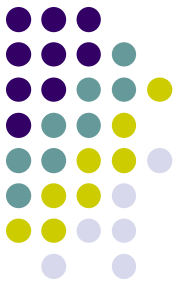
- Создание контекста воспроизведения
- `CreateRenderingContext(DC: HDC; Options: TRCOptions; ColorDepth: Integer; StencilBits: Byte): HGLRC;`
 - HGLRC – тип - контекста воспроизведения
 - DC – контекст устройства
 - Options – условия работы контекста
 - ColorDepth – глубина цвета
 - StencilBits – параметры трафарета

Функция библиотеки OpenGL

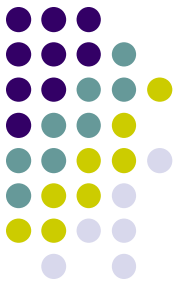


- удаляет контекст воспроизведения
- `DestroyRenderingContext(RC: HGLRC);`

Функция библиотеки OpenGL



- Сделать контекст DC текущим
- `wglMakeCurrent(DC: HDC:
Result : HGLRC):Boolean;`



Синтаксис команд

Имя команды(1, 2, 3, 4,)(b, s, i, f, d, us, ui)(v)

b – байт

s –короткое целое

i – целое

f – вещественное

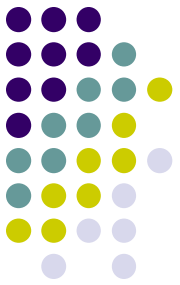
d – вещественное двойной длины

us – без знаковое короткое целое

ui – без знаковое целое

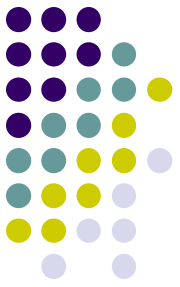
v – вектор.

Вершины



- процедура установления точки
- `glVertex(1 2 3 4)(s i f d)(v)`

Описание объекта из набора вершин

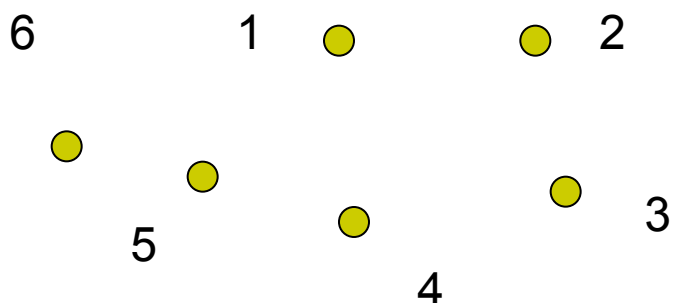


```
glBegin (Имя примитива)  
    ....  
    список вершин  
    ...  
glEnd;
```

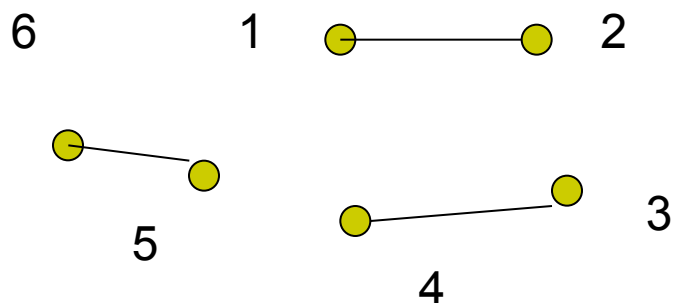


Примитивы OpenGL

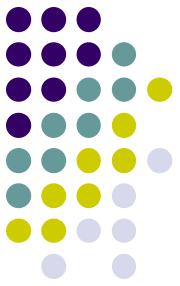
GL_POINTS - последовательность точек;



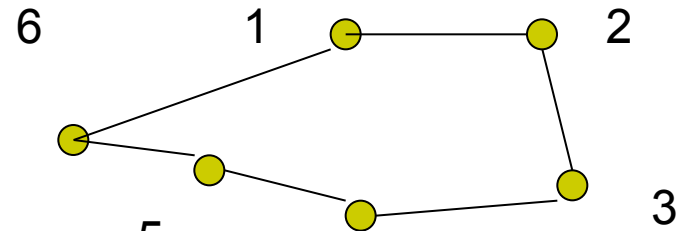
GL_LINES - отрезок линии;



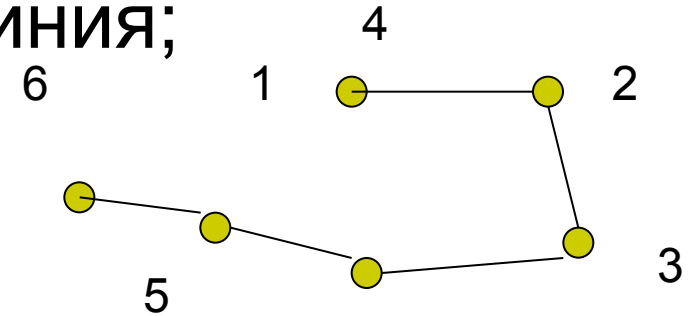
Примитивы OpenGL



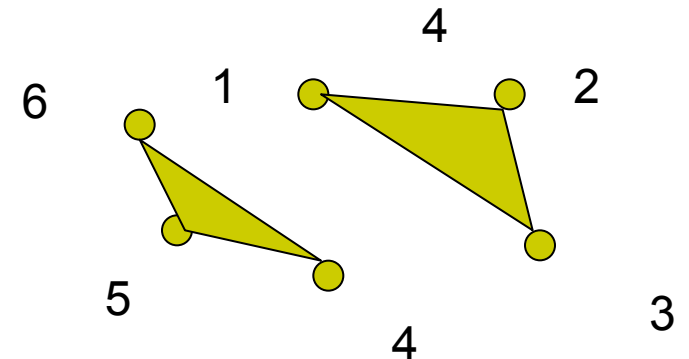
GL_LINE_LOOP - замкнутая ломаная линия;



GL_LINE_STRIP - ломаная линия;



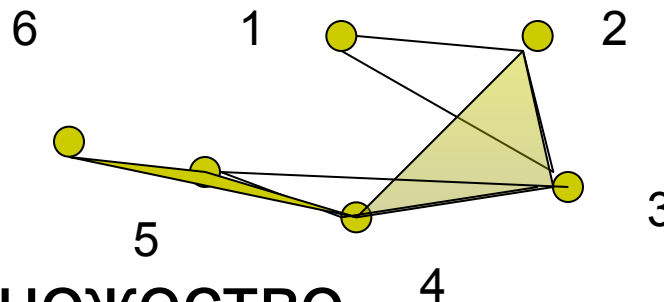
GL_TRIANGLES - отдельный
треугольник;



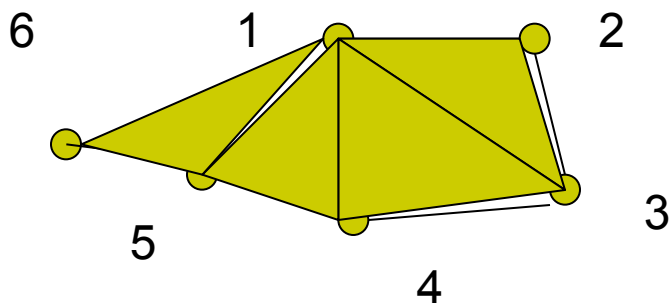


Примитивы OpenGL

`GL_TRIANGLE_STRIP` - множество СВЯЗНЫХ треугольников;



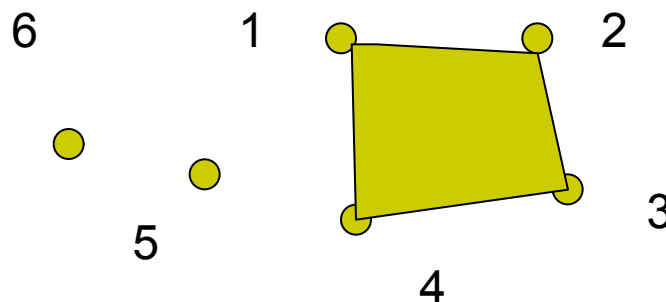
`GL_TRIANGLE_FAN` - множество треугольников, СВЯЗАННЫХ наподобие веера;



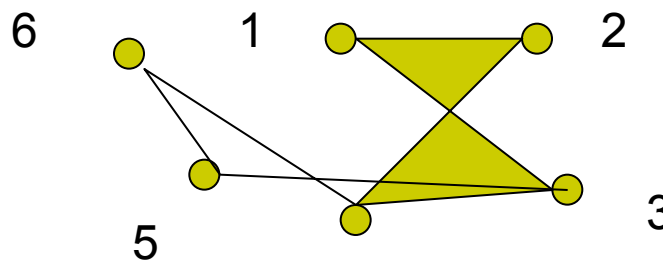
Примитивы OpenGL



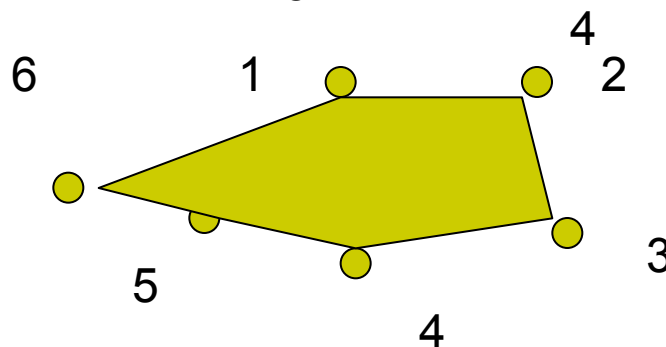
GL_QUADS - отдельный четырехугольник.



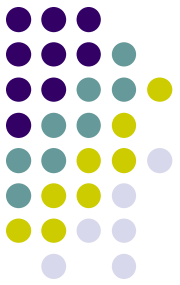
GL_QUAD_STRIP - связанные четырехугольники;



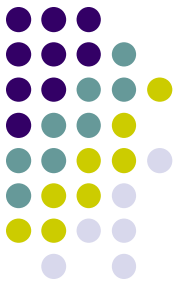
GL_POLYGON - задающая границы выпуклого многоугольника;



Цвет вершины



- glColor[3,4][f,b,l,s,ub ...][v](r,g,b,a:<тип>)

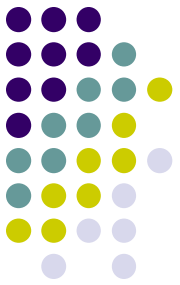


Включение света в OpenGL

- включение режима освещения
- `glEnable(GL_LIGHTING);`

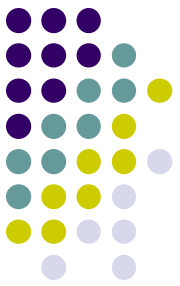
- ИСТОЧНИК СВЕТА
- `glEnable(GL_LIGHT0);`
-
- `glEnable(GL_LIGHT7);`

Составляющие освещения



- **Рассеянный свет**
- Направление нельзя определить

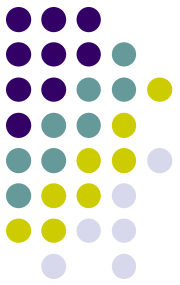
- **Диффузный свет**
- Направленный свет от источника света, который равномерно рассеивается поверхностью.



Составляющие освещения

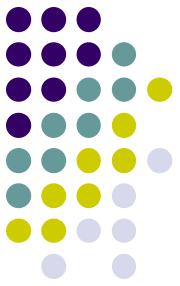
- **Отраженный (зеркальный) свет**
- Направленный свет от источника света, который отражается в определенном направлении

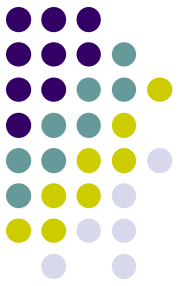
- **Эмиссионный свет (самосвечение)**
- Не добавляет освещение в общую сцену, делает объекты более яркими



- `glLight[if][v](light, pname: GLenum; param: GLType);`
- `Light` - источник
- `Pname` – название параметра
- `Param` – значение параметра

- Var
- `params: TGLArrayf4;`
- Begin
- ...
- `params[0] := 1;`
- `params[1] := 0;`
- `params[2] := 0;`
- `params[3] := 1;`
- `glLightfv(GL_LIGHT0, GL_AMBIENT,`
`@params);`
- ...
- End;

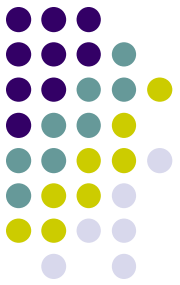




Позиция источника

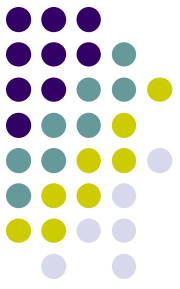
- `GL_POSITION`
- Значение : *указатель на массив* из четырёх значений (x, y, z, w)
- $W=1$
- $W=0$

Направление источника света



- `GL_SPOT_DIRECTION`
- `params` : *указатель* на четыре значения, определяющих направление источника света в однородных мировых координатах
- `GL_SPOT_CUTOFF` – отличен от 180
- `[0, 90]` или 180

Сфокусированность источника света

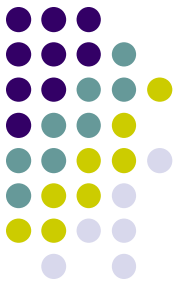


- `GL_SPOT_EXPONENT`
- `param` : целое или вещественное *значение* из диапазона $[0, 128]$.
- Максимальное *значение* соответствует максимальной сфокусированности источника света.
- *Значение 0* соответствует рассеянному освещению

Освещение примитивов

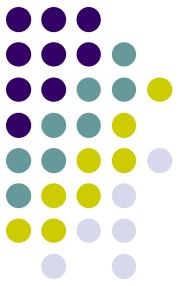


- Для каждой *грани* примитива параметры освещения задаются индивидуально.
- Набор параметров освещения для каждой *грани* называется свойствами материала.
- Каждая грань примитива имеет свой материал.
- Свойства материала определяют рассеянную, диффузную, отражённую и эмиссионную составляющую освещения материала.

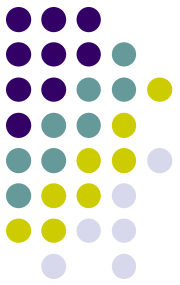


- `glMaterial[i f][v](face, pname: GLenum; param: GLType)`
- `Face` – сторона грани
- `Pname` – параметр материала
- `Param` – значение параметра

пример

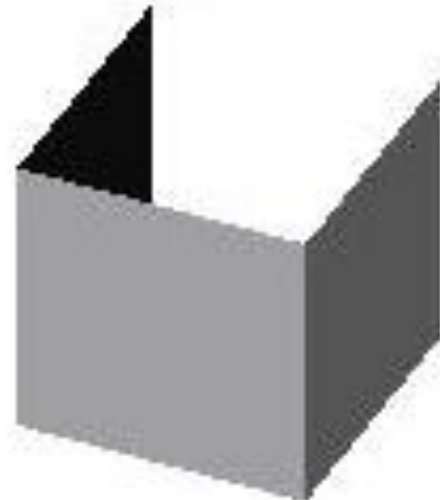


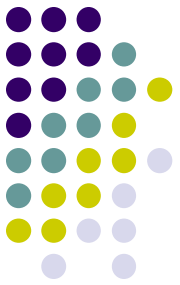
```
glMaterialfv(GL_FRONT, GL_AMBIENT, @mat1_amb);  
glMaterialfv(GL_FRONT, GL_DIFFUSE, @mat1_dif);  
glMaterialfv(GL_FRONT, GL_SPECULAR, @mat1_spec);  
glMaterialf(GL_FRONT, GL_SHININESS, @mat1_shininess)
```



Нормали

- `glNormal3f(nx, ny, nz);`
- `glNormal3f(1, 0, 0);`





Обычные разрешения

```
glEnable(GL_LIGHTING);  
glEnable(GL_LIGHT0);  
glEnable(GL_DEPTH_TEST);  
glEnable(GL_NORMALIZE);  
glEnable(GL_COLOR_MATERIAL);
```