

OpenGL Fiksna funkcionalnost – Spisak funkcija

1. Uvod u OpenGL

**int ChoosePixelFormat(HDC hdc,
CONST PIXELFORMATDESCRIPTOR * ppfd)**

hdc – handle DC-a u kome se traži format piksela najpribližniji zadatom
ppfd – pokazivač na strukturu koja opisuje željeni format piksela

Funkcija vraća indeks formata piksela najpribližniji zadatom formatu. Ukoliko ne uspe, funkcija vraća 0.

**BOOL SetPixelFormat(HDC hdc, int iPixelFormat,
CONST PIXELFORMATDESCRIPTOR * ppfd)**

hdc – handle DC-a čiji se format piksela postavlja
iPixelFormat – indeks koji određuje format piksela koji se postavlja
ppfd – pokazivač na strukturu koja opisuje format piksela (uglavnom nema nikakvu ulogu pri ovom pozivu)

Funkcija vraća TRUE, ukoliko postavljanje formata piksela uspe. U protivnom, vraća FALSE.

HGLRC wglCreateContext(HDC hdc)

hdc – handle DC-a za koji se kreira odgovarajući OpenGL *rendering context*

Funkcija kreira OpenGL *rendering context* pogodan za iscertavanje na uređaju čiji je DC prosleđen kao parametar. OpenGL *rendering context* ima isti format piksela kao i prosleđeni DC. Ako funkcija uspe, vraća *handle* kreiranog OpenGL *rendering context*-a. U protivnom, vraća NULL.

BOOL wglDeleteContext(HGLRC hglrc)

hglrc - handle OpenGL *rendering context* koji funkcija briše

Funkcija briše OpenGL *rendering context* čiji je *handle* prosleđen. Ako uspe, funkcija vraća TRUE. U suprotnom vraća FALSE.

BOOL wglMakeCurrent(HDC hdc, HGLRC hglrc)

hdc – handle DC-a uređaja na kome će OpenGL funkcije koje slede vršiti iscertavanje
hglrc - handle OpenGL *rendering context*-a koji postaje tekući za nit koja je pozvala ovu funkciju

Funkcija postavlja tekući OpenGL *rendering context* za nit koja je pozvala. Ako je *hglrc* NULL, funkcija oslobađa DC koji je do tada koristio OpenGL *rendering context*, i nakon toga deselektuje tekući *rendering context*.

BOOL SwapBuffers(HDC hdc)

hdc – handle DC-a čiji se baferi menjaju

Funkcija zamenjuje prednji (*front*) i zadnji (*back*) bafer prozora, ako format piksela uključuje *back*-bafer. Funkcija vraća TRUE, ako uspe. U protivnom, vraća FALSE.

CDC* CWnd::GetDC()

Funkcija članica prozora (CWnd). Vraća pokazivač na DC datog prozora.

int CWnd::ReleaseDC(CDC* pDC)

pDC – pokazivač na DC koji se oslobađa

Funkcija članica prozora (CWnd). Oslobađa DC, čiji se pokazivač prosleđuje kao parametar. Vraća 0 ako ne uspe. U protivnom vrednost različitu od 0.

2. Crtanje primitiva

```
void glColor3{ubsidf}(TYPE red, TYPE green, TYPE blue) ;  
void glColor4{ubsidf}(TYPE red, TYPE green, TYPE blue, TYPE alpha) ;  
void glColor{34}{ubsidf}v( const TYPE *v );
```

red – intenzitet crvene komponente
green – intenzitet zelene komponente
blue – intenzitet plave komponente
alpha – intenzitet *alpha* komponente (providnost)
v – pokazivač na polje koje sadrži RGB(A) komponente boje

Postavlja tekuću boju. Primenjuje se na sve primitive koje se iscrtavaju nakon poziva ove funkcije.

```
void glVertex2{dfis}( TYPE x, TYPE y );  
void glVertex3{dfis}( TYPE x, TYPE y , TYPE z);  
void glVertex4{dfis}( TYPE x, TYPE y , TYPE z, TYPE w);  
void glVertex{234}{dfis}v( const TYPE *v);
```

x, y, z, w – koordinate
v – pokazivač na polje sa koordinatama

Definiše prostorne koordinate temena.

```
void glBegin( GLenum mode ); / void glEnd();
```

mode – tip primitive koja se kreira na osnovu definicije prostornih koordinata.
Može imati jednu od sledećih vrednosti:
GL_POINTS – pojedinačne tačke

GL_LINES – pojedinačne duži
 GL_LINE_STRIP – serija međusobno povezanih duži
 GL_LINE_LOOP – petlja
 GL_TRIANGLES – pojedinačni trouglovi
 GL_TRIANGLE_STRIP – traka trouglova
 GL_TRIANGLE_FAN – lepeza trouglova
 GL_QUADS – pojedinačni četvorouglovi
 GL_QUAD_STRIP – traka četvorouglova
 GL_POLYGON – poligon

Ove dve funkcije uvek idu u paru. Između njih navode se definicije temena. Tip primitive koji se kreira zavisi od parametra prosleđenog funkciji **glBegin()**.

void glPointSize(GLfloat size) ;

size – veličina tačke

Definiše veličinu tačke (primitive GL_POINT) u pikselima.

void glGetBooleanv(GLenum pname, GLboolean * params);
void glGetDoublev (GLenum pname, GLdouble * params);
void glGetFloatv (GLenum pname, GLfloat * params);
void glGetIntegerv(GLenum pname, GLint * params);

pname – ime parametra koji je potrebno očitati. U ovom poglavlju korišćeni su sledeći parametri:

GL_POINT_SIZE – veličina tačke (GL_POINT) u pikselima,
 GL_POINT_SIZE_GRANULARITY – rezolucija sa kojom se može menjati veličina tačke,
 GL_POINT_SIZE_RANGE – opseg u kome se može menjati veličina tačke,
 GL_LINE_WIDTH – debljina linije u pikselima,
 GL_LINE_WIDTH_GRANULARITY – rezolucija sa kojom se može menjati debljina linije,
 GL_LINE_WIDTH_RANGE – opseg u kome se može menjati debljina linije,
 GL_LINE_STIPPLE_PATTERN – tekući šablon linije,
 GL_LINE_STIPPLE_REPEAT – faktor množenja (ponavljanja) šablona,
 GL_LINE_STIPPLE – indikator da li je uključen šablon

params – očitana vrednost (ili vrednosti).

Navedene funkcije služe za očitavanje različitih parametara i stanja u kome se nalazi OpenGL *rendering context*. U OpenGL-u 1.1 mogu se na ovaj način očitati preko 200 različitih vrednosti.

void glLineWidth(GLfloat width);

width – debljina linije u pikselima

Postavlja debljinu linije na zadatu vrednost, pod uslovom da se nalazi u granicama

koje su određene parametrom `GL_LINE_WIDTH_RANGE` i sa korakom `GL_LINE_WIDTH_GRANULARITY`.

`void glLineStipple(GLint factor, GLushort pattern);`

factor – faktor kojim se množi šablon,
pattern – sam šablon.

Definiše šablon linije.

`void glEnable(GLenum cap) / void glDisable(GLenum cap)`

cap – stanje koje se uključuje/isključuje. Može imati preko 40 različitih vrednosti, ne računajući podoblike. U ovom poglavlju koriste se samo:

`GL_LINE_STIPPLE` – uključuje/isključuje šablon za linije,
`GL_CULL_FACE` – uključuje/isključuje odbacivanje poligona koji se vide sa prednje ili zadnje strane.

Ovaj par funkcija uključuje/isključuje odgovarajuće stanje u OpenGL *rendering context*-u.

`void glPolygonMode(GLenum face, GLenum mode);`

face – strana poligona za koju važi dati način iscrtavanja. Može imati jednu od sledećih vrednosti:

`GL_FRONT` – prednja,
`GL_BACK` – zadnja,
`GL_FRONT_AND_BACK` – i prednja i zadnja.

mode – način iscrtavanja. Može imati jednu od sledećih vrednosti:

`GL_POINT` – tačke,
`GL_LINE` – linije,
`GL_FILL` – ispunjena.

Definiše način iscrtavanja trouglova, četvorouglova i poligona. Moguće je iscrtavati samo tačke u temenima (`GL_POINT`), linije na ivicama (`GL_LINE`) ili spuniti čitavu površinu (`GL_FILL`). Podrazumevano iscrtavanje je `GL_FILL`.

`void glFrontFace(GLenum mode)`

mode – redosled zadavanja temena. Može imati jednu od sledeće dve vrednosti:

`GL_CCW` – prednja strana postaje ona koja ima orijentaciju suprotnu od kretanja kazaljki na časovniku (ovo je podrazumevano stanje),
`GL_CW` – prednja strana postaje ona koja ima orijentaciju u pravcu kretanja kazaljki na časovniku.

Definiše pozitivnu orijentaciju, tj. orijentaciju strane trougla, četvorougla ili poligona koja se smatra prednjom.

void glCullFace(GLenum mode);

mode – definiše da li će poligoni koji se vide sa prednje ili zadnje strane biti odbačeni prilikom iscrtavanja. Može imati jednu od sledećih vrednosti:

GL_FRONT – odbacuju se poligoni koji se vide sa prednje strane,
GL_BACK – odbacuju se poligoni koji se vide sa zadnje strane,
GL_FRONT_AND_BACK – poligoni se odbacuju bez obzira sa koje se strane vide.

Definiše da li će poligon biti iscrtan zavisno od strane koja je vidljiva.

void glVertexPointer(GLint size, GLenum type, GLsizei stride, const GLvoid *pointer);

size – broj koordinata po jednom temenu (2, 3 ili 4),
type – tip podataka u polju (GL_SHORT, GL_INT, GL_FLOAT, ili GL_DOUBLE),
stride – rastojanje u bajtovima između dva sukcesivna temena,
pointer – pokazivač na polje prostornih koordinata temena.

Definiše memorijsku organizaciju polja prostornih koordinata temena i pokazivač na dato polje.

void glColorPointer(GLint size, GLenum type, GLsizei stride, const GLvoid *pointer);

size – broj koordinata po jednom temenu (3 ili 4),
type – tip podataka u polju (GL_BYTE, GL_UNSIGNED_BYTE, GL_SHORT, GL_UNSIGNED_SHORT, GL_INT, GL_UNSIGNED_INT, GL_FLOAT, ili GL_DOUBLE),
stride – rastojanje u bajtovima između dva sukcesivna temena,
pointer – pokazivač na polje boja temena.

Definiše memorijsku organizaciju polja boja temena i pokazivač na dato polje.

void glEnableClientState(GLenum array);
void glDisableClientState(GLenum array);

array – definiše polje atributa temena koje se aktivira/deaktivira. U ovom poglavlju korišćene su sledeće vrednosti:

GL_VERTEX_ARRAY – polje prostornih koordinata temena,
GL_COLOR_ARRAY – polje boja temena,

Aktivira/deaktivira odgovarajuće polje atributa temena.

void glArrayElement(GLint i);

i – indeks temena, u aktiviranim poljima temena, koje se iscrtava

Javlja se između poziva funkcija **glBegin()** i **glEnd()** i služi za iscrtavanje primitiva korišćenjem polja temena, proizvoljnim „skakanjem“ po polju i pristupom pojedinačnim elementima. Koriste se atributi iz svih aktiviranih polja, a tip primitive koja se formira zavisi od parametra funkcije **glBegin()**.

void glDrawArrays(GLenum mode, GLint first, GLsizei count);

mode – tip primitive koja se formira (GL_POINTS, GL_LINES, itd.)
first – indeks prvog temena kome se pristupa u svim aktivnim poljima
count – broj temena koja se koriste za formiranje primitiva

Isertava primitive definisane parametrom *mode*, sekvencijalnim pristupom elementima svih aktivnih polja temena i njihovih atributa, počev od indeksa *first*, a zaključno sa indeksom *first+count-1*.

void glDrawElements(GLenum mode, GLsizei count, GLenum type, const GLvoid *indices);

mode – tip primitive koja se formira (GL_POINTS, GL_LINES, itd.)
count – broj elemenata u polju indeksa
type – tip podataka smeštenih u polju indeksa
indices – pokazivač na polje indeksa

Isertava primitive definisane parametrom *mode*, pristupom elementima svih aktivnih polja temena i njihovih atributa na osnovu polja indeksa, na koje ukazuje pokazivač *indices*. Broj elemenata u polju indeksa definisan je parametrom *count*, a njihov tip parametrom *type*.

void glShadeModel(GLenum mode)

mode – način senčenja. Može imati jednu od sledećih vrednosti:
GL_FLAT – jednobožno senčenje,
GL_SMOOTH – umekšano senčenje (boje se interpoliraju na osnovu vrednosti zadatih u temenima).

Definiše način senčenja poligona. Podrazumevano je umekšano senčenje, kod koga se boja piksela u unutrašnjosti poligona izračunava interpolacijom boje temena. Kod jednobožnog senčenja, boja poslednjeg temena u primitivi definiše boju čitave primitive (osim kod poligona, gde prvo teme definiše boju primitive).

3. Pogled u 3D

void glMatrixMode(GLenum mode);

mode – magacin matrica koji se selektuje, može imati jednu od sledećih vrednosti:
GL_MODELVIEW,
GL_PROJECTION i
GL_TEXTURE

Selektuje tekući magacin matrica.

void glLoadMatrix{fd}(const TYPE *m);

m – pokazivač na matricu dimezija 4×4 smeštene u vektor po kolonama

Zamenjuje matricu na vrhu tekućeg magacina matrica prosleđenom matricom.

void glLoadIdentity();

Zamenjuje matricu na vrhu tekućeg magacina matrica jediničnom matricom.

void glMultMatrix{fd}(const TYPE *m);

m – pokazivač na matricu dimezija 4×4 smeštene u vektor po kolonama

Množi matricu na vrhu tekućeg magacina matrica (C) prosleđenom matricom (M) sa desne strane. $C = CM$

void glTranslate{fd}(TYPE x, TYPE y, TYPE z);

x, y, z – koordinate vektora translacije

Pomera koordinatni početak lokalnog koordinatnog sistema u tačku (*x, y, z*).

void glRotate{fd}(TYPE angle, TYPE x, TYPE y, TYPE z);

angle – ugao rotacije zadat u stepenima

x, y, z – koordinate vektora translacije

Vrši rotaciju u smeru suprotnom od kazaljki na časovniku za ugao *angle* oko vektora [*x, y, z*].

void glScale{fd}(TYPE x, TYPE y, TYPE z);

x, y, z – faktori skaliranja po X, Y i Z-osi, respektivno

Vrši skaliranje po X, Y i Z-osi za zadate vrednosti.

void glEnable(GLenum cap) / void glDisable(GLenum cap)

cap – stanje koje se uključuje/isključuje. Može imati preko 40 različitih vrednosti, ne računajući podoblike. U ovom poglavlju koriste se samo 3:

GL_NORMALIZE – uključuje/isključuje automatsku normalizaciju vektora normala

GL_DEPTH_TEST – uključuje/isključuje poređenje „dubine“ fragmenata i ažuriranje Z-bafera

GL_CULL_FACE – uključuje/isključuje odbacivanje primitiva u zavisnosti od orijentacije

Ovaj par funkcija uključuje/isključuje odgovarajuće stanje u OpenGL *rendering context*-u

void glPushMatrix() / void glPopMatrix()

Ovaj par funkcija menja stanje selektovanog magacina matrica. Funkcija **glPushMatrix()** kopira matricu na vrhu i smešta je u magacin. Nakon poziva ove funkcije i na vrhu magacina i ispod njega nalaze se iste matrice. Funkcija **glPopMatrix()** uklanja matricu sa vrha magacina.

**void gluLookAt(GLdouble eyex, GLdouble eyey, GLdouble eyez,
GLdouble centerx, GLdouble centery, GLdouble centerz,
GLdouble upx, GLdouble upy, GLdouble upz);**

eyex, eyey, eyez – pozicija oka posmatrača
centerx, centery, centerz – tačka u koju je usmeren pogled
upx, upy, upz – vektor vertikale

Definiše transformaciju pogleda.

**void glOrtho(GLdouble left, GLdouble right, GLdouble bottom,
GLdouble top, GLdouble near, GLdouble far);**

left, right – koordinate leve i desne vertikalne ravni odsecanja
bottom, top – koordinate donje i gornje horizontalne ravni odsecanja
near, far – udaljenost prednje i zadnje ravni odsecanja od posmatrača

Definiše ortografsku projekciju.

**void glFrustum(GLdouble left, GLdouble right, GLdouble bottom,
GLdouble top, GLdouble near, GLdouble far);**

left, right – x-koordinate leve i desne ravni odsecanja na preseku sa prednjom ravni odsecanja
bottom, top – y-koordinate donje i gornje ravni odsecanja na preseku sa prednjom ravni odsecanja
near, far – udaljenost prednje i zadnje ravni odsecanja od posmatrača

Definiše perspektivnu projekciju.

**void gluPerspective(GLdouble fovy, GLdouble aspect,
GLdouble near, GLdouble far);**

fovy – ugao vidnog polja u YZ-ravni izražen u stepenima
aspect – odnos širine i visine vidnog polja
near, far – udaljenost prednje i zadnje ravni odsecanja od posmatrača

Definiše perspektivnu projekciju.

void glViewport(GLint x, GLint y, GLsizei width, GLsizei height);

x, y – koordinate donjeg levog temena pravougaonika u kome se prikazuje slika
width, height – širina i visina pravougaonika u kome se prikazuje slika

Definiše **viewport** transformaciju, tj. deo prozora u kome se prikazuje slika.

Koordinata (0,0) odgovara donjem levom uglu prozora.

void glClearDepth(GLclampd depth);

depth – vrednost koja se koristi za „čišćenje“ Z-bafera

Definiše vrednost iz intervala [0, 1] koja se upisuje u sve lokacije Z-bafera prilikom poziva funkcije **glClear()** ukoliko je postavljen odgovarajući fleg.

void glClear(GLbitfield mask);

mask – polje flegova (indikatora) koji sve baferi trebaju biti obrisani pri pozivu ove funkcije. Koristi se bitska „ili“ kombinacija sledećih predefinisani vrednosti:

GL_COLOR_BUFFER_BIT – bafer u kome se formira slika
za prikaz (*framebuffer*)

GL_DEPTH_BUFFER_BIT – Z-bafer

Briše specificirane bafere, odnosno postavlja vrednosti definisane odgovarajućim funkcijama. Bafer slike ispunjuje se bojom definisanom funkcijom **glClearColor()**, a Z-bafer vrednošću postavljenom funkcijom **glClearDepth()**.

4. Osvetljenje

void glNormal3{bsidf}(TYPE nx, TYPE ny, TYPE nz)
void glNormal3{bsidf}v(const TYPE *v)

nx, ny, nz – koordinate vektora normale
v – koordinate vektora normale zadate kao vektor

Definiše normalu u temenima objekta.

void glEnable(GLenum cap) / void glDisable(GLenum cap)

cap – stanje koje se uključuje/isključuje. Može imati preko 40 različitih vrednosti, ne računajući podoblike. U ovoj glavi koriste se samo 3:

GL_NORMALIZE – uključuje/isključuje automatsku normalizaciju vektora normala

GL_LIGHTING – uključuje osvetljenje

GL_LIGHTi – uključuje i-ti izvor svetlosti (npr. GL_LIGHT1)

Ovaj par funkcija uključuje/isključuje odgovarajuće stanje u OpenGL *rendering context*-u

glMaterial{f}[v] (GLenum face, GLenum pname, TYPE param)

face – stranu poligona na koju se primenjuje dati materijal i može imati vrednost:

GL_FRONT – samo prednja strana,

GL_BACK – samo zadnja strana ili

GL_FRONT_AND_BACK – i prednja i zadnja strana.

pname – naziv parametra čija se vrednost modifikuje, može biti:

GL_AMBIENT – ambijentalna boja,
GL_DIFFUSE – difuziona boja,
GL_AMBIENT_AND_DIFFUSE – ambijentalna i difuziona boja
GL_SPECULAR – reflektovana boja (odsjaaj),
GL_EMISSION – emisiona boja, ili
GL_SHININESS – eksponent odsjaja

param – vrednost na koju se postavlja prethodni parametar

Funkcija postavlja sve parametre materijala.

void glLightModel{fi}[v] (GLenum *pname*, TYPE *param*)

pname – naziv parametra čija se vrednost modifikuje, a može biti:

GL_LIGHT_MODEL_AMBIENT – globalno ambijentalno osvetljenje,
GL_LIGHT_MODEL_LOCAL_VIEWER – lokalni ili udaljeni
posmatrač,
GL_LIGHT_MODEL_TWO_SIDE – jednostrano ili dvostrano
osvetljenje

param – vrednost na koju se postavlja prethodni parametar

Funkcija postavlja parametre modela osvetljenja.

void glLight{if}[v](GLenum *light*, GLenum *pname*, TYPE *param*)

light – izvor svetlosti čiji se parametar modifikuje, može biti: GL_LIGHT0,
GL_LIGHT1, GL_LIGHT2, GL_LIGHT3, GL_LIGHT4, GL_LIGHT5,
GL_LIGHT6, ili GL_LIGHT7

pname – naziv parametra čija se vrednost modifikuje, može biti:

GL_AMBIENT – ambijentalna komponenta,
GL_DIFFUSE – difuziona komponenta,
GL_SPECULAR – refleksiona komponenta,
GL_POSITION – pozicija,
GL_CONSTANT_ATTENUATION – konstantni faktor slabljenja,
GL_LINEAR_ATTENUATION – linearni faktor slabljenja,
GL_QUADRATIC_ATTENUATION – kvadratni faktor slabljenja,
GL_SPOT_DIRECTION – orijentacija usmerenog izvora svetlosti,
GL_SPOT_CUTOFF – ugao rasipanja svetlosti u odnosu na dati pravac,
GL_SPOT_EXPONENT – eksponent distribucije svetlosti.

param – vrednost na koju se postavlja prethodni parametar

Funkcija postavlja parametre izvora osvetljenja.

5. Teksture

```
void glPixelStore{if}(GLenum pname, TYPE param);
```

pname – naziv parametra (postoji 12 različitih parametara: GL_PACK_SWAP_BYTES, GL_PACK_LSB_FIRST, GL_PACK_ROW_LENGTH, GL_PACK_SKIP_PIXELS, GL_PACK_SKIP_ROWS, GL_PACK_ALIGNMENT, GL_UNPACK_SWAP_BYTES, GL_UNPACK_LSB_FIRST, GL_UNPACK_ROW_LENGTH, GL_UNPACK_SKIP_PIXELS, GL_UNPACK_SKIP_ROWS, GL_UNPACK_ALIGNMENT)

param – vrednost parametra

Postavlja modove smeštanja piksela. Korišćen je samo parametar GL_UNPACK_ALIGNMENT za postavljanje poravnanja nezapakovanih tekstura.

```
void glTexEnv{if}( GLenum target, GLenum pname, GLint param );
```

target – mora biti postavljen na GL_TEXTURE_ENV

pname – mora biti postavljen na GL_TEXTURE_ENV_MODE

param – način primene teksture, može imati sledeće vrednosti:

GL_REPLACE – potpuno menja boju fragmenta podacima iz teksture,

GL_DECAL – mešanje boje fragmenta i teksture definisano je providnošću teksture,

GL_MODULATE – boja fragmenata je „modulisana“ teksturom i

GL_BLEND – boja se dobija mešanjem boje fragmenta, teksture i posebno zadate boje.

Definiše način primene teksture.

```
void glTexEnv{if}v(GLenum target, GLenum pname, GLint param );
```

target – mora biti postavljen na GL_TEXTURE_ENV

pname – mora biti postavljen na GL_TEXTURE_ENV_COLOR

param – definiše RGBA vrednost boje koja se koristi za mešanje.

Definiše boju koja se koristi u modu GL_BLEND.

```
void glGenTextures(GLsizei n, GLuint *textureNames);
```

n – broj identifikatora koji se vraćaju

textureNames – polje slobodnih identifikatora

Vraća *n* trenutno slobodnih identifikatora, u obliku neoznačenih celih brojeva, koje smešta u polje *textureNames*.

```
void glBindTexture(GLenum target, GLuint textureName);
```

target – tip teksture koja se aktivira. Može imati jednu od sledećih vrednosti:

GL_TEXTURE_1D – 1D tekstura,

GL_TEXTURE_2D – 2D tekstura,

GL_TEXTURE_3D – 3D tekstura,
 GL_TEXTURE_CUBE_MAP – kubna tekstura,
textureName – identifikator teksture koja se aktivira.

Kada se prvi put pozove za odgovarajući identifikator, dobijen pozivom funkcije **glGenTextures()**, objekat teksture se kreira i aktivira (selektuje). Sve kasnije modifikacije parametara odnoseće se na selektovani objekat. Kada se pozove za prethodno kreirani objekat, onda samo aktivira dati objekat. A kada se pozove za vrednost 0, isključuje rad sa objektima tekstura i aktivira neimenovanu „podrazumevanu“ teksturu.

void glTexImage2D(GLenum target, GLint level, GLint internalFormat, GLsizei width, GLsizei height, GLint border, GLenum format, GLenum type, const GLvoid *texels);

target – tip teksture (GL_TEXTURE_2D),
level – nivo teksture koji se popunjava,
internalFormat – broj komponentata jednog teksela (1, 2, 3, 4 ili jednu od 55 drugih predefinisanih vrednosti), najčešće koristimo GL_RGB (3) ili GL_RGBA (4)
width – širina u tekselima,
height – visina u tekselima,
border – širina okvira,
format – format teksela (GL_RGB, GL_RGBA, ...),
type – tip podataka koji formiraju texel (GL_UNSIGNED_BYTE, ...),
texels – pokazivač na bafer sa podacima, koje treba preneti u teksturu.

Definiše format teksture i prenosi podatke.

int gluBuild2DMipmaps(GLenum target, GLint internalFormat, GLint width, GLint height, GLenum format, GLenum type, void *texels);

target – tip teksture (GL_TEXTURE_2D),
internalFormat – broj komponenti jednog teksela (vidi **glTexImage2D()**)
width, height – širina i visina u tekselima,
format – format teksela (GL_RGB, GL_RGBA, ...),
type – tip podataka koji formiraju texel (GL_UNSIGNED_BYTE, ...),
texels – pokazivač na bafer sa podacima, koje treba preneti u teksturu.

Definiše format teksture, prenosi podatke i formira kompletnu mipmapu. Automatski prilagođava najviši nivo detalja tako da dimenzije budu stepen broja 2.

AUX_RGBImageRec* auxDIBImageLoad(LPCWSTR fileName);

fileName – naziv (BMP) datoteke iz koje se učitava slika

Učitava sliku u strukturu *AUX_RGBImageRec* koja ima tri atributa:

- *data* – bafer u kome je smeštena sadržina slike (pikseli),
- *sizeX* – širina slike i
- *sizeY* – visina slike.

Ova funkcija pripada pomoćnoj biblioteci **glaux.lib**, koja se mora uključiti u projekat da bi funkcija bila dostupna.

void glTexParameteri(GLenum target, GLenum pname, TYPE param);

target – tip teksture (GL_TEXTURE_2D),
pname – naziv parametra koji se postavlja (GL_TEXTURE_MAG_FILTER, GL_TEXTURE_MIN_FILTER, GL_TEXTURE_WRAP_S, GL_TEXTURE_WRAP_T),
param – vrednost na koju se postavlja parametar (GL_NEAREST, GL_LINEAR, GL_NEAREST_MIPMAP_NEAREST, GL_NEAREST_MIPMAP_LINEAR, GL_LINEAR_MIPMAP_NEAREST, GL_LINEAR_MIPMAP_LINEAR, GL_CLAMP, GL_REPEAT).

Definiše različite parametre tekstura.

void glGetIntegerv(GLenum pname, GLint * params);

pname – ime parametra koji je potrebno očitati. U ovom poglavlju korišćen je samo sledeći parametar:
GL_MAX_TEXTURE_SIZE – maksimalna dimenzija teksture u
tekstselima.

params – očitana vrednost (ili vrednosti).

Navedene funkcije služe za očitavanje različitih parametara i stanja u kome se nalazi OpenGL *rendering context*. U OpenGL-u 1.1 mogu se na ovaj način očitati preko 200 različitih vrednosti.

**void glTexCoord{1234}{sfd}(TYPE coords);
void glTexCoord{1234}{sfd}v(const TYPE *coords);**

coords – teksturne koordinate datog temena

Postavlja teksturne koordinate temena čija definicija sledi nakon poziva ove funkcije.

void glEnable(GLenum cap) / void glDisable(GLenum cap)

cap – stanje koje se uključuje/isključuje. Može imati preko 40 različitih vrednosti, ne računajući podoblike. U ovoj glavi koriste se samo:
GL_TEXTURE_2D – uključuje/isključuje 2D teksture.

Ovaj par funkcija uključuje/isključuje odgovarajuće stanje u OpenGL *rendering context*-u.

void glDeleteTextures(GLsize n, const GLuint *textureNames);

n – broj tekstura koje se brišu
textureNames – polje identifikatora tekstura.

Briše teksture sa identifikatorima koji su navedeni u polju *textureNames*.