MkBGFire v0.9.5

Algorithm for the fire model: **Pencho Marinov**
Application interface and visuals: **Nikolay Ikonomov**

March 15, 2019

Contents

MkBGFire is an application for creating game models of forest fires in Bulgaria, the full name is “Make Bulgarian Fire”.

1 Application

1.1 Running the application

Download the application from `http://justmathbg.info/mkbgfire.html` and extract it to the hard drive of the computer. Start the application by `MkBGFire.exe` (32-bit) and `MkBGFire64.exe` (64-bit) – requires Windows 7 and later, `MkBGFire64.sh` – requires Linux Mint, Ubuntu Linux, and compatible 64-bit operating system.

1.2 Using the application

Start MkBGFire. Click on the button “New Forest”. From the drop-down menu select the desired vegetation type, use the left mouse button to draw in the forest. Fire start cells must be marked in order to enable the fire. Click on the button “Make Fire Model” to create the fire model. Use the slider below the button to view the time steps.

Vegetation types include Water, Pine, Oak, Grass, Paliurus, also the menu provides the Fire start option. A new forest is initialized with grass only. Hover with the mouse over cells to display information about each cell in the panel below the slider.

The user interface consists of five separate panels. The top-left one contains the menu items, directly below it is the *forest panel*, next is the *view panel*, finally – the *model panel*. The center panel contains the OpenGL display, where the forest is visualized.

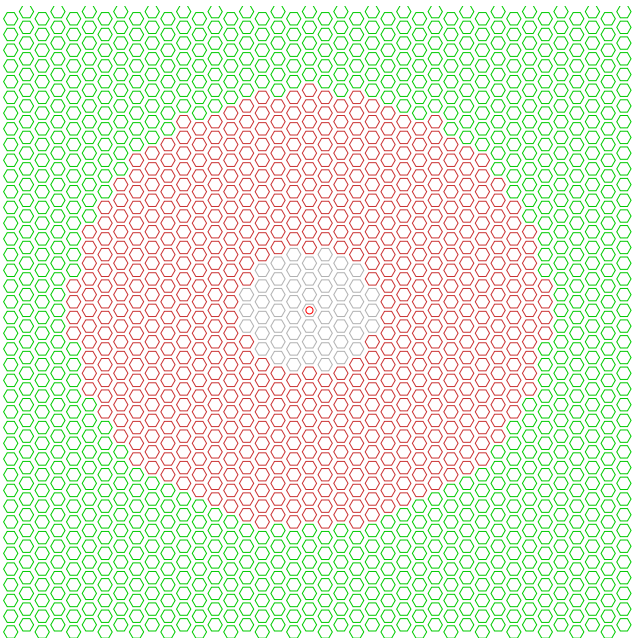

Figure 1: The hexagons denote grass, the green ones – normal state of the vegetation, red ones – on fire, grey ones – ashes, the vegetation has thoroughly burnt out. The red circle in the center is a cell marked with fire start, this is the origin of the fire in this case. This is the 16-th step of the algorithm, corresponds to the screen shot of the application, provided as `ver094.png`.

The forest panel: Select the size of the square matrix, from 40 to 200 cells. Next, select the cell type, either normal squares or simulated hexagons, the internal representation is always in a square matrix.

The button “New Forest” creates the forest with grass only and displays it in the center panel, see Fig. 1. Select the vegetation type and drag the mouse over the display.

The number of affected cells in the forest can be chosen by the brush size. Force drawing in the forest by the "Draw button" or by switching vegetation type.

The view panel: "Clear sky" toggles the display background. "Directions" toggles the visualization of the OpenGL internal axes.

"Rotate" enables rotation of the display with the left and right mouse buttons, also enables zoom of the display by the mouse scroll button. Rotation speed can be chosen by "Angle". "Lock" prevents rotation. "Reset" restores the default view of the display.

The model panel: Select wind speed, this is a relative parameter. Select wind angle, where 0 means North, 90 – East, 180 – South, 270 – West, relative to compass directions. Finally, select the time steps for the algorithm. The button "Make Fire Model" calls the algorithm with these parameters. After that, use the slider below the button to view the time steps in the center panel.

Cell information is displayed below the slider, below that is displayed the last message by the application. All messages can be viewed from menu File → Message Log.

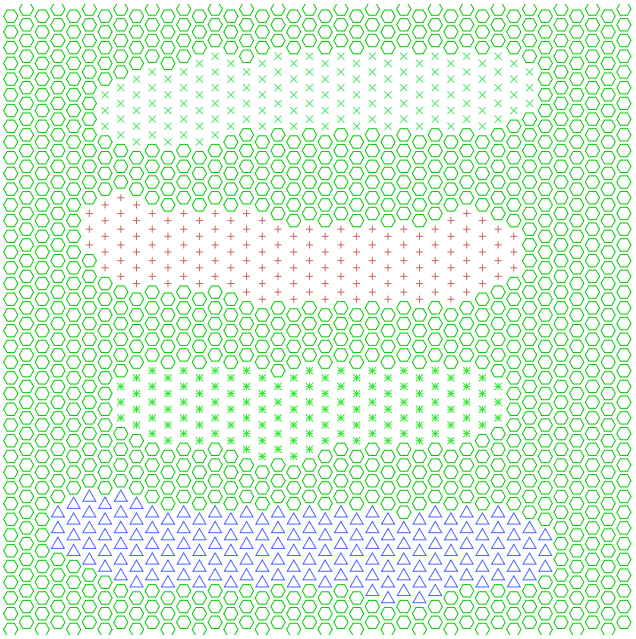

Figure 2: Demonstration of vegetation types: × pine, + oak, ∗ paliurus, △ water. There is also color for each type: pine green, oak brown, paliurus light green, water blue.

Menu File: "Open File" can open an existing forest `*.mff` data file. "Save File" saves this forest as `*.mff` data file. "Save PNG" makes a screen shot of the forest as a `*.png` image file.

"Export to PS" can save selected time steps into PostScript format, which is a vector format, useful for importing into LaTeX and simply creating high-resolution images.

Finally, there are "Message Log" and "Exit" options. All menu items can be invoked by the Control button on the keyboard and the respective key mentioned, Ctrl-Q exits the application.

Menu Forest: This menu exists to bind the keyboard shortcuts to the respective buttons in the user interface: "New Forest", "Draw in the Forest", "Reset", "Lock", "Rotate".

Vegetation type can be switched to by Ctrl-1 to Ctrl-6, see Fig. 2

Menu Model: First menu item binds Ctrl-M to the action "Make Fire Model". Second and third menu items display options for the algorithm.

Parameters: Fine tuning of the algorithm:

- Parameters Rka and Rkb – look ahead/angled look of fire front, determine the shape of the fire front.
- Parameters Lpa and Lpb – the fire start cell and the cell to burn are connected by a line, parameter Lpa is the length of the normals around that line, that length may include several cells of the forest, parameter Lpb is the length of the line itself.
- Parameter Lpp – the fire front "jumps through / skips" this amount of non-burnable cells, like water, value of 0 means that the water is an impenetrable obstruction.

Presets can be saved and loaded, for efficient testing of the algorithm.

Options: Save the output of the algorithm in the old format, when "Make Fire Model" action is invoked, saved as `*.mfp` text file.

Menu Help: About the application.

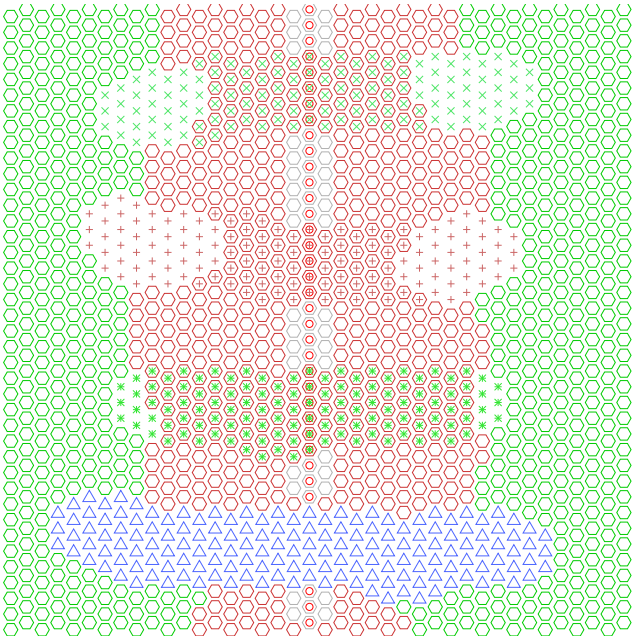

Figure 3: Demonstration of applying fire to different vegetation types, the forest corresponds to Fig. 2. A line of fire has been drawn vertically through the forest. The vegetation on fire has a red outline. Some patches of the vegetation left and right from the center have not caught on fire yet, they have no outline so far. This is the 10-th step of the algorithm.

1.3 File format for the application

The application saves and opens a file with extension `*.mff`. This is a normal text file, which can be opened and edited with Notepad++ and similar text editors.

The first three lines contain: the version number of the application, v0.9.5 as 95, the number of rows and columns of the forest. Each line after that contains information about one cell of the forest, which is 10 values separated by a space. This version reads only the first four values of each row of the file, the other six are for forward compatibility, meaning that future versions can save and read at most 10 parameters for each cell.

Cell type is not saved in the file. This way one forest can be loaded as square and hexagon cells, for comparing the difference between the two cell types.

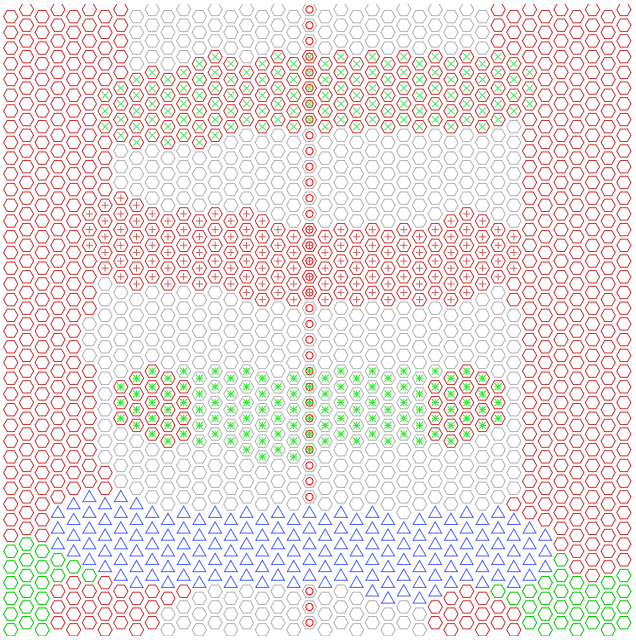

Figure 4: The grass in the center has already burned out. Same applies to part of the paliurus, third row. This is the 20-th step of the algorithm.

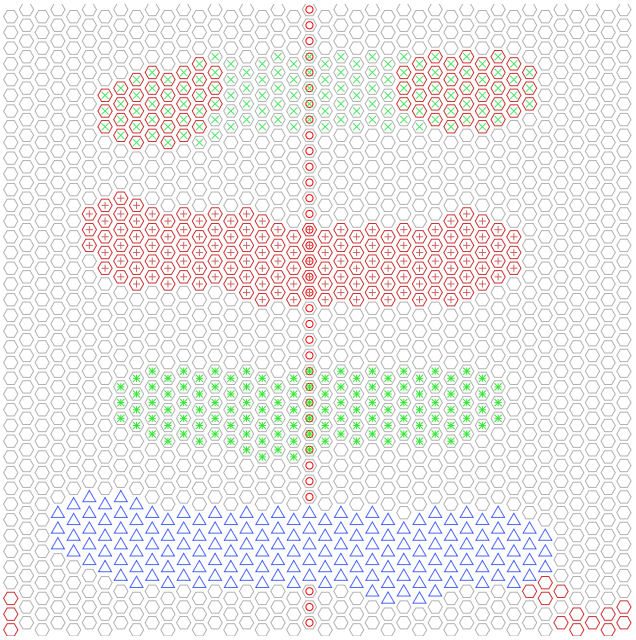

Figure 5: The grass in the forest has mostly burned out, has some burning in the south part. The paliurus has thoroughly burned out, denoted by grey outline. Same applies for part of the pines. The oaks are still burning. This is the 30-th step of the algorithm.

Now it is a good time to mention the constants used: *resistance to fire* is reduced to 0 by each time step (default visuals), after that the state of the cell is changed to burning and the *vegetation fuel* is reduced to 0 by each time step (red outline), finally the vegetation burns out (grey outline) and that is its final state. These constants are arbitrarily defined, in hope that the model looks realistic.

- Pine: 15 resist, 21 fuel
- Oak: 20 resist, 40 fuel
- Grass: 4 resist, 8 fuel
- Paliurus: 7 resist, 12 fuel
- Water: -1 resist, -1 fuel

2 Technical information

2.1 OpenGL

This application complies with OpenGL 1.1. See the OpenGL 2.1 Reference Pages for command details: `https://www.khronos.org/registry/OpenGL-Refpages/gl2.1/`

The following commands are used:

```
glBegin(GL_LINE_LOOP)
- GL_POINTS, GL_LINES, GL_LINE_STRIP, GL_LINE_LOOP
- GL_TRIANGLES, GL_TRIANGLE_STRIP, GL_TRIANGLE_FAN
- GL_QUADS, GL_QUAD_STRIP, GL_POLYGON
glClear(GL_COLOR_BUFFER_BIT)
- GL_COLOR_BUFFER_BIT, GL_DEPTH_BUFFER_BIT
- GL_ACCUM_BUFFER_BIT, GL_STENCIL_BUFFER_BIT
glClearColor(float red, float green, float blue, float alpha)
glColor3f(float red, float green, float blue)
glEnd()
glLoadIdentity()
glRotatef(float angle, float x, float y, float z)
glScalef(float x, float y, float z)
glVertex2f(float x, float y)
glVertex3f(float x, float y, float z)
glViewport(int x, int y, int w, int h)
```

2.2 PostScript

The application can export each time step of the algorithm to PostScript format. Install GhostScript `https://ghostscript.com/` to be able to view and convert these files.

- View a PostScript file:

  `gswin64 file.ps`

- Convert to PDF format in the console, otherwise use `gswin64`:

  `gswin64c -sDEVICE=pdfwrite -o out.pdf file.eps`

- Convert to PNG image, change image resolution by `-r500` flag:

  `gswin64c -sDEVICE=png16m -r700 -o out.png out.pdf`

- View available devices, these are the file formats supported:

  `gswin64c -h`

- Install also ImageMagick `https://www.imagemagick.org/` to convert PostScript directly to PNG, change image resolution by `-density` flag:

  `convert -density 200 file.eps out.png`

Sources, the presentation in the first link is a really good introduction to PostScript:
`https://sus.ziti.uni-heidelberg.de/Lehre/WS1617_Tools/POSTSCRIPT/PostScript_PeterFischer.pptx.pdf`
`https://stackoverflow.com/questions/785436/ghostscript-command-line-parameters-to-convert-eps-to-pdf`
`http://www.unitconversion.org/typography/postscript-points-to-pixels-x-conversion.html`
`http://www.math.ubc.ca/~cass/courses/ps.html`
`http://www.ugrad.math.ubc.ca/Flat/paths-ref.html`
`http://www.ugrad.math.ubc.ca/Flat/setup-ref.html`

2.3 Credits

Algorithm for the fire model developed by Pencho Marinov, Institute of Information and Communication Technologies at the Bulgarian Academy of Sciences.

Application user interface on FLTK, and cell visuals on OpenGL, by Nikolay Ikonomov, Institute of Mathematics and Informatics at the Bulgarian Academy of Sciences.

MkBGFire is based in part on the work of the FLTK project (`http://www.fltk.org`).

2.4 Changelog

- Version 0.9.5 (March 15, 2019)
  - Time steps can be switched by page up/down
  - Slider can be activated by fire model
- Version 0.9.4 (February 17, 2019)
  - Added export to PostScript
  - Forest 2D visuals redesigned
  - User interface redesigned
- Version 0.9.3 (November 18, 2018)
  - Main functions: create forest and make fire model
  - Vegetation type can be selected, when drawing in the forest
  - Added brush size for faster drawing
  - Forest can be saved as data file and image file
  - Added option for saving the fire model in the old format
  - Forest visuals improved, both in 2D and 3D
  - User interface improved
- Version 0.9.2 (October 21, 2018)
  - Algorithm improved significantly
  - Fire start can be selected with the mouse
  - Cell data can be inspected
  - User interface redesigned
- Version 0.9.1 (September 1, 2018)
  - Algorithm can be selected
  - Terrain can be switched between drawing and viewing
- Version 0.9 (August 16, 2018)
  - Terrain can be created
  - Water can be drawn on the terrain with the left mouse button
  - Hexagons are rotated back, to correspond with the algorithm
- Version 0.7 (August 14, 2018)

  - Hexagons are drawn rotated by 90 degrees, to correspond with the algorithm
  - Drawing shapes are extended to have three dimensions
  - The view can be rotated and scaled with the mouse and the buttons
  - Two toggle options are added, to show the axes and to fill the shapes
- Version 0.5 (August 10, 2018)
  - Initial version

3 Technical information for v0.9.3 and later

Instructions for compiling and linking of the application.

3.1 Linux – dynamic linking

Repository `universe` contains the FLTK package.

```
# sudo add-apt-repository universe
# sudo apt install g++ gcc libc6-dev libfltk1.3 libfltk-gl1.3 \
    libfltk1.3-dev libgl1-mesa-dev libpng-dev zlib1g-dev
# c++ mfiremodel.c mgldraw.cxx minterface.cxx -o MkBGFire64.sh \
    -lfltk -lfltk_gl -lGL -lpng -lz -s -O1
# ./MkBGFire64.sh
```

3.2 Linux – static linking

Operating system is Linux Mint 19 64-bit and compatible.

FLTK: `http://www.fltk.org/software.php`

```
# sudo apt install g++ gcc libc6-dev \
    libgl1-mesa-dev libglu1-mesa-dev freeglut3-dev \
    libxrender-dev libxcursor-dev libxft-dev libfontconfig1-dev
# cd ~/work/
# tar -xvf fltk-1.3.4-2-source.tar.gz
# cd fltk-1.3.4-2/
# ./configure
# make
# cd ~/work/
# mkdir -p fltk/include/ fltk/lib/
# cp -r fltk-1.3.4-2/FL/ fltk/include/
# cp -r fltk-1.3.4-2/lib/*.a fltk/lib/
```

ZLIB: `https://sourceforge.net/projects/libpng/files/zlib/`

```
# cd ~/work/
# tar -xvf zlib-1.2.11.tar.gz
# cd zlib-1.2.11/
# ./configure --static
# make
# cp zlib.h zconf.h ../fltk/include/
# cp libz.a ../fltk/lib/
```

PNGLIB: `https://sourceforge.net/projects/libpng/files/libpng12/`

```
# cd ~/work/
# tar -xvf libpng-1.2.7-config.tar.gz
# cd libpng-1.2.7-config/
# ./configure CPPFLAGS=-I../fltk/include/ LDFLAGS=-L../fltk/lib/
# make
# cp png.h pngconf.h ../fltk/include/
# cp .libs/libpng.a ../fltk/lib/
```

Compiling and linking:

```
# c++ mfiremodel.c mgldraw.cxx minterface.cxx -o MkBGFire64.sh \
    -Ifltk/include -Lfltk/lib -lfltk -lfltk_gl -lGL -ldl -lX11 -lXext \
    -lXfixes -lXrender -lXcursor -lXft -lfontconfig -lpng -lz -s -O1
# ./MkBGFire64.sh
```

3.3 Windows – static linking

Operating system is Windows 7 64-bit and any later version.

1. Download mingw-get-setup and install to `C:\work\MinGW`:
   `https://sourceforge.net/projects/mingw/files/Installer/mingw-get-setup.exe`
   Mark for installation `mingw-developer-toolkit` and `msys-base`.
   Then select Installation → Apply Changes.
   This part (1.) is architecture independent.
2. TDM-GCC: `http://tdm-gcc.tdragon.net/download`
   Download `tdm-gcc-5.1.0-3.exe` for the 32-bit version.
   Download `tdm64-gcc-5.1.0-2.exe` for the 64-bit version.
   Install to `C:\work\TDM-GCC-32` and `C:\work\TDM-GCC-64`.
3. Start → Search → Environment Variables → Add to `PATH`:
   `C:\work\TDM-GCC-32\bin;C:\work\MinGW\bin;C:\work\MinGW\msys\1.0\bin`
4. FLTK: `http://www.fltk.org/software.php`
   ZLIB: `https://sourceforge.net/projects/libpng/files/zlib/`
   `1.2.11/zlib-1.2.11.tar.gz`
   PNGLIB: `https://sourceforge.net/projects/libpng/files/libpng12/`
   `older-releases/1.2.7/libpng-1.2.7-config.tar.gz`

FLTK 1.3.4-2 – start an MSYS shell from `C:\work\MinGW\msys\1.0\msys.bat`:

```
# cd /c/work/
# tar -xvf fltk-1.3.4-2-source.tar.gz
# cd fltk-1.3.4-2/
# ./configure
# make
# cd /c/work/
# mkdir -p fltk/include/ fltk/lib/
# cp -r fltk-1.3.4-2/FL/ fltk/include/
# cp -r fltk-1.3.4-2/lib/*.a fltk/lib/
```

ZLIB 1.2.11 – start a Windows command prompt:

```
> cd c:\work\zlib-1.2.11
> copy win32\Makefile.gcc .\Makefile.gcc
> mingw32-make -fMakefile.gcc
> copy zlib.h ..\fltk\include\
> copy zconf.h ..\fltk\include\
> copy libz.a ..\fltk\lib\
```

PNGLIB 1.2.7 – start an MSYS shell from `C:\work\MinGW\msys\1.0\msys.bat`:

```
# cd /c/work/
# tar -xvf libpng-1.2.7-config.tar.gz
# cd libpng-1.2.7-config/
# ./configure CPPFLAGS=-I../fltk/include/ LDFLAGS=-L../fltk/lib/
# make
# cp png.h pngconf.h ../fltk/include/
# cp .libs/libpng.a ../fltk/lib/
```

Repeat the same steps for 64-bit version by replacing the occurrences of 32 with 64 and replace the path `fltk/lib/` with `fltk/lib64/` everywhere.

Compile for 32-bit – start a Windows command prompt:

```
> set PATH=C:\work\TDM-GCC-32\bin;%PATH%
> cd C:\work
> mingw32-c++ mfiremodel.c mgldraw.cxx minterface.cxx
    -o MkBGFire.exe -Ifltk/include -Lfltk/lib -lfltk -lfltk_gl
    -lopengl32 -mwindows -lcomctl32 -lole32 -luuid -lpng -lz -s -O1
> .\MkBGFire.exe
```

Compile for 64-bit – start a Windows command prompt:

```
> set PATH=C:\work\TDM-GCC-64\bin;%PATH%
> cd C:\work
> x86_64-w64-mingw32-c++ mfiremodel.c mgldraw.cxx minterface.cxx
    -o MkBGFire64.exe -Ifltk/include -Lfltk/lib64 -lfltk -lfltk_gl
    -lopengl32 -mwindows -lcomctl32 -lole32 -luuid -lpng -lz -s -O1
> .\MkBGFire64.exe
```

Sources:
`https://www.allegro.cc/forums/thread/608407`
`https://sourceforge.net/p/gnuwin32/discussion/74807/thread/085498b9/`
`http://www.imagemagick.org/discourse-server/viewtopic.php?t=14253`

Linking with recent version of PNGLIB
PNGLIB 1.6.36 – start an MSYS shell from `C:\work\MinGW\msys\1.0\msys.bat`:

```
# cd /c/work/
# tar -xvf libpng-1.6.36.tar.gz
# cd libpng-1.6.36/
# ./configure CPPFLAGS=-I../fltk/include/ LDFLAGS=-L../fltk/lib/ \
    --enable-static --disable-shared
# make
# cp png.h pngconf.h pnglibconf.h ../fltk/include/
# cp .libs/libpng16.a ../fltk/lib/
```

Instructions apply to both Linux and Windows, use linker flag `-lpng16`.

4 Technical information for v0.9.2 and earlier

The task is to create an executable file of `mygl.cxx` by using FLTK and OpenGL.

4.1 Linux

Operating system is Linux Mint 19 64-bit and compatible.

4.1.1 Dynamic linking

```
# sudo add-apt-repository universe
# sudo apt install g++ gcc libc6-dev libfltk1.3 libfltk-gl1.3 \
    libfltk1.3-dev libgl1-mesa-dev
# c++ mygl.cxx -o mygl -lfltk -lfltk_gl -lGL
# ./mygl
```

4.1.2 Static linking

Download FLTK from `http://www.fltk.org/software.php`

```
# sudo apt install g++ gcc libc6-dev \
    libgl1-mesa-dev libglu1-mesa-dev freeglut3-dev \
    libxrender-dev libxcursor-dev libxft-dev libfontconfig1-dev
# cd ~/work/
# tar -xvf fltk-1.3.4-2-source.tar.gz
# cd fltk-1.3.4-2/
# ./configure
# make
# cd ~/work/
# mkdir -p fltk/include/ fltk/lib/
# cp -r fltk-1.3.4-2/FL/ fltk/include/
# cp -r fltk-1.3.4-2/lib/*.a fltk/lib/
# c++ mygl.cxx -o mygl -Ifltk/include -Lfltk/lib -lfltk -lfltk_gl -lGL \
    -ldl -lX11 -lXext -lXfixes -lXrender -lXcursor -lXft -lfontconfig
# ./mygl
```

Packages `libxft-dev` and `libfontconfig1-dev` are required for font management. Find out the respective flags for static compilation with `fltk-config --ldstaticflags`.

4.2 Windows

Operating system is Windows 7 64-bit and any later version.

4.2.1 Dynamic linking

GCC/MinGW/TDM compilers generate `*.a` static libraries. Visual Studio C++ compiler generates `*.lib` static libraries (see `fltk.lib.vcxproj`) and `*.dll` dynamic libraries (see `fltkdll.vcxproj`), project files are in `fltk-1.3.4-2\ide\VisualC2010`.

`http://www.fltk.org/doc-1.3/opengl.html#opengl3`
`http://www.fltk.org/articles.php?L371`
`http://www.fltk.org/articles.php?L372`

4.2.2 Static linking

1. Download TDM-GCC 5.1.0 and install to `C:\work\TDM-GCC-64`:

   `http://tdm-gcc.tdragon.net/download`

   Select `tdm64-gcc-5.1.0-2.exe` for the 64-bit version.

2. Download mingw-get-setup and install to `C:\work\MinGW`:

   `https://sourceforge.net/projects/mingw/files/Installer/mingw-get-setup.exe`

   Mark for installation `mingw-developer-toolkit` and `msys-base`.

   Then select Installation → Apply Changes.

3. Start → Search → Environment Variables → Add to `PATH`:

   `C:\work\TDM-GCC-64\bin;C:\work\MinGW\bin;C:\work\MinGW\msys\1.0\bin`

4. Download FLTK from `http://www.fltk.org/software.php`

Start an MSYS shell from `C:\work\MinGW\msys\1.0\msys.bat`:

```
# cd /c/work/
# tar -xvf fltk-1.3.4-2-source.tar.gz
# cd fltk-1.3.4-2/
# ./configure
# make
# cd /c/work/
# mkdir -p fltk/include/ fltk/lib/
# cp -r fltk-1.3.4-2/FL/ fltk/include/
# cp -r fltk-1.3.4-2/lib/*.a fltk/lib/
```

Start a Windows command prompt:

```
> cd C:\work
> x86_64-w64-mingw32-c++ mygl.cxx -o mygl.exe -Ifltk/include -Lfltk/lib
    -lfltk -lfltk_gl -lopengl32 -mwindows -lcomctl32 -lole32 -luuid
> .\mygl.exe
```

Flag `-mwindows` creates a no-console Windows executable, if console window is needed, replace that flag by `-lgdi32 -lcomdlg32`.

For all compilations, flags `-s -O1` can be used, for stripping the executable of debugging symbols and for optimizations `O1`, `O2`, `O3`, respectively. Flag `-save-temps` can be used to save the preprocessed source file, in `*.i` for C and `*.ii` for C++.

Sources:
https://groups.google.com/forum/#!topic/ppp-public/CuTWEfUcPiw
https://stackoverflow.com/questions/29329016/link-failure-undefined-references-to-graphical-functions
https://stackoverflow.com/questions/47058727/link-fltk-with-g
https://stackoverflow.com/questions/22314120/imp-link-errors-using-g-running-under-mingw
https://stackoverflow.com/questions/3933027/how-to-get-the-gl-library-headers
https://stackoverflow.com/questions/30154115/fltk-complains-about-gcc-on-windows
https://github.com/mmgen/mmgen/wiki/Install-MinGW-64-and-MSYS-on-Microsoft-Windows
http://tdm-gcc.tdragon.net/bugs
https://sourceforge.net/p/mingw-w64/mailman/message/34523082/