

# Praktikumsbericht

Interactive Architectural Design Toolkit

---

*Author: Felix Kreuzer*

*Supervisor: Przemyslaw Musialski*

---

## Introduction

Traditional CAD tools for architects offer a vast variety of drawing utilities to design buildings. It is up to the architect to maintain structural constraints such as beams and room-accessibility.

The main purpose of this internship was to design and implement an architectural CAD tool which automatically optimizes the building layout, based on constraints derived from architectural best-practice concepts. The toolkit offers a user interface, where drawing and editing can be performed and a backend interface, where optimization algorithms i.e. constraint solvers can be implemented.

## API Documentation

The API documentation can be generated using *Sandcastle Help File Builder* which can be found at <https://github.com/EWSoftware/SHFB>. Once Sandcastle is installed, open up the *CodeDocumentation.shfbproj* file located in the *Doku* directory, choose what kind of output you prefer (e.g. HTML) and click the build button.



Figure 1: The red circle marks Sandcastle's build button.

## Running the Toolkit

The Toolkit requires a MATLAB installation to be present in order to run. Once MATLAB is installed, use Visual Studio 2012 to compile and run the code. If you want to use a different version of Visual Studio keep in mind, that the toolkit is configured for dot Net Framework 4.0.

## Basic Usage

The User interface consists of a Windows-style File Menu, a Canvas for drawing, and a Toolbox.

The File Menu offers load and save functionality as well as a help section. Use the toolbox to select which type of object will be drawn and to navigate through the building's floors.

Initially the canvas contains the floorplan's boundary only. New shapes can be drawn by holding the SHIFT key while dragging the mouse. Newly drawn shapes are automatically selected. Selected items can be resized or dragged.

For detailed information on how to draw on the canvas open the help section in the file menu.

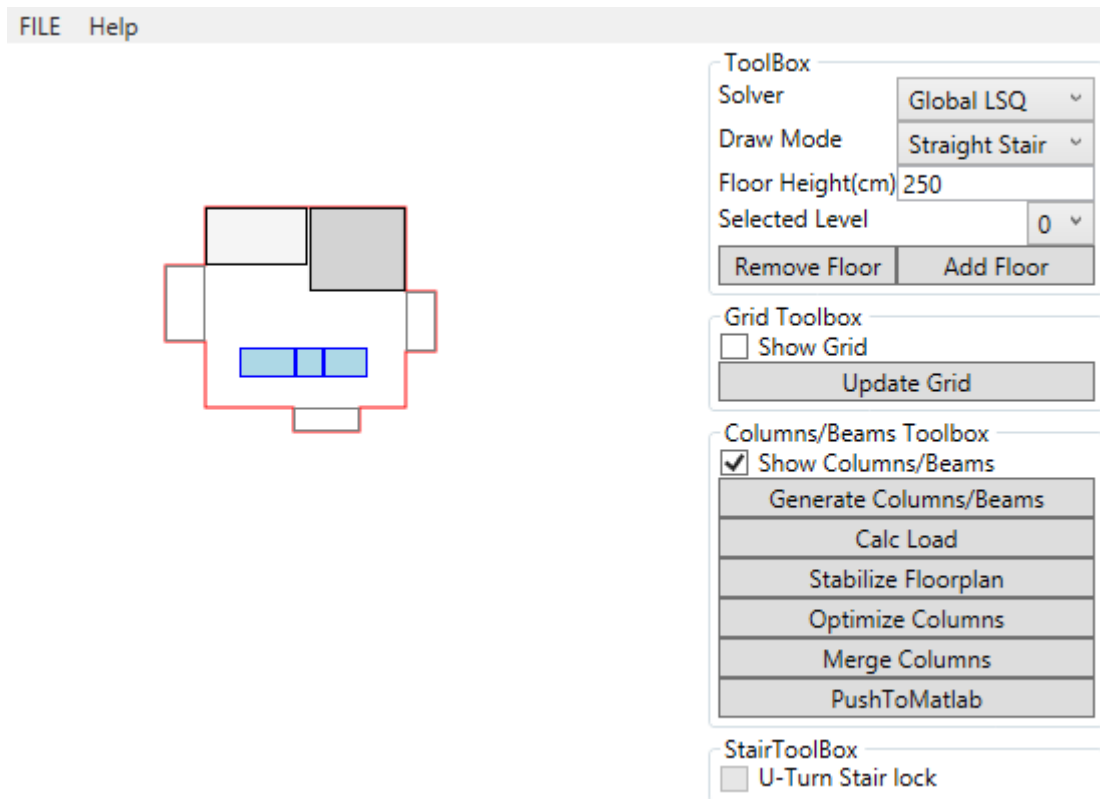


Figure 2: User Interface with Canvas and Toolboxes.

## Implementation

The project was implemented in c# using the Microsoft WPF SDK. The user interface enables the user to draw elementary structures i.e. rooms, stairs and floorplan projections using drag and drop. Whenever the buildings layout is changed, a solver is executed and the layout automatically adapts to meet the criteria defined by the constraints. The while the tool offers a simple sequential solver to meet the constraints an interface is provided allowing support for more complex solvers.

The implementations follows the MVVM concepts. The User interface (View) has its own user-control wrappers for each class in the model (see figure 3). Using Microsoft's ItemsControl class we can traverse the floorplan's nested structures dynamically in the view.

Solvers can be integrated by inheriting the IRuleSolver Interface.

Model

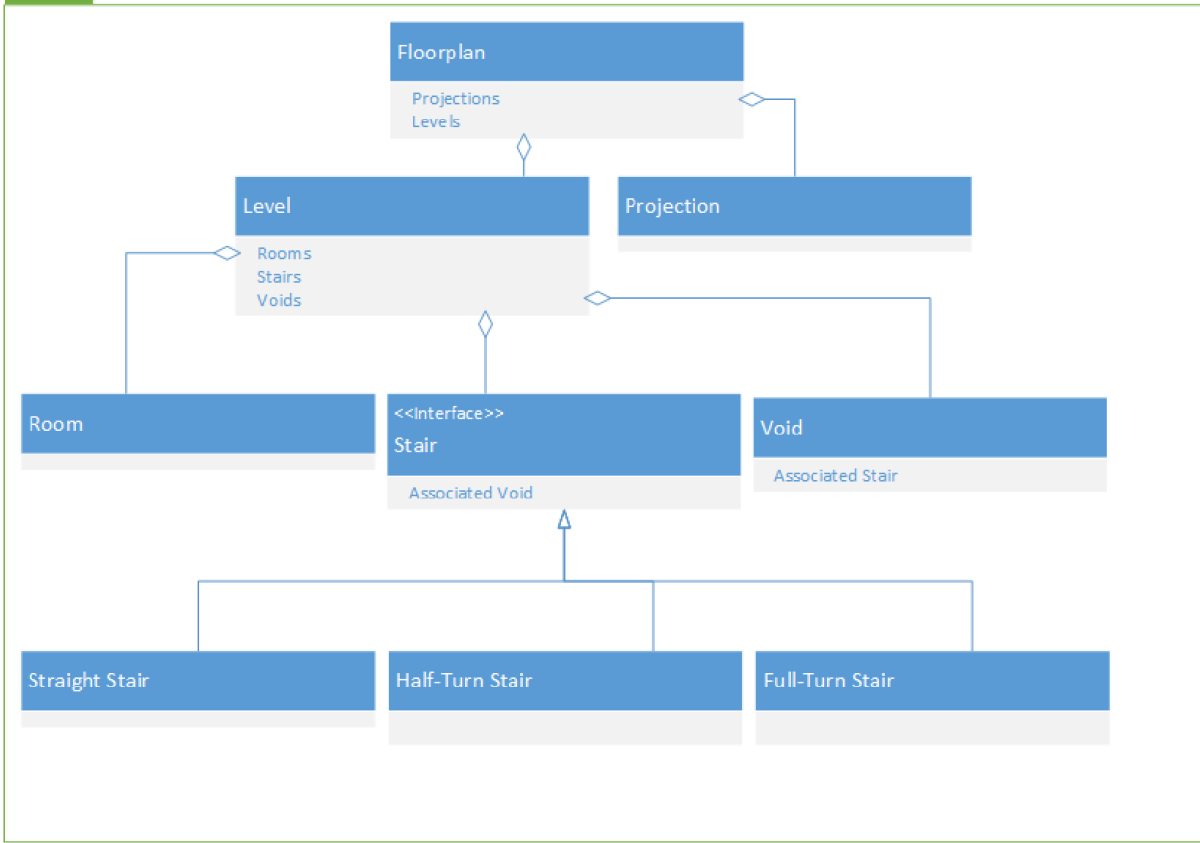


Figure 3: Simplified UML Diagram of the Model Classes. The View has User Controls for each of these Classes.

For more details concerning the actual implementation consider reading the accompanying API documentation.