Annals of DAAAM for 2011 & Proceedings of the 22nd International DAAAM Symposium, Volume 22, No. 1, ISSN 1726-9679
ISBN 978-3-901509-83-4, Editor B. Katalinic, Published by DAAAM International, Vienna, Austria, EU, 2011
Make Harmony between Technology and Nature, and Your Mind will Fly Free as a Bird
Annals & Proceedings of DAAAM International 2011

THE CALCULATION INSOLATION OF HOUSES

POKORNY, P[avel] & MAHDALICEK, P[etr]

Abstract: *This paper describes the program, which was developed at the Faculty of Applied Informatics, Tomas Bata University in Zlin in the last year. It is designed to simulate the insolation of houses. The created application reflects the valid standard in the Czech Republic and based on entered data it can reliably assess, if these standards are met or not. The application was designed for the Linux platform using the C++ programming language and Ogre graphics engine.*

Key words: *visualization, simulation, application, programming, insolation*

1. INTRODUCTION

There is evidence that solar radiation has a positive effect on the human psyche. Therefore it cannot be regarded only as a source of light and heat, but also necessarily as a source of high spirits. For this reason, the standards setting principles for the designing of houses or its parts are the considerable part devoted to the issue of insolation. These standards define the rules, by which is evaluated whether the sun's rays insolate the rooms. They also determine the minimum time periods, after which the evaluated space must be insolated. These standards must take into account not only the construction of new apartments and houses, but also in the case of construction of buildings, which its shape can suppress the breakthrough of sunlight to existing buildings.

The people in the building design deal with evaluation of these data. This procedure includes several processing steps. It is necessary to create the 3D model of evaluated area, including neighboring buildings. It is also necessary to specify the location in which the object is evaluated. With these data, it is possible to calculate the position of the sun and then the period during which the sun's rays fall into the rooms of residential spaces.

There are several applications, which make this issue easier. Some of them solve this issue very commonly including problems such as illumination by artificial light sources, other applications can fully calculate insolation. But none of them available in the Czech Republic supports Linux operating system. In addition, the Czech standard dealing insolation (ČSN 73 4301) was updated in the year 2009 (***, 2009). For these reasons, we created our own application, which reflects this new update.

2. INSOLATION OF HOUSES

The assessment of the house insolation is taken into account only the living rooms (bedroom, living room, etc.). The Czech standard specifies the following rule: The house is enough insolated, if its area of insolated rooms is equal to at least one third of all its rooms. The room is sunny, if all of the following conditions are met: (***, 2009).

- The plane angle between the sun and the plane of the room window must be at least 25°.
- The window must be based on a pure transparent material.
- The smallest window size must be at least 900 mm.
- Height of the sun above the horizon must be at least 5°.
- If we neglect the clouds, on 1 March and 21 June the period of the insolation must be at least 90 minutes.
- The sunlights must fall on the critical point of window**.** The critical point is located inside the plane window at the height of 300 mm above the center of the bottom edge of the lighting hole, but at least 1200 mm above the level considered rooms.

It is also necessary to know the object orientation to the compass. When we determine the direction of the meridian, we can get the meridian convergence C(°). The size of meridian convergence can be determined in several ways:

- By subtracting from the base map.
- If the situation is treated by the calculation of a trigonometric network in the single coordinate system, the meridian convergence can be solved by the equation

$$C = \frac{24°50 - \varepsilon}{1.34} \qquad (1)$$

ε is the longitude of location.

- By asking the Centre of Geodesy and Cartography.

The observed or calculated angle is applied clockwise from the vertical coordinate lines of the cartographic terrain and with this, the northern direction is determined.

We also need to know the exact position of the sun. To get this, we must to determine the real solar time (it is created by splitting the time interval between two successive upper culminations of the sun for 24 hours), declination (one of the two coordinates of the equatorial coordinate system, which gives the angular sun distance from the celestial equator and depends on the date of calculation), sun altitude and azimuth (angle applied to the meridian clockwise - the horizontal angular distance from a reference direction). All balance equations are described in (***, 2009).

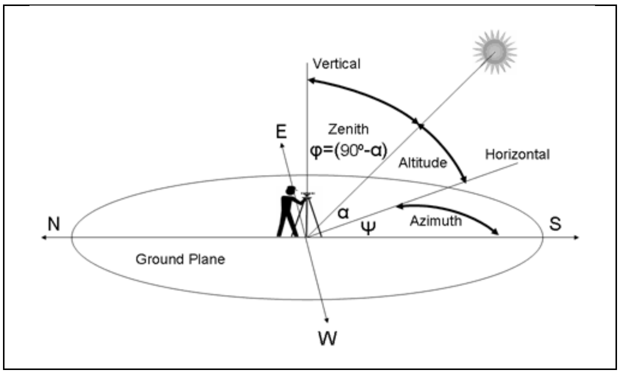

Fig. 1. Azimuth and Sun altitude for Northern Latitudes

3. THE APPLICATION – PROGRAMMER VIEW

3.1 Requirements

The main requirements for the created application were set as follows:

- It allows to make complex 3D polygon models of houses.

- Selecting and marking of flats, rooms and windows inside the houses.
- It calculates the insolation of flats and evaluates it according to defined standards.
- Creating the final report.

The most difficult task is the first one – the 3D modeling. This can be addressed in two ways. The first way is to create our application like the plugin of the one complex 3D modeling software (AutoCAD, BricsCad, etc.). The second way is to create own modeling environment. The first solution is clearly less difficult and if the user knows the used host application, the work with our software would be very simple. However, this solution has also some disadvantages, especially the dependence on the third-party applications. Therefore, we tried to create custom modeling environments and there will be supported the only basic modeling operations (Lander, 2003).

3.2 Used technologies

Our application was written in C++ language. C++ is the most used object-oriented programming language, featuring all advantages of object-oriented programming - encapsulation, inheritance, polymorphism, overloading and templates (Soustrup, 2000).

We also used QT library. This library is primarily used to development of graphics applications, but it offers more. Besides the core and GUI it is the support for OpenGL library, scripting, work with networks, databases, XML and multimedia files (***, 2008).

Further, our application shows all graphics 3D outputs via OGRE engine. Since 2001, OGRE has grown to become one of the most popular open-source graphics rendering engines, and has been used in a large number of production projects, in such diverse areas as games, simulators, educational software, interactive art, scientific visualization, and others (***, 2001).

The last used technologies in our application are Boost C++ libraries. Boost includes around 90 helpful libraries of general algorithms that facilitate programming with C++ programming language. Boost libraries are intended to be widely useful and usable across a broad spectrum of applications (***, 2005).

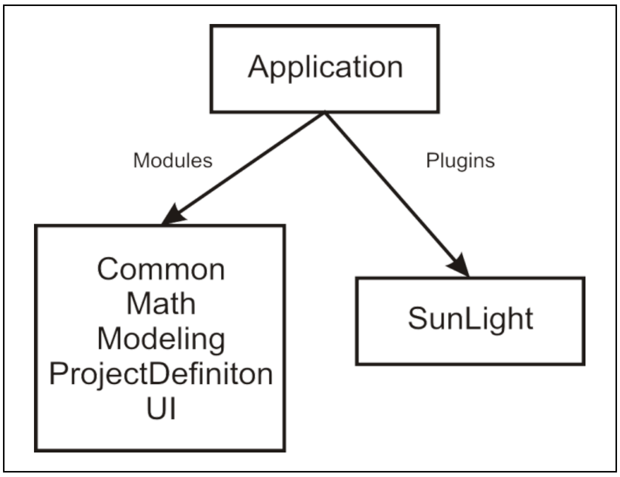

Fig. 2. The sub-modules of our application

3.3 Application modules

We designed the concept of the whole application as a sum of several separate modules. They are shown on the Figure 2.

The Application module has defined the only two objects. The first object is the Plugin class, which defines the interface used to load any plugin of supported applications. The second object is the application class and this class controls the entire application. Its main task is to read the settings and available plugins and then create the main application window. It also implements methods common to all projects, such as projects saving and loading.

The Common module contains the most general definition of classes applicable in many different situations. The most important types used in our project include String (work with strings), Event (class describing a common event), List (extended std::vector to work with arrays), MapEx (extended std::map) (***, 2011), Properties (it allows us to save named values of any type) and Location (represents a named location).

Inside the Math module we can find classes representing mathematical objects. For standard operations such as transformations we use Ogre engine classes, in addition here we added new classes which describe the basic geometric objects.

The Modeling module includes all objects inside the whole 3D scene. The basic element of the scene is node. The geometric node is derived from him and it contains the geometry shape (standard 3D polygonal representation) and its material.

The ProjectDefinition module contains the definition of classes which represent the project items. It also defines the project itself. Moreover it includes classes Layer, Item, StaticItem, Project, ProjectData and ProjectFactory.

The UI module implements all used widgets inside the application. The most of these objects were defined by the visual tools in QTCreator (***, 2008).

The last module - Sunlight - implements an application plugin. This is an extension that solves the insolation of houses. It implements own project management and defines the classes derived from StaticItem class. These classes are Apartment (represents the flat), Room (represents the class room) and Window (represents the window). All necessary algorithms to calculation of the insolation are also implemented here.

4. CONCLUSION

There are existing the standards, which deal with the needs of the houses insolation. These standards define the minimum values of insolation and specify the methods for calculation of the relevant parameters. This allows determine whether the house meets the standard.

In order to facilitate this evaluation, we created own application that solves the whole issue. This application designed for Linux platform allows to make 3D models of spaces through simple graphical tools and display position of the sun due to the solved house. Our application is tested and fully functional. It is available on our site.

5. REFERENCES

Lander, J. (2003). *Graphics programming methods,* Charles River Media inc., ISBN 1-58450-299-1, Massachusetts, United States of America

Soustrup, B. (2000). *The C++ Programming Language: Special Edition,* Addison-Wesley Professional, ISBN 978-0201700732, Massachusetts, United States of America

*** (2001) http://ogre3d.org – OGRE – Open Source 3D Graphics Engine, *Accessed on: 2011-08-02*

*** (2005) http://boost.org – Boost – C++ libraries, *Accessed on: 2011-08-02*

*** (2008) http://qt.nokia.com – Qt – Cross-platform application and UI framework, *Accessed on: 2011-08-02*

*** (2009) http://csnonline.unmz.cz - ČSN 73 0581 (Czech Technical Norm), *Accessed on: 2011-08-02*

*** (2011) http://sgi.com/tech/stl – Standard Template Library Programmer’s Guide, *Accessed on: 2011-08-02*