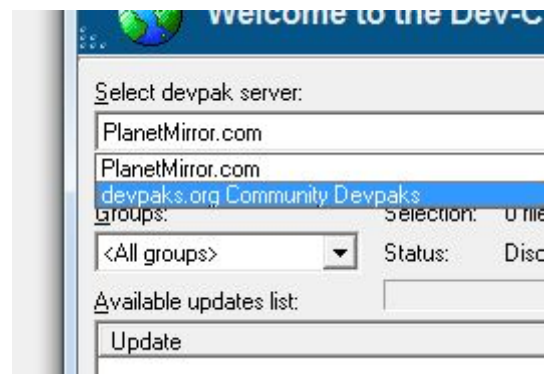
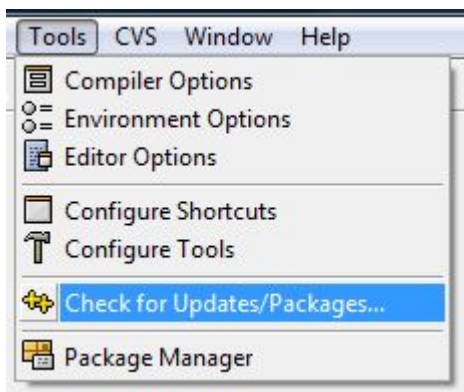


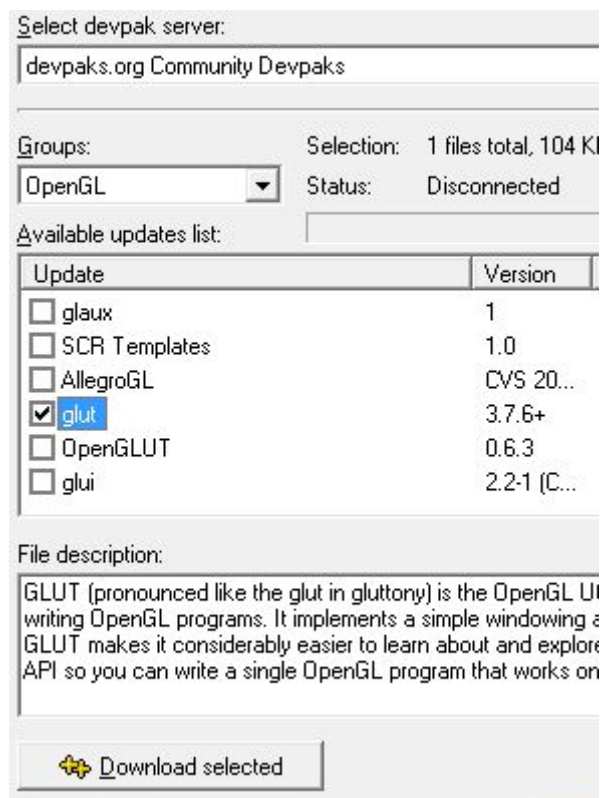
OPENGL პრაქტიკულის დამზმარე (1-ი ლექცია)

პროგრამის დასაწერად საჭიროა გარსი სადაც ავკრიფავთ კოდს დავაკომპილირებთ და გავუშვებთ. ჩვენ გამოვიყენებთ DevC++ IDE-ს.

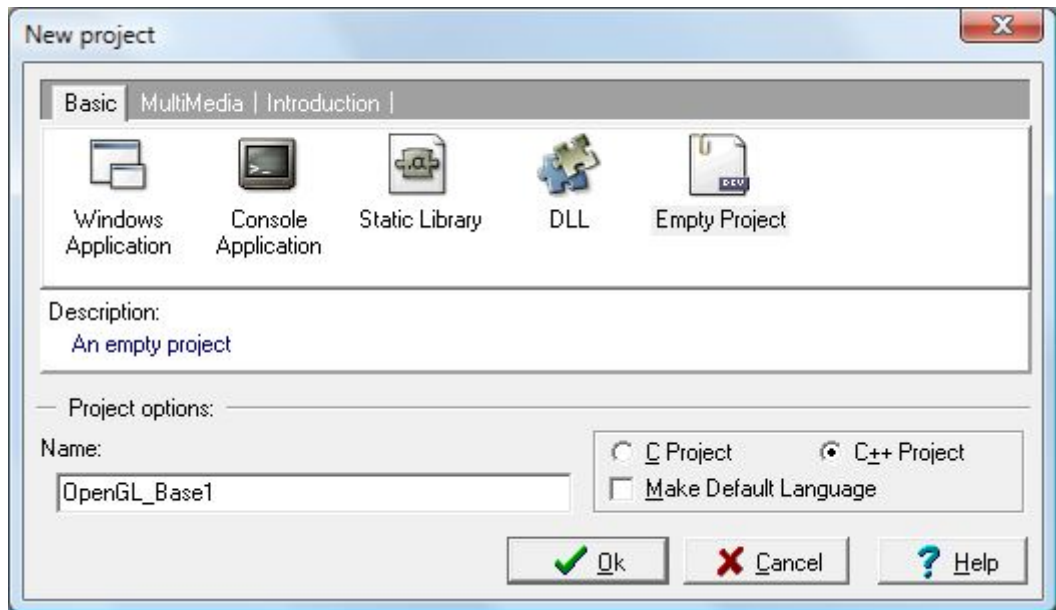
1. მოძებნეთ თუ აყენია კომპიუტერში DevC++-სი. თუ უკვე დაყენებულია გადადით 2-ე პუნქტზე.
 - a. ჩამოქაჩეთ DevC++-ის პაკეტი შემდეგი მისამართიდან:
<http://www.bloodshed.net/dev/devcpp.html> (მიმდინარე ვერსია 4.9.9.2)
 - b. დააინსტალირეთ კომპიუტერზე.
 - c. გაუშვით დარწმუნდით რომ ნორმალურად ეშვება და მუშაობს.
 - d. OpenGL-ზე პროგრამების წერისას ჩვენ ვიყენებთ GLUT ბიბლიოთეკას. თუ უკვე დამატებულია GLUT-ი ან freeGLUT-ი მაშინ გადადით 3-ე პუნქტზე.
 - e. გაუშვით DevC++-ი.
 - f. გახსენით შემდეგი მენიუ და აირჩიეთ "devpaks.org..." საიდანაც დავქაჩავთ ბიბლიოთეკას:



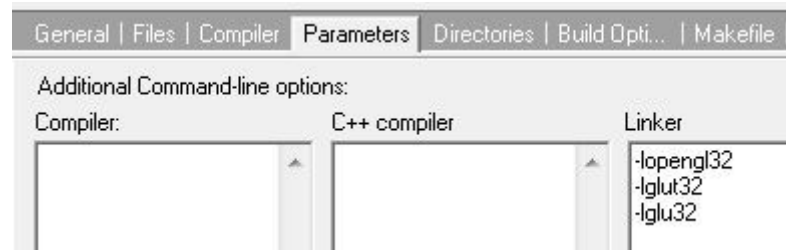
- h. დააჭირეთ ღილაკს "Check for updates" და დაელოდეთ სანამ დაიქაჩება ბიბლიოთეკების სია.



- i. ჩამოქაჩული სიიდან აირჩიეთ და მონიშნეთ ან glut ან freeglut ბიბლიოთეკა და დააჭირეთ ლილავს "download selected". დაელოდეთ ბიბლიოთეკის ჩამოქაჩვას და დაყენებას.
2. ამის მერე DevC++-სი მზადაა OpenGL-ზე პროგრამების დასაწერად. შექმენით ახალი ბაზური პროექტი.
 - a. დესკტოპზე შექმენით ახალი ფოლდერი, მაგალითად OpenGL Projects.
 - b. გახსენით ეს ფოლდერი და შექმენით პროექტის ფოლდერი, მაგალითად OpenGL_base1
 - c. DevC++-ში აირჩიეთ File->new Project. აირჩიეთ Empty Project და შეცვალეთ პროექტის სახელი.



- d. დააჭირეთ OK-ს და შეინახეთ პროექტი Desktop->OpenGL Projects->OpenGL_base1 ფოლდერში.
- e. აირჩიეთ Project->Project Options მენიუ ან დააჭირეთ ALT+P. გადადით "Parameters" ტაბზე და ჩაამატეთ ლინკერისთვის შემდეგი მნიშვნელობები (ამით ლინკერი მიხვდება რა GLUT-ის ბიბლიოთეკები გვჭირდება და რომლები მიუერთოს



კომპილაციის მერე) :

- f. OK და აირჩიეთ File New->Source File ან Ctrl+N. შევინახოთ *.cpp ფაილი Ctrl+S და დავარქვათ სახელი.
3. პროგრამის გასაშვებად დაგვჭირდება glut32.dll. მოქაჩეთ აქედან (<http://ikabika.narod.ru/files/glut32.dll>) და ჩააგდეს პროექტის ფოლდერში (Desktop->OpenGL Projects->OpenGL_base1).
4. აკრიფეთ შემდეგ გვერდზე მოყვანილი ბაზური პროგრამის ტექსტი და გაუშვით შესრულებაზე (F9).

```
#include <GL/freeglut.h> თუ DEV++-ში freeglut-ის ნაცვლად აირჩიეთ GLUT-ი მაშინ აქ უნდა ეწეროს
#include <GL/glut.h>
#include <stdlib.h> აქ განსაზღვრულია exit(0) ფუნქცია
```

void init(void) { ვიძახებთ ერთხელ პროგრამის ჩათვითისას და ვაყენებთ OpenGL-ის საწყის მდგომარეობებს.

```
glClearColor(1, 1, 1, 1); ვირჩევთ ეკრანის გასუფთავების ფერს (RGBA) - თეთრი
}
```

void display(void) { ეს ფუნქცია გამოიძახება ყოველი ეკრანის გადახატვის დროს. აქ ვწერთ ჩვენი ხატვის კოდს.

```
glClear (GL_COLOR_BUFFER_BIT) ; გასუფთავებთ ეკრანს

glColor3f(0,0,0); მიმდინარე ხატვის ფერად ვაყენებთ შავს
glBegin(GL_QUADS); ვხატავთ გვადრატს
glVertex2f(-1,-1); 1-ი წვეროს კოორდინატა X,Y
glVertex2f(1,-1); 2-ე წვეროს კოორდინატა X,Y
glVertex2f(1,1); 3-ე წვეროს კოორდინატა X,Y
glVertex2f(-1,1); 4-ე წვეროს კოორდინატა X,Y
glEnd(); მოვრჩით გეომეტრიული ინფორმაციის შეტანას

glutSwapBuffers(); ვუგზავნით OpenGL-ს დასახატად
}
```

void reshape(int w, int h) { ეს ფუნქცია გამოიძახება ეკრანის ზომების შეცვლის დროს. აქ ხდება კამერის და ფრუსტუმის, ხედვის(მოდელ-ვიუ) და პროექციის მატრიცების დაყენება. დაწვრილებით 3-ე ლექციის კონსპექტში.

```
glViewport (0, 0, (GLsizei) w, (GLsizei) h); ვირჩევთ თუ ეკრანის რა ნაწილზე გვინდა ვხატოთ. ამ
შემთხვევაში მთელი ეკრანზე ვხატავთ. 0,0, ეკრანის სიგანე, ეკრანის სიგრძე
glMatrixMode (GL_PROJECTION); მიმდინარე მატრიცად ვირჩევთ პროექციის მატრიცას
glLoadIdentity(); ვასუფთავებთ პროექციის მატრიცას
gluPerspective(60,w/h,1,500); ვირჩევთ პერსპექტიულ პროექციას.
კამერის ხედვის კუთხე (60 გრადუსი), ასპექტ რაციო ეკრანის სიგანის , სიგრძესთან შეფარდება, ახლო
მოჭრის სიბრტყე, შორი მოჭრის სიბრტყე.
gluLookAt(0.0, 0.0, 4.0, 0.0,0.0, 0.0, 0.0, 1.0, 0.0 ); კამერის პარამეტრები. 1-ი სამეული (x,y,z) ესაა კამერის
პოზიცია, 2-ე კამერის ფოკუსის წერტილი ანუ რას უყურებს კამერა და 3-ე კამერის ორიენტაცია.
glMatrixMode(GL_MODELVIEW); მიმდინარე მატრიცად ვირჩევთ მოდელ-ვიუ მატრიცას
glLoadIdentity(); ვასუფთავებთ მიმდინარე მატრიცას
}
```

void myKeyboard(unsigned char key, int x, int y) { აქ ხდება კლავიატურის კლავიშების დამუშავება

```
switch (key) { key-ში გადმოცემა დაჭერილი კლავიშის კოდი
case 27: 27- ესაა ESC-ის კოდი
exit(0); ESC -ზე დაჭერის შემთხვევაში ვხურავთ პროგრამას
break;
}
}
```

```
int main(int argc, char** argv) { პროგრამის შესვლის წერტილი
glutInit(&argc, argv); GLUT-ის ბიბლიოთეკის ინიციალიზაცია
glutInitDisplayMode (GLUT_DOUBLE | GLUT_DEPTH); ვქმნით კადრის და სიღრმის ბუფერებს
glutInitWindowSize (400, 400); ვირჩევთ ფანჯრის სიგრძეს და სიგანეს
glutInitWindowPosition (100, 100); ვირჩევთ ფანჯრის საწყის პოზიციას
glutCreateWindow ("Base 1.0"); ვქმნით ფანჯარას და გადაცემთ ფანჯრის სათაურს
init(); ჩვენი ინიციალიზაციის ფუნქცია
glutDisplayFunc(display); აქ ვირჩევთ ჩვენი ხატვის ფუნქციას
glutReshapeFunc(reshape); აქ ვირჩევთ, რა ფუნქცია გამოიძახოს ეკრანის ზომის შეცვლის დროს
glutKeyboardFunc(myKeyboard); აქ ვირჩევთ კლავიატურის კლავიშების დამუშავებელ ფუნქციას
glutIdleFunc(display); აქ ვირჩევთ, რა ფუნქცია გამოიძახოს თუ სხვა საქმე არ აქვს აპლიკაციას
glutMainLoop(); უსასრულო ციკლი რათა აპლიკაცია არ მორჩეს და არ დაიხუროს
return 0;
}
```