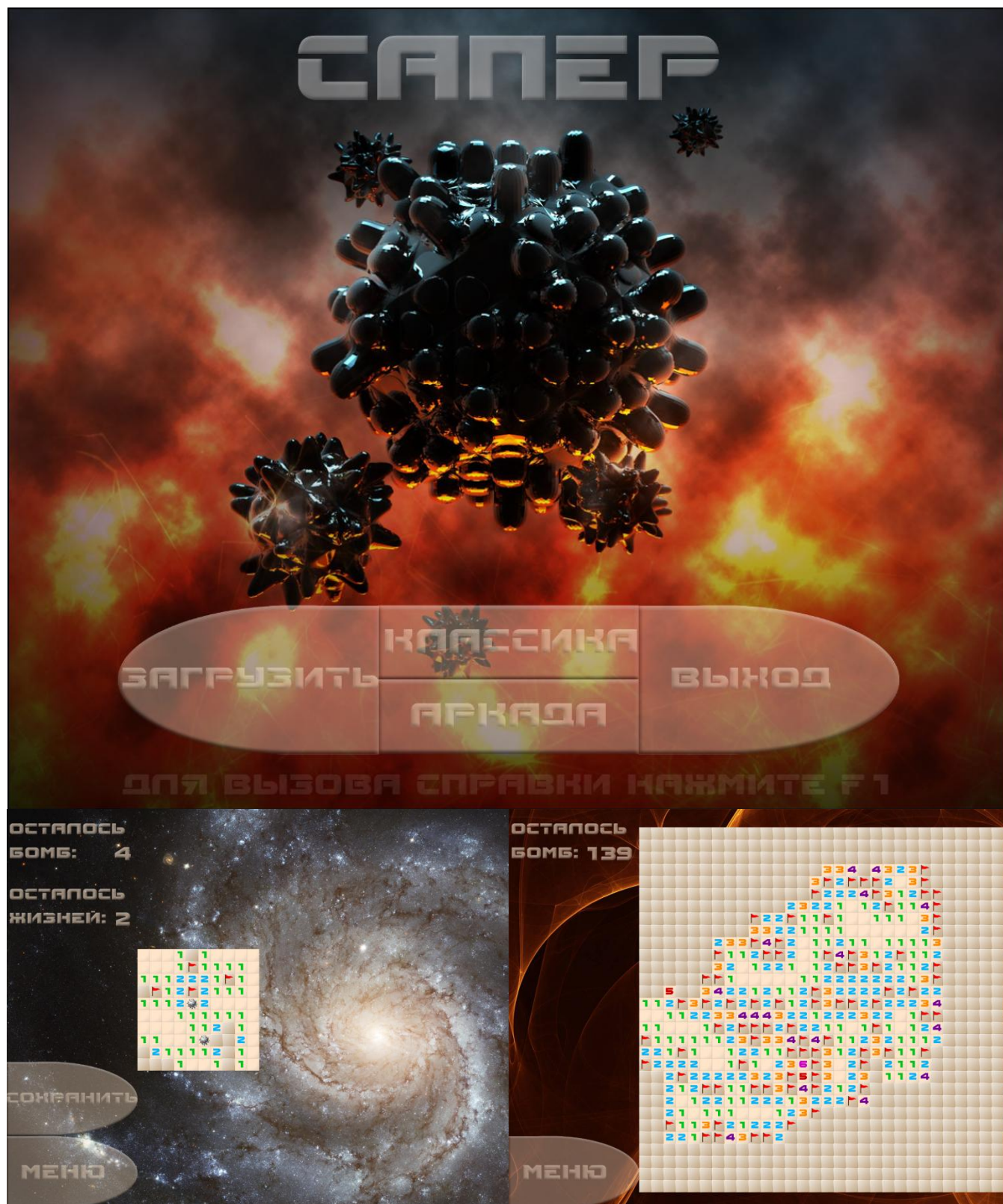


Конкурс творческих проектов
в области информационных технологий
для детей и молодёжи
«Наш 21 век»

Номинация: Программирование

Игра «САПЕР»



Проект выполнил: Мельник Максим Юрьевич, 26.06.97 г.р., ученик «Школы информатики» Муниципального Образовательного Учреждения Дополнительного Образования «Дом Детского Творчества» г. Вилючинска Камчатского края.

Этот проект выполнен на языке программирования **Pascal** под руководством педагога дополнительного образования Романенко Игоря Николаевича.

При работе над проектом были использованы:

- Опыт работы в **ОС Windows**
- Опыт работы в текстовом редакторе **Notepad++**
- Опыт работы в графическом редакторе **Photoshop CS3**
- Опыт работы с компилятором **Free Pascal Compiler (FPC)**
- Опыт работы в командной строке
- Библиотека **Bass**
- Библиотека **OpenGL**
- Библиотека **GLFW**

Обоснование проекта:

Проект представляет собой компьютерную игру «Сапер». Главными целями работы были обучение программированию и создание красочной версии игры «Сапер» с использованием возможностей трехмерной графики.

Для создания была выбрана игра «Сапер», потому что она развивает логическое мышление, внимательность, является известной и популярной во всем мире, но при этом имеет только оконный интерфейс. Также «Сапер» - одна из моих любимых игр.

В ходе работы я углубил свои знания в языке **Pascal** и научился создавать графический игровой интерфейс при помощи **OpenGL**.

Библиотеки, использованные при создании:

OpenGL – это программный интерфейс к графической аппаратуре. Этот интерфейс состоит приблизительно из 250 отдельных команд (около 200 команд в самой **OpenGL** и еще 50 в библиотеке утилит), которые используются для указания объектов и операций, которые необходимо выполнить, чтобы получить интерактивное приложение, работающее с трехмерной графикой.

GLFW - это свободная мультиплатформенная библиотека с открытым исходным кодом для создания окон, контекста **OpenGL** и обработки различных событий в оконном приложении. Эта библиотека имеет встроенную поддержку для Windows, Mac OS X и многих Unix-подобных систем, использующих X Window System, такие как Linux и FreeBSD.

BASS (BASS audio library) - это небольшая аудио-библиотека, которую можно использовать в операционных системах Windows, Mac OS X, Linux, Win64, WinCE, Android и iOS. Цель библиотеки заключается в том, чтобы предоставить разработчикам образец аудио-центра, в частности, обеспечить работу с потоками (MP3, MP2, MP1, OGG, WAV, AIFF), MOD файлами (XM, IT, S3M, MOD, MTM, UMX), MO3 файлами (MP3, OGG и сжатых MOD файлов), функциями записи, редактирования тегов аудиофайлов, конвертирования популярных форматов.

dglOpenGL – это заголовочный файл, включающий в себя все функции **OpenGL** и **GLU**. Он постоянно обновляется, чтобы обеспечить устойчивую поддержку новых возможностей **OpenGL**.

Ход выполнения работы:

1. Разработка идеи, определение целей и задач
2. Создание игрового движка
3. Работа с текстурами
4. Работа со звуком

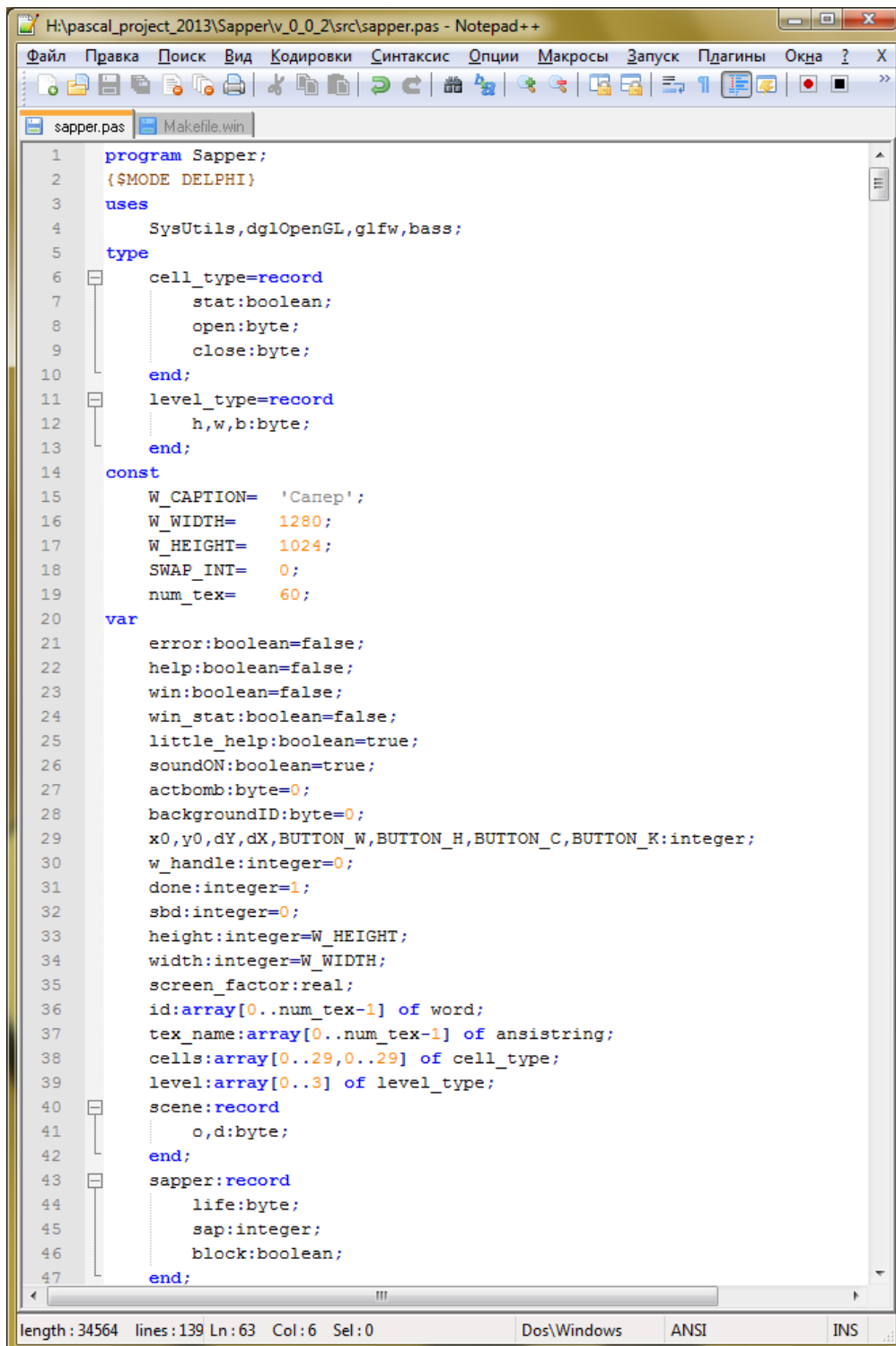
1 этап. Разработка идеи, определение целей и задач.

Главной идеей было обучение программированию в ходе создания игры. Чтобы игра имела преимущества над оригиналом, в неё был добавлен аркадный режим, внешний вид программы был переведен в полноэкранный режим, для вывода графики используется библиотека **OpenGL**. Затем для достижения результата работа была разбита на этапы.

2 этап. Создание игрового движка.

В моем проекте существует три сцены: сцена **Меню**, сцена **Классического режима**, сцена **Аркадного режима**.

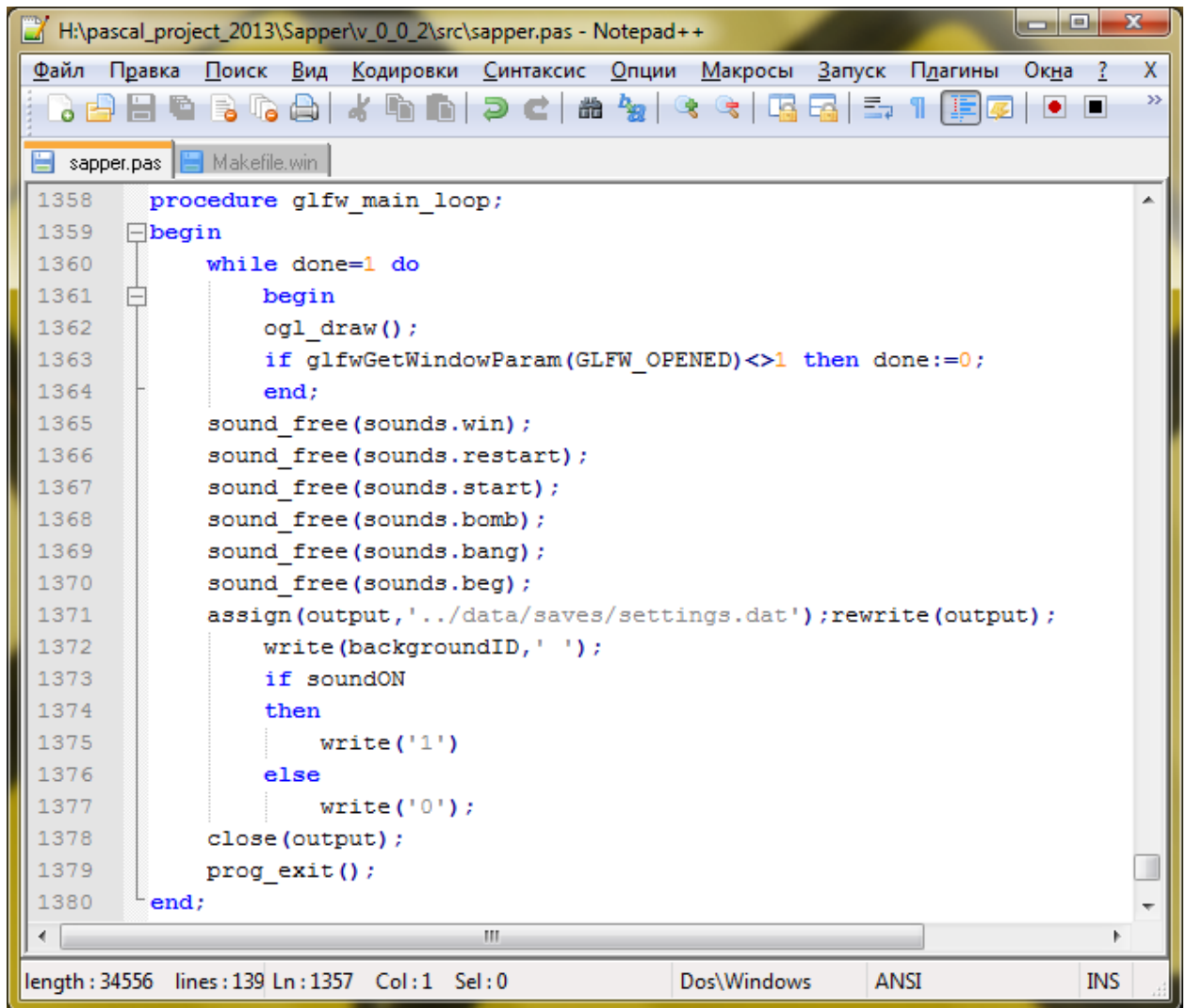
Вначале я подключил нужные модули, объявил типы, константы и переменные.



```
1  program Sapper;
2  {$MODE DELPHI}
3  uses
4      SysUtils, dglOpenGL, glfw, bass;
5  type
6      cell_type=record
7          stat:boolean;
8          open:byte;
9          close:byte;
10     end;
11     level_type=record
12         h,w,b:byte;
13     end;
14  const
15      W_CAPTION= 'Сапер';
16      W_WIDTH= 1280;
17      W_HEIGHT= 1024;
18      SWAP_INT= 0;
19      num_tex= 60;
20  var
21      error:boolean=false;
22      help:boolean=false;
23      win:boolean=false;
24      win_stat:boolean=false;
25      little_help:boolean=true;
26      soundON:boolean=true;
27      actbomb:byte=0;
28      backgroundID:byte=0;
29      x0,y0,dY,dX,BUTTON_W,BUTTON_H,BUTTON_C,BUTTON_K:integer;
30      w_handle:integer=0;
31      done:integer=1;
32      sbd:integer=0;
33      height:integer=W_HEIGHT;
34      width:integer=W_WIDTH;
35      screen_factor:real;
36      id:array[0..num_tex-1] of word;
37      tex_name:array[0..num_tex-1] of ansistring;
38      cells:array[0..29,0..29] of cell_type;
39      level:array[0..3] of level_type;
40      scene:record
41          o,d:byte;
42      end;
43      sapper:record
44          life:byte;
45          sap:integer;
46          block:boolean;
47      end;
```

Затем написал процедуры программы. Сейчас я опишу ключевые из них.

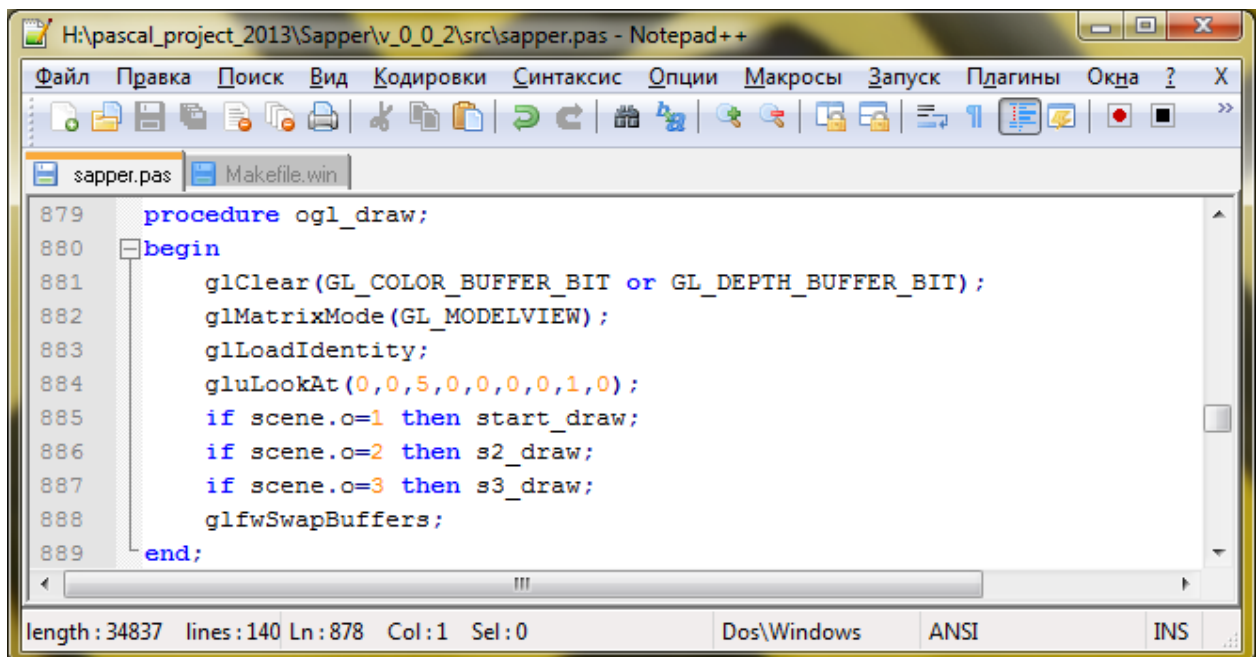
Процедура **glfw_main_loop** – основной цикл программы. Также она очищает память.



```
1358 procedure glfw_main_loop;
1359 begin
1360     while done=1 do
1361     begin
1362         ogl_draw();
1363         if glfwGetWindowParam(GLFW_OPENED)<>1 then done:=0;
1364     end;
1365     sound_free(sounds.win);
1366     sound_free(sounds.restart);
1367     sound_free(sounds.start);
1368     sound_free(sounds.bomb);
1369     sound_free(sounds.bang);
1370     sound_free(sounds.beg);
1371     assign(output, '../data/saves/settings.dat');rewrite(output);
1372     write(backgroundID, ' ');
1373     if soundON
1374     then
1375         write('1')
1376     else
1377         write('0');
1378     close(output);
1379     prog_exit();
1380 end;
```

length: 34556 lines: 139 Ln: 1357 Col: 1 Sel: 0 Dos\Windows ANSI INS

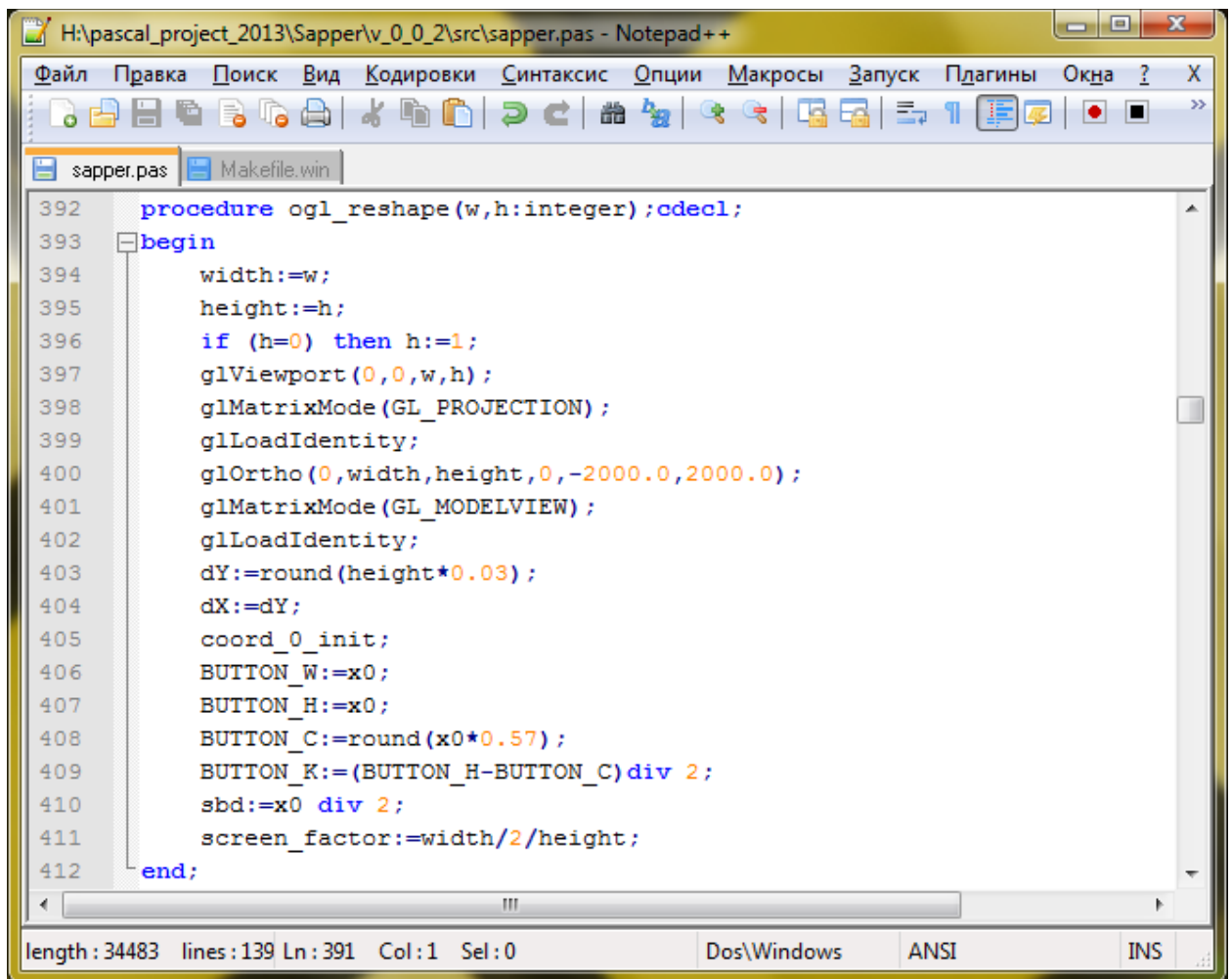
Процедура **ogl_draw** – основной цикл визуализации. Также в нем происходит переключение сцен.



```
879 procedure ogl_draw;
880 begin
881     glClear(GL_COLOR_BUFFER_BIT or GL_DEPTH_BUFFER_BIT);
882     glMatrixMode(GL_MODELVIEW);
883     glLoadIdentity;
884     gluLookAt(0,0,5,0,0,0,0,1,0);
885     if scene.o=1 then start_draw;
886     if scene.o=2 then s2_draw;
887     if scene.o=3 then s3_draw;
888     glfwSwapBuffers;
889 end;
```

length: 34837 lines: 140 Ln: 878 Col: 1 Sel: 0 Dos\Windows ANSI INS

Процедура **ogl_reshape** – благодаря ней программа подстраивается под разрешение экрана автоматически, изменяя размер всех элементов и не допуская искажения текстур.

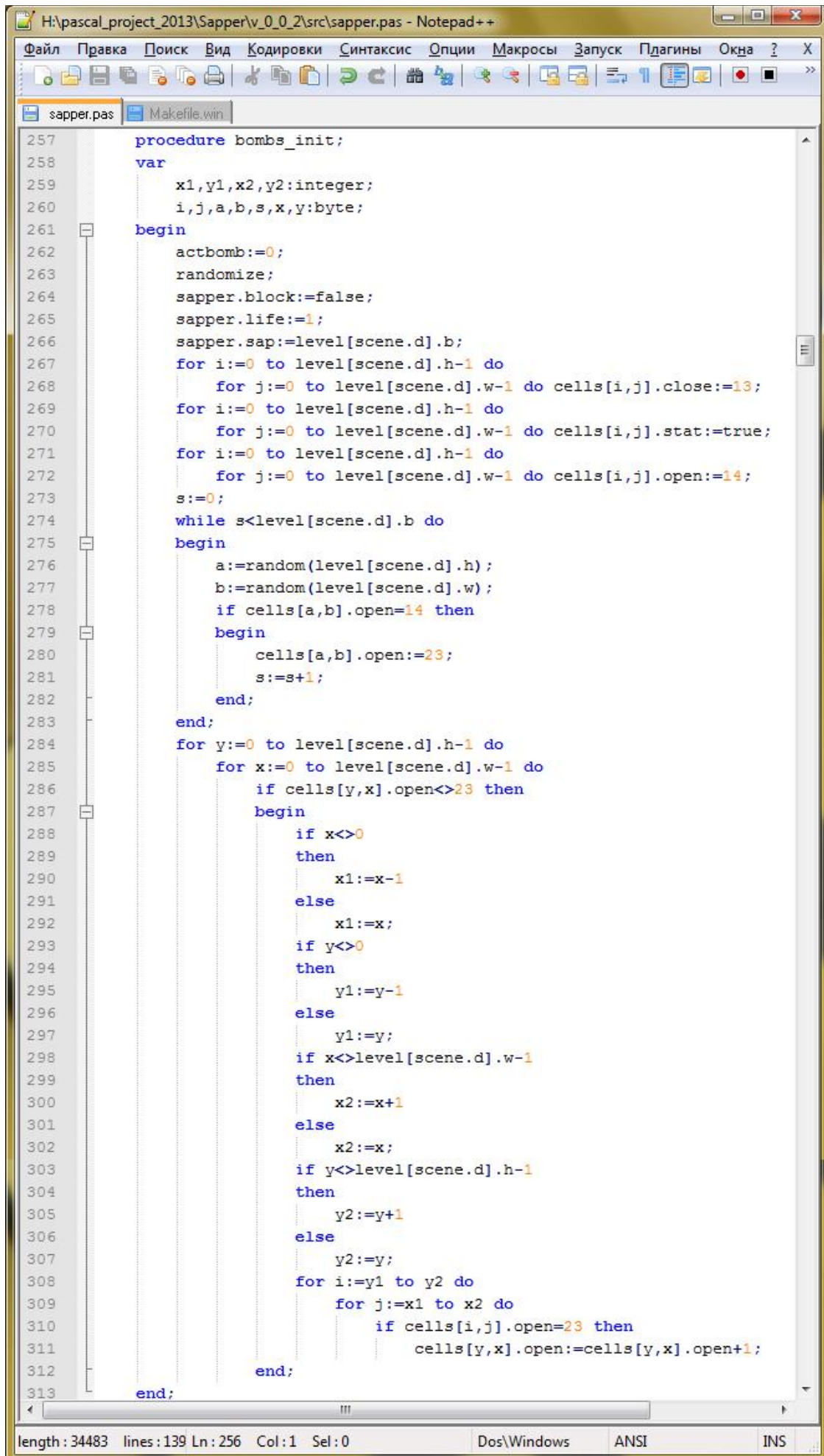


The screenshot shows a Notepad++ window with the title bar "H:\pascal_project_2013\Sapper\v_0_0_2\src\sapper.pas - Notepad++". The menu bar includes "Файл", "Правка", "Поиск", "Вид", "Кодировки", "Синтаксис", "Опции", "Макросы", "Запуск", "Плагины", "Окна", and "?". The toolbar contains various icons for file operations and editing. The editor shows two tabs: "sapper.pas" and "Makefile.win". The "sapper.pas" tab is active, displaying Pascal code for the `ogl_reshape` procedure. The code is as follows:

```
392 procedure ogl_reshape(w,h:integer); cdecl;  
393 begin  
394     width:=w;  
395     height:=h;  
396     if (h=0) then h:=1;  
397     glViewport(0,0,w,h);  
398     glMatrixMode(GL_PROJECTION);  
399     glLoadIdentity;  
400     glOrtho(0,width,height,0,-2000.0,2000.0);  
401     glMatrixMode(GL_MODELVIEW);  
402     glLoadIdentity;  
403     dY:=round(height*0.03);  
404     dX:=dY;  
405     coord_0_init;  
406     BUTTON_W:=x0;  
407     BUTTON_H:=x0;  
408     BUTTON_C:=round(x0*0.57);  
409     BUTTON_K:=(BUTTON_H-BUTTON_C) div 2;  
410     sbd:=x0 div 2;  
411     screen_factor:=width/2/height;  
412 end;
```

The status bar at the bottom shows "length: 34483 lines: 139 Ln: 391 Col: 1 Sel: 0", "Dos\Windows", "ANSI", and "INS".

Процедура **bombs_init** заполняет случайные ячейки бомбами. Также заполняет ячейки цифрами, указывающими количество прилежащих бомб.



```
257 procedure bombs_init;
258 var
259     x1,y1,x2,y2:integer;
260     i,j,a,b,s,x,y:byte;
261 begin
262     actbomb:=0;
263     randomize;
264     sapper.block:=false;
265     sapper.life:=1;
266     sapper.sap:=level[scene.d].b;
267     for i:=0 to level[scene.d].h-1 do
268         for j:=0 to level[scene.d].w-1 do cells[i,j].close:=13;
269     for i:=0 to level[scene.d].h-1 do
270         for j:=0 to level[scene.d].w-1 do cells[i,j].stat:=true;
271     for i:=0 to level[scene.d].h-1 do
272         for j:=0 to level[scene.d].w-1 do cells[i,j].open:=14;
273     s:=0;
274     while s<level[scene.d].b do
275     begin
276         a:=random(level[scene.d].h);
277         b:=random(level[scene.d].w);
278         if cells[a,b].open=14 then
279         begin
280             cells[a,b].open:=23;
281             s:=s+1;
282         end;
283     end;
284     for y:=0 to level[scene.d].h-1 do
285         for x:=0 to level[scene.d].w-1 do
286             if cells[y,x].open<>23 then
287             begin
288                 if x<>0
289                 then
290                     x1:=x-1
291                 else
292                     x1:=x;
293                 if y<>0
294                 then
295                     y1:=y-1
296                 else
297                     y1:=y;
298                 if x<>level[scene.d].w-1
299                 then
300                     x2:=x+1
301                 else
302                     x2:=x;
303                 if y<>level[scene.d].h-1
304                 then
305                     y2:=y+1
306                 else
307                     y2:=y;
308                 for i:=y1 to y2 do
309                     for j:=x1 to x2 do
310                         if cells[i,j].open=23 then
311                             cells[y,x].open:=cells[y,x].open+1;
312             end;
313     end;
```

length: 34483 lines: 139 Ln: 256 Col: 1 Sel: 0 Dos\Windows ANSI INS

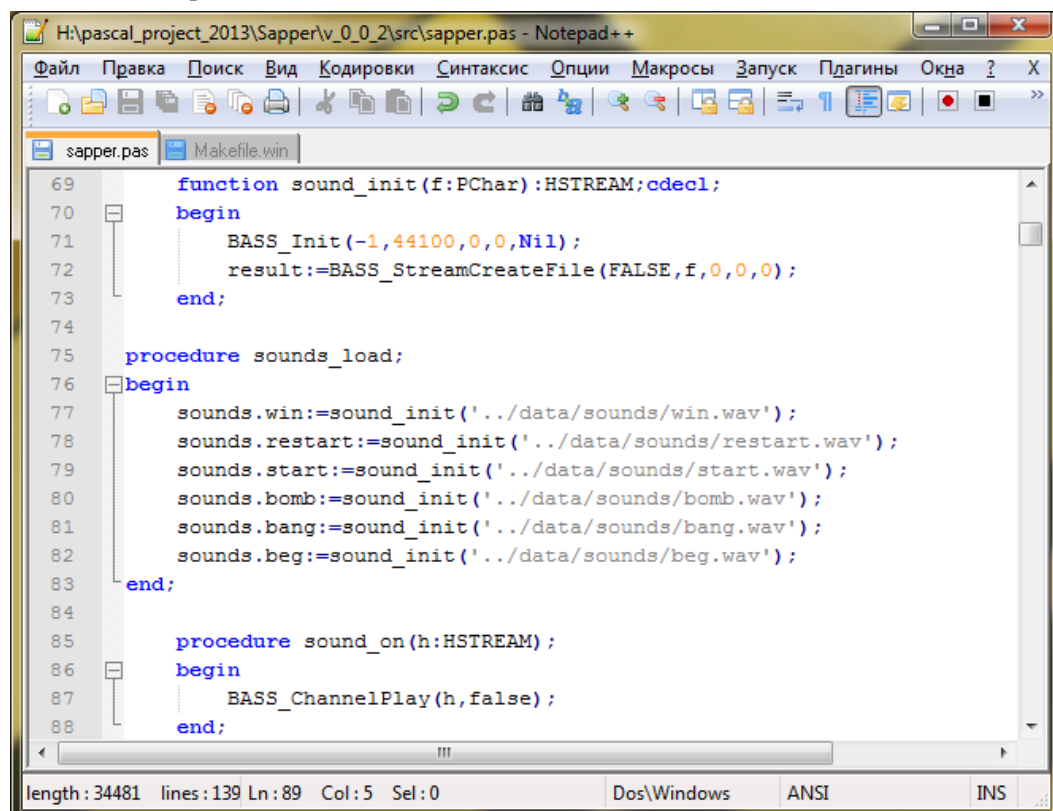
3 этап. Работа с текстурами.

В графическом редакторе Adobe **Photoshop CS3** я создал различные текстуры кнопок, ячеек, окон помощи и информативных элементов. Сохранил их в формате **Targa** (*.TGA). В процедуре **Load_textures** загрузил текстуры в программу.



4 этап. Работа со звуком.

В конце работы я добавил в игру звук. Выбрав подходящие звуки, я написал процедуру загрузки и добавил команды на их проигрывание после определенных моментов в игре, например, после победы или взрыва бомбы.



Заключение:

В ходе работы над проектом я глубоко изучил возможности языка программирования Pascal, а также возможности его интеграции с различными дополнительными библиотеками: OpenGL, GLFW, Bass и т.д.

Я создал игровое приложение «Сапер», развивающее логическое мышление и внимательность и имеющее полноэкранный интерфейс.

Данный проект показывает область практического применения языка программирования Pascal для создания игровых приложений.

В дальнейшем я планирую продолжать совершенствовать свой проект. В перспективе развитие аркадного режима, создание трехмерного режима, добавление анимации и других визуальных эффектов.