**Runnable example**

Each time you click with the mouse on the window, a tmpX.ppm file is created with the current screenshot.

You can view this file for example with eog on Linux, and inspect it with a text editor.

To render without showing a window, see: [How to use GLUT/OpenGL to render to a file?](http://stackoverflow.com/questions/3191978/how-to-use-glut-opengl-to-render-to-a-file)

#include <math.h>

#include <stdlib.h>

#include <stdio.h>

#define GL\_GLEXT\_PROTOTYPES 1

#include <GL/gl.h>

#include <GL/glu.h>

#include <GL/glut.h>

#include <GL/glext.h>

static GLubyte \*pixels = NULL;

static const GLenum FORMAT = GL\_RGBA;

static const GLuint FORMAT\_NBYTES = 4;

static const unsigned int HEIGHT = 500;

static const unsigned int WIDTH = 500;

static unsigned int nscreenshots = 0;

static unsigned int time;

/\* Model. \*/

static double angle = 0;

static double angle\_speed = 45;

static void init(void) {

glReadBuffer(GL\_BACK);

glClearColor(0.0, 0.0, 0.0, 0.0);

glPixelStorei(GL\_UNPACK\_ALIGNMENT, 1);

glViewport(0, 0, WIDTH, HEIGHT);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

glMatrixMode(GL\_MODELVIEW);

pixels = malloc(FORMAT\_NBYTES \* WIDTH \* HEIGHT);

time = glutGet(GLUT\_ELAPSED\_TIME);

}

static void deinit(void) {

free(pixels);

}

static void create\_ppm(char \*prefix, int frame\_id, unsigned int width, unsigned int height,

unsigned int color\_max, unsigned int pixel\_nbytes, GLubyte \*pixels) {

size\_t i, j, k, cur;

enum Constants { max\_filename = 256 };

char filename[max\_filename];

snprintf(filename, max\_filename, "%s%d.ppm", prefix, frame\_id);

FILE \*f = fopen(filename, "w");

fprintf(f, "P3\n%d %d\n%d\n", width, HEIGHT, 255);

for (i = 0; i < height; i++) {

for (j = 0; j < width; j++) {

cur = pixel\_nbytes \* ((height - i - 1) \* width + j);

fprintf(f, "%3d %3d %3d ", pixels[cur], pixels[cur + 1], pixels[cur + 2]);

}

fprintf(f, "\n");

}

fclose(f);

}

static void draw\_scene() {

glClear(GL\_COLOR\_BUFFER\_BIT);

glLoadIdentity();

glRotatef(angle, 0.0f, 0.0f, -1.0f);

glBegin(GL\_TRIANGLES);

glColor3f(1.0f, 0.0f, 0.0f);

glVertex3f( 0.0f, 0.5f, 0.0f);

glColor3f(0.0f, 1.0f, 0.0f);

glVertex3f(-0.5f, -0.5f, 0.0f);

glColor3f(0.0f, 0.0f, 1.0f);

glVertex3f( 0.5f, -0.5f, 0.0f);

glEnd();

}

static void display(void) {

draw\_scene();

glutSwapBuffers();

glReadPixels(0, 0, WIDTH, HEIGHT, FORMAT, GL\_UNSIGNED\_BYTE, pixels);

}

static void idle(void) {

int new\_time = glutGet(GLUT\_ELAPSED\_TIME);

angle += angle\_speed \* (new\_time - time) / 1000.0;

angle = fmod(angle, 360.0);

time = new\_time;

glutPostRedisplay();

}

void mouse(int button, int state, int x, int y) {

if (state == GLUT\_DOWN) {

puts("screenshot");

create\_ppm("tmp", nscreenshots, WIDTH, HEIGHT, 255, FORMAT\_NBYTES, pixels);

nscreenshots++;

}

}

int main(int argc, char \*\*argv) {

GLint glut\_display;

glutInit(&argc, argv);

glutInitWindowSize(WIDTH, HEIGHT);

glutInitWindowPosition(100, 100);

glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGBA);

glutCreateWindow(argv[0]);

init();

glutDisplayFunc(display);

glutIdleFunc(idle);

glutMouseFunc(mouse);

atexit(deinit);

glutMainLoop();

return EXIT\_SUCCESS;

}

<http://stackoverflow.com/questions/3191978/how-to-use-glut-opengl-to-render-to-a-file>

https://eonstrife.wordpress.com/2007/06/02/taking-a-screenshot-from-an-opengl-application/