



LUND UNIVERSITY

Designing a new User Interface for "Objectiveframe"

Lindemann, Jonas

2003

Document Version:

Publisher's PDF, also known as Version of record

[Link to publication](#)

Citation for published version (APA):

Lindemann, J. (2003). *Designing a new User Interface for "Objectiveframe"*. (TVSM-7000; No. TVSM-7137). Division of Structural Mechanics, LTH.

Total number of authors:

1

General rights

Unless other specific re-use rights are stated the following general rights apply:

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Read more about Creative commons licenses: <https://creativecommons.org/licenses/>

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00



LUND
UNIVERSITY



DESIGNING A NEW USER INTERFACE FOR "OBJECTIVEFRAME"

JONAS LINDEMANN

Structural
Mechanics

Structural Mechanics

ISRN LUTVDG/TVSM--03/7137--SE (1-43)

ISSN 0281-6679

DESIGNING A NEW USER INTERFACE
FOR "OBJECTIVEFRAME"

JONAS LINDEMANN

Copyright © 2003 by Structural Mechanics, LTH, Sweden.
Printed by KFS I Lund AB, Lund, Sweden, May, 2003.

For information, address:
Division of Structural Mechanics, LTH, Lund University, Box 118, SE-221 00 Lund, Sweden.
Homepage: <http://www.byggmek.lth.se>

Contents

1	Introduction	1
1.1	Vision and goals	1
2	Initial user study	3
2.1	Goals	3
2.2	Knowledge and skill level	4
2.3	Work environment	4
2.4	Objects and tasks	4
3	Analysis of ObjectiveFrame	7
3.1	General user interface description	7
3.2	Cursor placement	7
3.3	Object selection	8
3.4	Moving nodes	9
3.5	Applying loads and boundary conditions	10
4	User interface design (New prototype)	11
4.1	Conceptual design	11
4.2	Interaction design	12
4.3	Graphical design	17
5	Usability study	19
5.1	User test	19

5.2	Heuristic evaluation	25
6	Conclusions and improvements	29
A	Test instructions and tasks	31

List of Figures

3.1	Current ObjectiveFrame user interface	8
3.2	Cursor mapping problems	9
3.3	Object selection problems	9
3.4	Load and boundary condition windows	10
4.1	Ball-and-stick models	12
4.2	A Meccano kit. Image from [1]	13
4.3	Prototype main window	13
4.4	Updated 3d cursor	14
4.5	Node creation state diagram	14
4.6	Z-mapping method	15
4.7	Move cursor	16
4.8	Moving/copying objects	16
4.9	Applying loads	17
4.10	New object selection method	18
5.1	User times for task 1	21
5.2	User times for task 2	21
5.3	User times for task 3	22
5.4	User times for task 4	23
5.5	User times for task 5	24
5.6	User times for task 6	24

5.7 User times for task 7 25

List of Tables

2.1	Typical user goals	3
4.1	ObjectiveFrame building blocks	12

Chapter 1

Introduction

During sometime the Division of Structural Mechanics has been teaching mechanics to engineering, architectural and industrial design students. Special software tools has been developed to aid in this process, ForcePAD and ObjectiveFrame. ForcePAD is a 2D finite element application using a paint program metaphor. This metaphor has been successfully used in many courses with good comments in the student evaluations. The second application ObjectiveFrame is a 3D frame analysis software. With ObjectiveFrame the students can interact in real-time with the structures they create. This functionality of the software has been successfully used with prebuilt models. The problems start when the users have to create the structures themselves. The general structure of ObjectiveFrame is that of a classical finite element application. In a classical finite element application you first create element properties (Section data, material properties etc.). These properties can then be assigned to the elements in the model. To make the model flexible an element property can be assigned to many elements. The same approach is also used in conjunction with loads and boundary conditions. The advantage of the above described method is flexibility. The disadvantage is that novice users and users not familiar with finite element software have difficulties using the applications. We have seen this in the courses in which we have used ObjectiveFrame. Other issues that are problematic in ObjectiveFrame is how to interact and create the 3d frame model in an easy to use way.

1.1 Vision and goals

The vision or goal of this project is look at how the interface of ObjectiveFrame can be designed in a way that it is more accessible to a wider range of users, especially students learning mechanics at different levels. The project will not look at all aspects of the ObjectiveFrame user interface, but focus on some problem areas as described above.

Chapter 2

Initial user study

The main users of ObjectiveFrame can be described as follows:

- **Architectural students** learning mechanics as a part of the courses in high structures during the final part of their education.
- **Industrial design students** learning basic mechanics in the first terms of their education.
- **Civil engineering students** learning basic mechanics in the first and second semester
- **Teachers** in mechanics

2.1 Goals

The typical goals of the users can be described as in the following table:

User group	Goals
Architectural students	Create and study the behaviour of buildinglike structures.
Industrial design students	Create and study the behaviour of structures.
Civil engineering students	Create and study the mechanical behaviour of beams and beam structures. Create beam topology for input to other finite element software.
Teachers	Create illustrative model for use in teaching mechanics.

Table 2.1: Typical user goals

2.2 Knowledge and skill level

Most of the users are quite computer literate and have little problem handling complex user interfaces. The students in architecture and industrial design often use applications such as Adobe Photoshop, Illustrator and 3DStudio, which have a very complex user interface. Users in civil engineering often have had experience with CAD software such as AutoCAD.

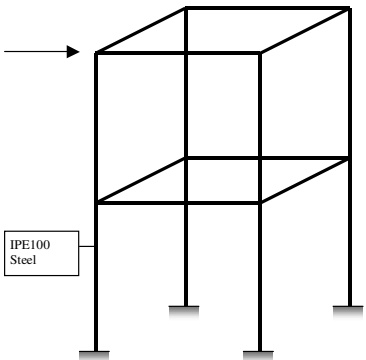
2.3 Work environment

The student work environment can be divided into two groups. Civil engineering students often work in a lecture and exercise based form, attending lectures given by staff at the departments and then taking part in scheduled 2 hour exercises. The students in architecture and industrial design often work on larger projects attending lectures and then applying the knowledge in the projects. Scheduled exercises exists in some courses.

The application will be used to complete certain exercises. The application is installed in the computer labs, but is also available for home use by downloading it from the web.

2.4 Objects and tasks

The assignments given in the courses are typical to study some kind of structure and then discuss some phenomena. A typical task can be describes as follows:



The diagram shows a 3D wireframe structure of a frame. It consists of a rectangular frame with a horizontal member at the top and a vertical member on the left. A horizontal arrow points to the top-left corner of the frame, indicating the application of a force. A label 'IPE100 Steel' is placed next to the vertical member on the left. The structure is supported by three vertical columns at the bottom, each with a fixed support symbol.

1. Create the given structure. Add additional members to the structure so that the deflection of the top structure is maximum 10 mm.

The objects involved in this exercise are:

- Structural nodes (connection points)
- Beam elements
- Forces
- Boundary conditions or constraints
- Simulation
- Results
- View

Typical tasks in the exercise are:

- Modifying the viewpoint
- Creating, modifying and deleting nodes
- Connecting nodes with beam elements
- Removing beam elements
- Modifying beam element properties
- Adding, modifying and deleting loads
- Adding, modifying and deleting boundary conditions
- Running simulation
- Viewing and interpreting results
- Modifying model after interpreting results

Chapter 3

Analysis of ObjectiveFrame

To understand the behaviour of structures subjected to loads the 3d beam analysis program ObjectiveFrame has been developed. One of the features of this program is that it can visualise the response of a structure to a user-defined load in real-time. This makes it suitable for use as an educational tool in design science, architecture and structural mechanics.

When ObjectiveFrame has been used in our courses at the architectural school and design school, it was apparent to us that it had usability problem. This section will try to describe the current user interface and the problems related with its design.

3.1 General user interface description

The main user interface in ObjectiveFrame is the workspace. The workspace is the "workbench" on which the beam models are assembled and consists of an axis and a transparent grid. The user can change the workspace size to fit the model constructed.

The main user interface is centered around the 3d view showing the "virtual" workbench. Around the 3d view the main toolbars of ObjectiveFrame are placed: the edit toolbar for manipulating existing objects in the view, the model toolbar for creating new objects in the scene and the view toolbar for manipulating the current 3d view. The current user interface is shown in figure 3.1

Object placement, movement and creation are done using a special 3d cursor. The position of the 3d cursor is determined by mapping the mouse coordinates on to the 3d grid.

3.2 Cursor placement

A central concept in ObjectiveFrame is the node. A node is a connection point to which beams, loads and boundary conditions are connected. The node has to be placed in 3d

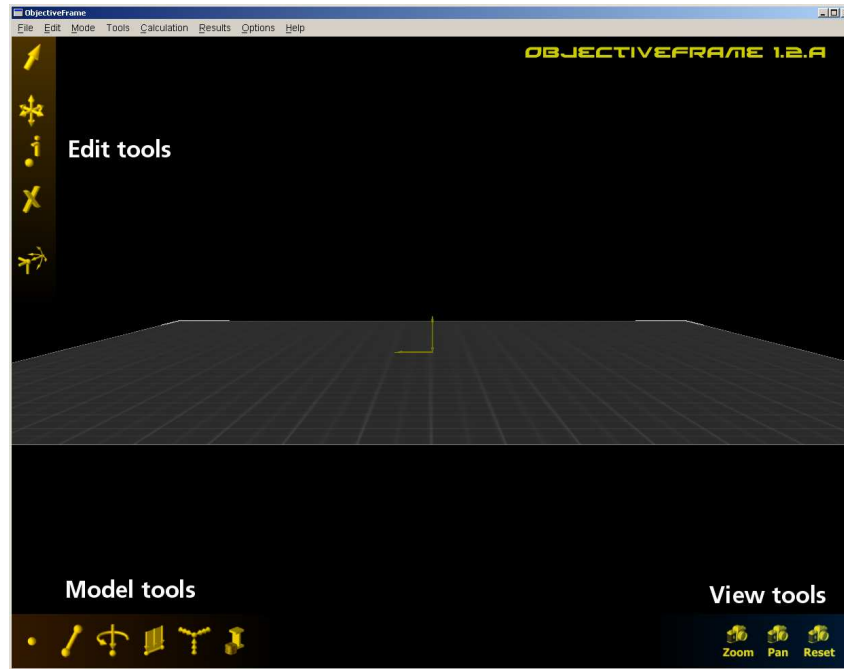


Figure 3.1: Current ObjectiveFrame user interface

in some way. To implement node placement ObjectiveFrame uses a special 3d cursor, which can be moved using the mouse. Using a mouse to place 3d objects has certain drawbacks, because it only supports 2 degrees of freedom x and y. The current version of ObjectiveFrame projects a ray from the mouse position on the screen to the plane of the "virtual workbench". This enables cursor movement in the plane. To support movement out of the plane the cursor must be locked in the xy-plane. Cursor locking is done by holding down the [Ctrl] key.

The problem with node creation and cursor placements was that the information of locking was not visible in the user interface. This made it harder for new users to begin using the application. The process of creating nodes was also not clear from looking at the user interface. The cursor did not provide any clues of what to do when the node creation mode in ObjectiveFrame was selected, which lead to many questions from the students.

Another problem with the cursor placement method was the mapping of mouse movement when placing a node out of plane (z-direction). It is not always possible to figure out the mapping of the mouse movement to the movement of the 3d cursor. Figure 3.2 illustrate these problems.

3.3 Object selection

Modifying and removing objects in ObjectiveFrame requires objects to be selected. Selection is handled using the select tool in the edit toolbar. When the mouse is over a selectable object the object is highlighted in white. Clicking on the object changes the object color to yellow. The highlight effect is providing good feedback for the users, indi-

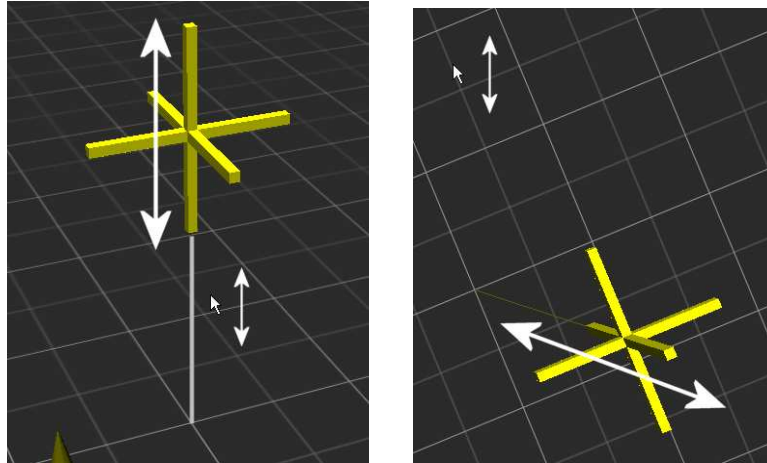


Figure 3.2: Cursor mapping problems

cating what objects are selectable. The method for indicating that an object is selected is causing a lot of problems. The greatest problem is the conflict between the coloring scheme in ObjectiveFrame. In ObjectiveFrame it is possible to assign different beams with colors indicating specific element properties. By selecting yellow for a given element property will greatly confuse object selection, see figure 3.3

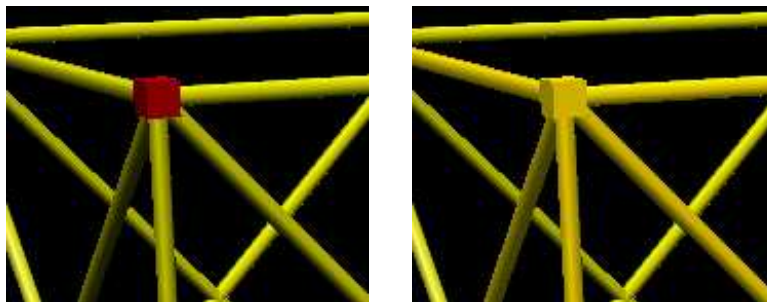


Figure 3.3: Object selection problems

3.4 Moving nodes

The method for moving nodes is also causing problems. The current method of moving nodes is as follows:

1. Select the the points to be moved using the select tool.
2. Select the move tool.
3. Place the 3d cursor at the initial movement point.
4. Drag the 3d cursor for moving the points. Use [Ctrl] for moving the points in the z-direction.

The method described above are giving users a lot of problems. The current method does not provide any clues of how it should be used. Many users click on the movement tool and try to drag the points directly, which will fail or will not provide the desired outcome.

3.5 Applying loads and boundary conditions

The main usability problem with ObjectiveFrame is that it was initially not designed to be used in teaching design and architectural students, but as an experimental user interface for a 3d beam analysis code. The target users were engineers or engineering students. These user groups often require the applications to be very general and flexible. This often leads to more complex user interfaces.

Loads and boundary conditions in ObjectiveFrame are handled using special windows (see figure 3.4), where the users can create load cases and apply these to selected nodes or elements.

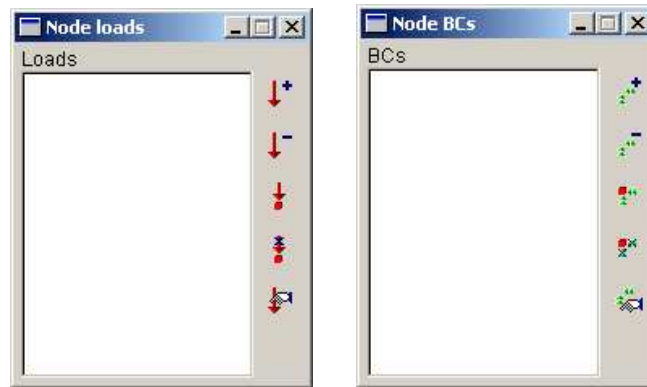


Figure 3.4: Load and boundary condition windows

Architect and design students are often unfamiliar with the concepts of load cases. The concept of applying a load defined in the window to many nodes at the same time is also confusing. A more direct approach would be preferable.

Chapter 4

User interface design (New prototype)

To remedy the deficiencies in the current user interface a new prototype interface for ObjectiveFrame will be constructed. The prototype will focus on the following areas as described in section 3:

- Node creation and cursor placement
- Object selection
- Moving and copying nodes
- Applying loads and boundary conditions

4.1 Conceptual design

The conceptual design in ObjectiveFrame is centered around the "virtual workshop" concept. In the concept the user has a "workbench" on which the model is assembled. As an aid in assembling a model the user also has a number of building blocks. ObjectiveFrame uses a ball-and-stick model for creating three-dimensional beam models. This model consists of two main constructional elements: the ball and the stick. The ball is used as a connection point to which the sticks are connected to form a structure. The ball and stick model can be found in chemistry for assembling molecular structures and in engineering for assembling space frames, see figure 4.1

The main building blocks in ObjectiveFrame are described in the following table:

These building blocks presents the user with a "construction kit" for beam and truss structures. It can be compared to having a "Meccano kit" with simple elements that can be assembled into more complex structures.

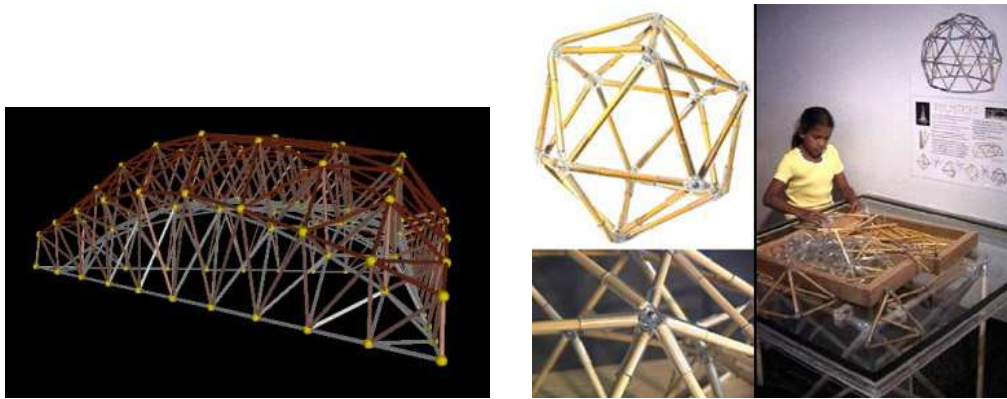


Figure 4.1: Ball-and-stick models




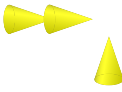
Representation	Name	Description
	Node	The node is the "ball" in the conceptual model of ObjectiveFrame. Beams, loads and boundary conditions can be attached to the nodes.
	Beam/bar	The beam is the "stick" in the conceptual model. It is also the main structural element in the finite element model.
	Load/Force	The load represents an external force applied to the structure and is represented by an arrow pointing in the direction of the force.
	BC	A boundary condition (BC) representation a constraint to the structure. For example a node fixed support.

Table 4.1: ObjectiveFrame building blocks

4.2 Interaction design

The ObjectiveFrame prototype was created using a combination of tools and libraries. The 2d user interface was created using the FLTK [2] library and the user interface designer FLUID, that comes with this library. The 3d user interface was implemented using the Ivf++ [3] visualisation library. These tools enable quick and easy creation of a functional prototype. It is also easy to change functionality of the prototype.

The main window of the prototype is divided into three parts: 3d view, toolbar and status view, as shown in figure 4.3. In the toolbar function for setting the current state and other functions are placed. The current state is indicated using a checkbox in the button. In the status view the users are given short information on the selected state or function. There is also a coordinate display showing the current coordinate of the 3d cursor. Hints are also

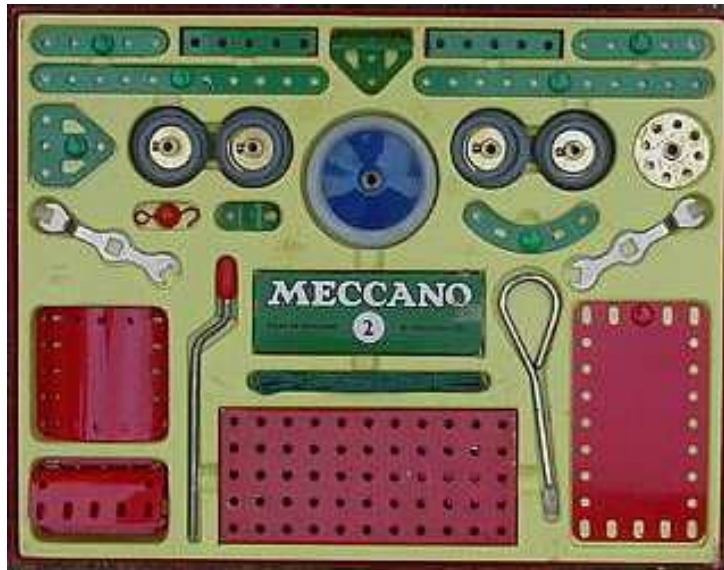


Figure 4.2: A Meccano kit. Image from [1]

used to display additional information about the state and functions. The appearance of the toolbar and status views will probably be changed in the final version.

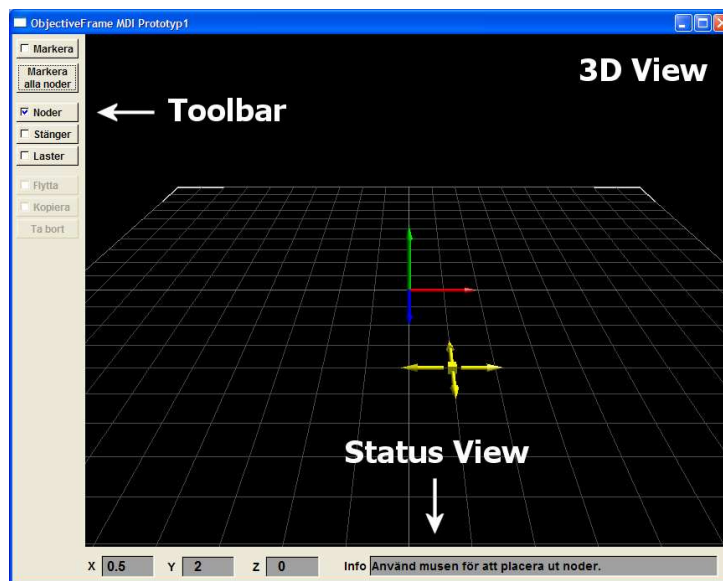


Figure 4.3: Prototype main window

The user interface of the ObjectiveFrame prototype is designed as a direct manipulation interface. That is:

- Continuous visual representation of objects and current operations.
- Physical actions instead of complex commands.
- Fast, incremental and reversible actions.
- Easy for beginners to start using the system.

In the ObjectiveFrame prototype direct manipulation is used in most operations, for example when moving a node will show a node representation being moved on the screen. Rotating the view is updated directly when the user moves the mouse.

Node creation and cursor placement

To make the node creation more intuitive the 3d cursor has been enhanced. The cursor now shows the object to be placed in the middle and has arrows showing the direction of allowable movement, see figure 4.4

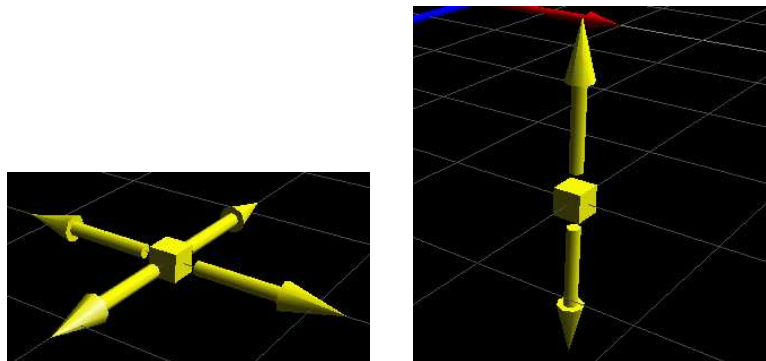


Figure 4.4: Updated 3d cursor

The first step when creating a node is placing the node in the XY-plane. The cursor shows for arrows representing the allowable movement direction in the XY-plane. When the user clicks at a position in the XY-plane the 3d cursor will lock at the selected position and will show two arrows in the Z-direction indicating that the allowable movement now is in the Z-direction. Clicking a second time will create a node at the specified position. The cursor is then return to the initial state, locked in XY-plane to allow the user to create additional nodes. A state diagram of the process is shown in figure 4.5

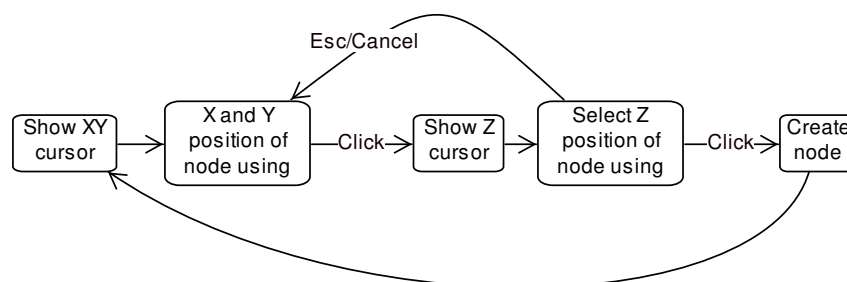


Figure 4.5: Node creation state diagram

In the current version of ObjectiveFrame cursor movement in the Z-direction was done by taking mouse y movement and scaling it against the workspace size. This method produced mapping problems when used in certain view angles as described earlier. To

solve this, the intersection point from the screen (x,y) position to a special plane containing the XY-position selected in the first click is calculated. This method will provide a more correct mapping of the mouse movements against the 3d cursor movement. The method is illustrated in figure 4.6.

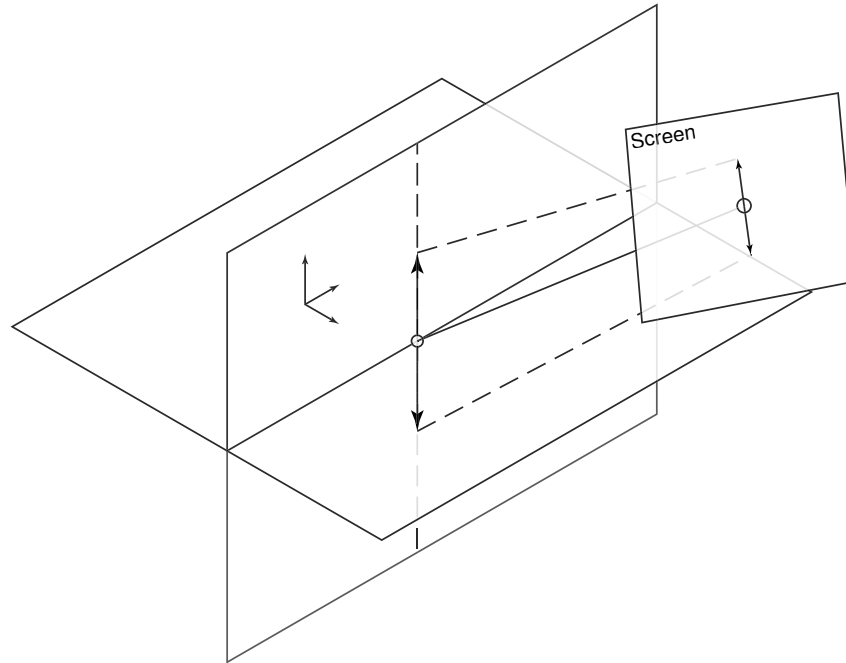


Figure 4.6: Z-mapping method

Object selection

Most functions such as moving, copying and deletion in ObjectiveFrame operate on an object selection. Object selection in the prototype application is done by clicking on a selectable object. Selectable objects are highlighted when the mouse is moved over it in the workspace. Clicking on additional objects will add these to the selection. If an object is already selected the object will be removed from the selection.

Moving and copying objects

Moving and copying operates on selected nodes. To move or copy a selection of objects, the user clicks on the object. The same method as in node creation is used to move a selection. To give the user a better understanding of what to do, a special cursor shape is shown over a node when the mouse is over it, see figure 4.7. The selection highlight is moved along with the 3d cursor when the user has clicked on the first node, see figure 4.8

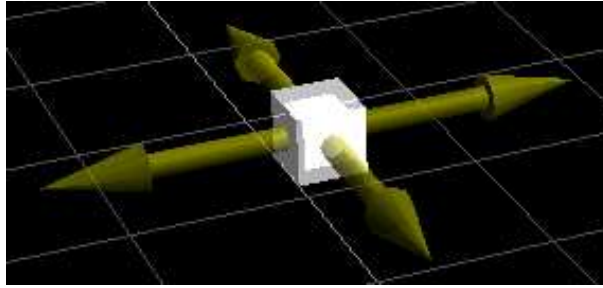


Figure 4.7: Move cursor

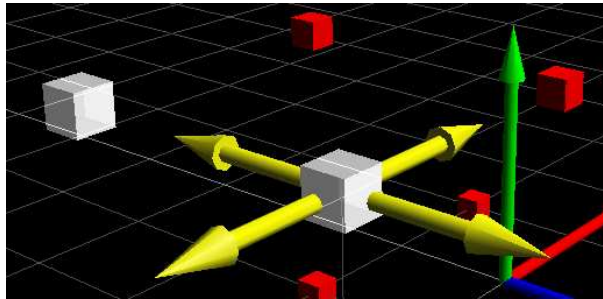


Figure 4.8: Moving/copying objects

Creating loads

In the current version of ObjectiveFrame loads are created in the load windows. The direction of the load is specified in x, y and z directions. Setting a negative value can also affect the direction of the load. To make the load creation more direct a special method for visually applying a load has been developed.

The difficulty of creating a node in 3d is to be sure in what direction the load is directed. The perspective transform distorts angles and distances making it hard to see in what direction a vector is directed. The prototype implements a special "widget" aiding the user in directing the load visually in 3d.

The load creation process starts by moving the mouse over a node. When the mouse is over a node a special load cursor is shown indicating that a load can be applied to the node. Clicking on the node will show the "widget" for directing the load. The "widget" consists of one vertical circle and a horizontal circle. On the vertical circle the load is attached. The load can "glide" on the vertical circle providing the first angle of the load. The vertical circle is attached to the horizontal circle which is rotated in the second click, providing the second angle for load placement. The complete process of applying a load is shown in figure 4.9

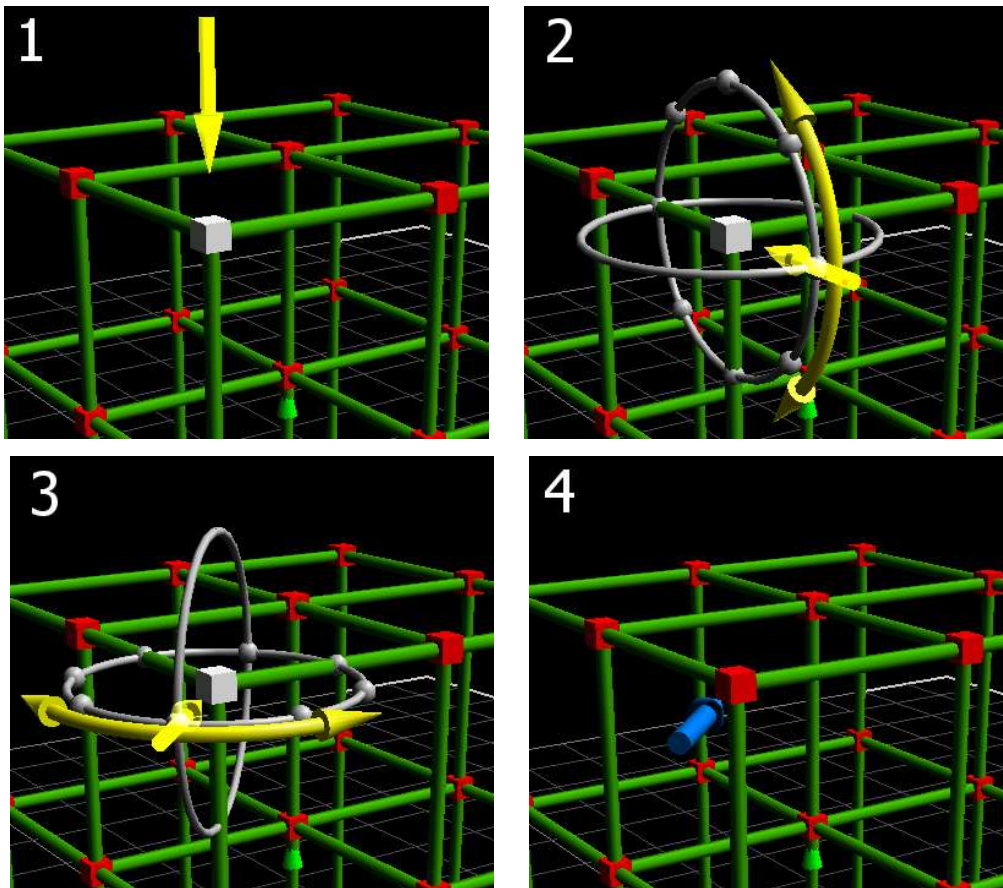


Figure 4.9: Applying loads

4.3 Graphical design

The focus in this evaluation has been on interaction design in the 3d user interface. Many of the guidelines available for graphical design apply only to 2d user interfaces. Some guidelines, such as those for color and icons can equally well be applied to 3d user interfaces.

Selection and highlight

A lot of work developing this prototype was done developing a new way of visualising the selection process. The previous method of using a single color was problematic, as the models created often used color to indicate material properties. To solve this a special method of object selection has been developed. In this method, an object selected is rendered twice. First the normal object is rendered using normal colors, second the object is rendered scaled to 120% and using a light grey color. The result is composited together. The result is an object that is highlighted regardless of initial object color. Figure 4.10 illustrate the new selection method.

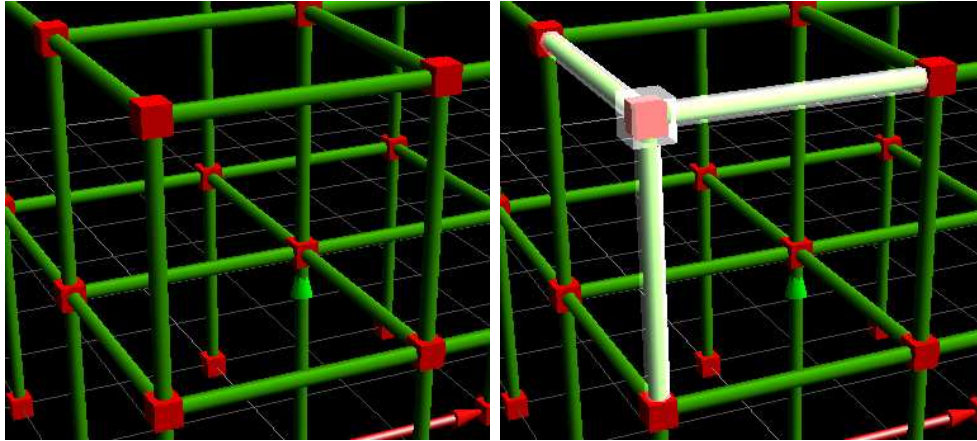


Figure 4.10: New object selection method

Color and lighting scheme

The building blocks in the prototype are given distinct colors or materials to make it easy for the user to distinguish between them. Nodes use a red material, beams a green material, and loads a blue material. Cursors and 3d icons all use a yellow material. Helper objects such as the guide circles in the load application method use a dull grey appearance, so that they don't draw the focus away from the cursor icons. The workbench also uses darker grey colors, so that the grid won't clutter the created model.

The entire model in the prototype is rendered with lighting. This is to enhance user experience of "assembling" a real model. The light is attached as a headlight to the current view direction. This will always produce a well lit model without any dark areas.

Chapter 5

Usability study

The usability study is divided into one part using real users evaluating the prototype and a second part evaluating the prototype using Shneiderman's 8 golden rules for interaction design. [4]

5.1 User test

The user test was conducted using the prototype designed in chapter 4. The test users consisted of 4 persons. 1 of the test users were already experienced with the current version of ObjectiveFrame the other 3 users had never used ObjectiveFrame before. The users were all from the civil engineering program. Architects and design students had been asked to participate in the test, but there were not volunteers.

Test setup

The test setup used a standard PC with keyboard, mouse and 3d graphics support. A normal DV video camera was used to record the screen, keyboard and mouse. To accurately capture user input a screen capture program recorded the user interactions for each task. The CamStudio [5] software was used. This is a free software which enables lossless screen recording without disturbing the user.

Test procedure

Before starting the test the users where informed of the reason for the test. They were also informed that the test was voluntary and that they could quit the test without being asked for a reason. I also told them that I would not answer any questions on the prototype during the test, unless they were completely stuck.

I told the test persons that if they could it would be good to "think aloud", but that didn't work out that well, so I asked the users after the test what they felt about the prototype and if they had any questions or suggestions.

User tasks

The users were given 7 tasks to complete. The tasks covered node creation, object selection, moving and copying and load creation. For each task a description on how to manipulate the view were given. The user tasks can be found in appendix A. The user interface for view manipulation has not been determined yet. Also the users were told that all operations in the prototype could be cancelled using the [Esc] key. The users were also instructed not to turn to the next page before the current task was completed.

Task evaluation

The user tasks were evaluated by studying the recorded screens and video. During the recording notes were taken about important events. Timing of specific tasks were done by studying the screen capture video and determining three timings:

Inactivity Amount of time where the user does not move the mouse. During this time the user is probably studying the task description.

Pre-operations Amount of time before the user selects the studied operation. For example moving around changing the view.

Operational time Amount of time for completing the task.

These timings can be found in the diagrams accompanying each task in the following sections.

Task 1 - Creating nodes

In task 1 the user should create a set of node according to the picture given. The two step node creation procedure seems to work. There is an initial time spent figuring out how to move the mouse, but when the first node has been created, the others are created very quickly. One of the users placed the first layer of nodes correctly but had problems placing the next layer correctly. The Esc-key was used successfully to abort the node creation process. An interesting side note: Instead of creating the nodes one by one, one user used the copy function for copying an entire node layer.

Most of the problems with the prototype in task 1 can probably be solved by enhancing the information displayed in the status view. Some kind of state indicator indicating showing the user where in the node creation process he or she is.

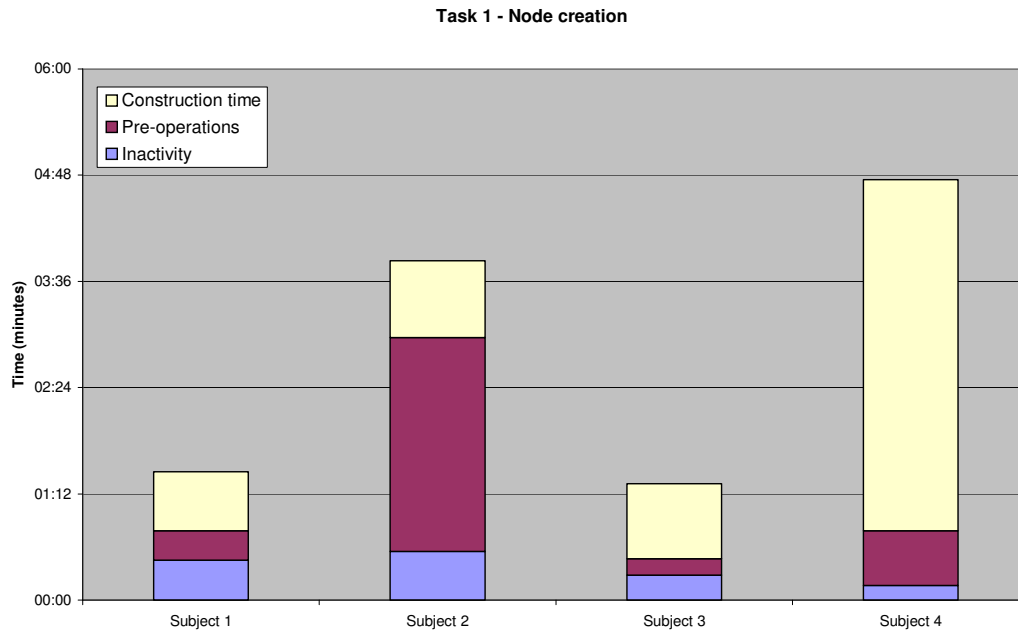


Figure 5.1: User times for task 1

Task 2 - Selecting and deleting nodes

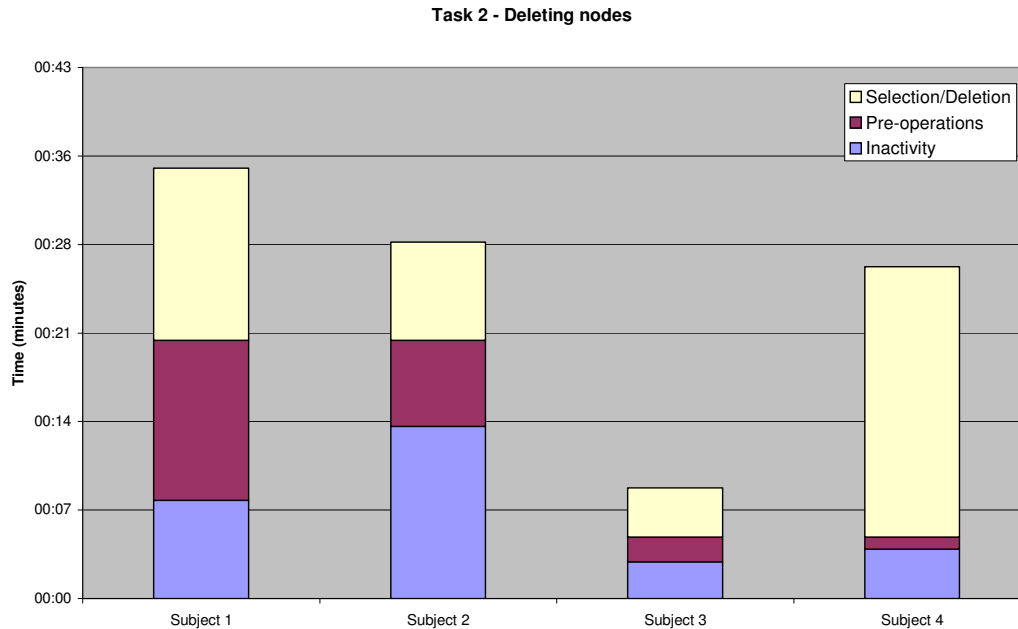


Figure 5.2: User times for task 2

In task 2 the user should select and delete the top layer of nodes in the given structure. The selection method seems to work fine. Some of the users used the view commands in this exercise to modify the view. One of the users didn't use the multiple selection, but instead deleted the nodes one by one.

It is unclear from these few tests how the selection process should be constructed. The first one is the one used in the prototype:

- Clicking on objects will add them to the selection. Clicking on already selected objects will remove them from the selection. Clicking outside the selection will remove all objects from the selection.
- Clicking objects will select them. Clicking on a non-selected object will deselect the previous one. Multiple selections are handled using modifier keys such as [Shift] and [Ctrl]. This method is common in many Microsoft applications, such as Word or Excel.

The method used in the prototype seems to work, and users don't have many problems with it. The highlighting of object when the mouse is over them seems to accelerate the selection.

Task 3 - Copying nodes

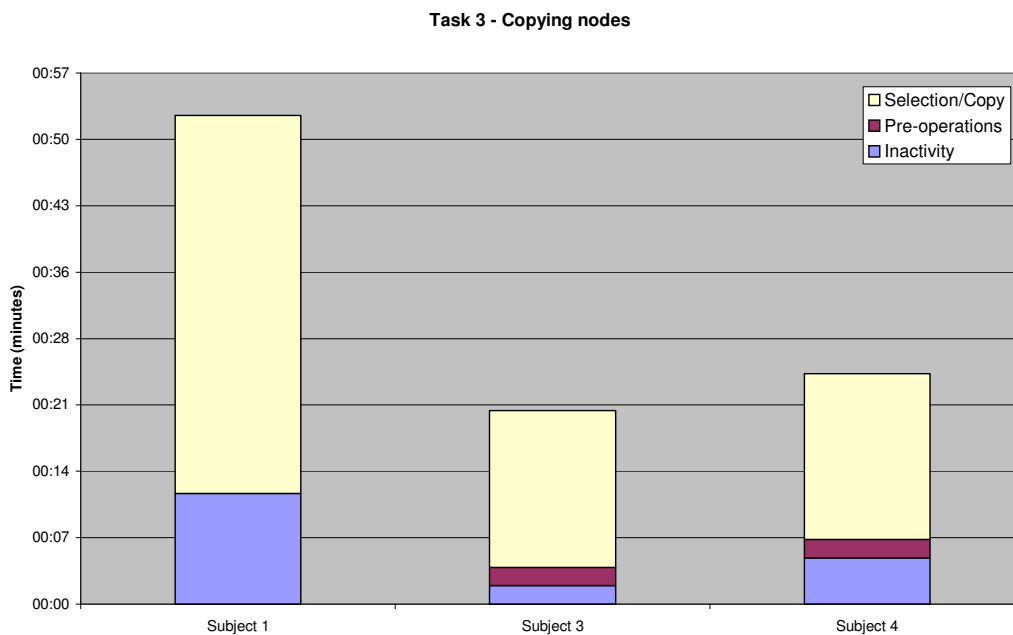


Figure 5.3: User times for task 3

In this task the users should copy the top four nodes in the given structure. The copy method works in approximately the same way as with moving nodes. Users seems to have little problem with this method. The special cursor that is shown when the mouse is over a node works well. Some of the users also used the select-deselect functionality of the prototype.

Most of the problems with the prototype in task 3 can probably be solved by enhancing the information displayed in the status view. Some kind of state indicator indicating

showing the user where in the copy process he or she is. The selection process should perhaps also be reviewed to solve some of the problems in this task.

Task 4 - Moving nodes

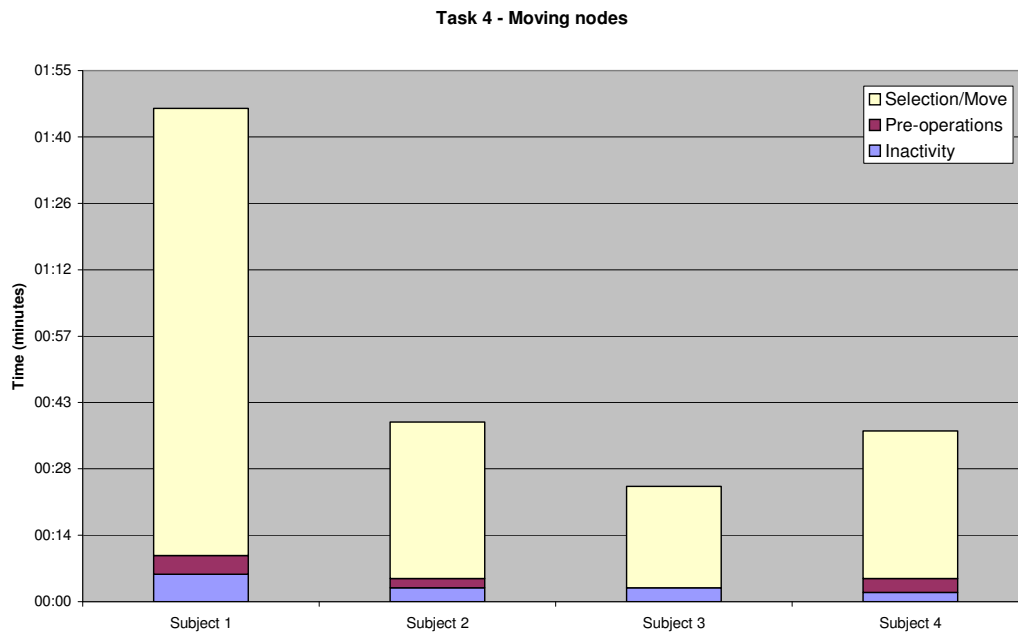


Figure 5.4: User times for task 4

In this task the users should move 2 nodes from the left and right sides. The move method works in approximately the same way as with moving nodes. Some users were probably expecting the selection to be removed when the move operation was finished. There were also some attempts to deselect nodes by clicking on the select button after the operation was finished. The special cursor that is shown when the mouse is over a node works well. Some of the users also used the select-deselect functionality of the prototype.

Most of the problems with the prototype in task 4 can probably be solved by enhancing the information displayed in the status view. Some kind of state indicator indicating showing the user where in the node creation process he or she is. The selection process should perhaps also be reviewed to solve some of the problems in this task.

Task 5 - Creating beams

In this task the users should create beam according the given structure. This process seems to have no problems. Highlighting of allowable objects seems to aid in the process of selecting the nodes for the beams. One user created beams in the wrong place, and used the select tool and deleted these without problems.

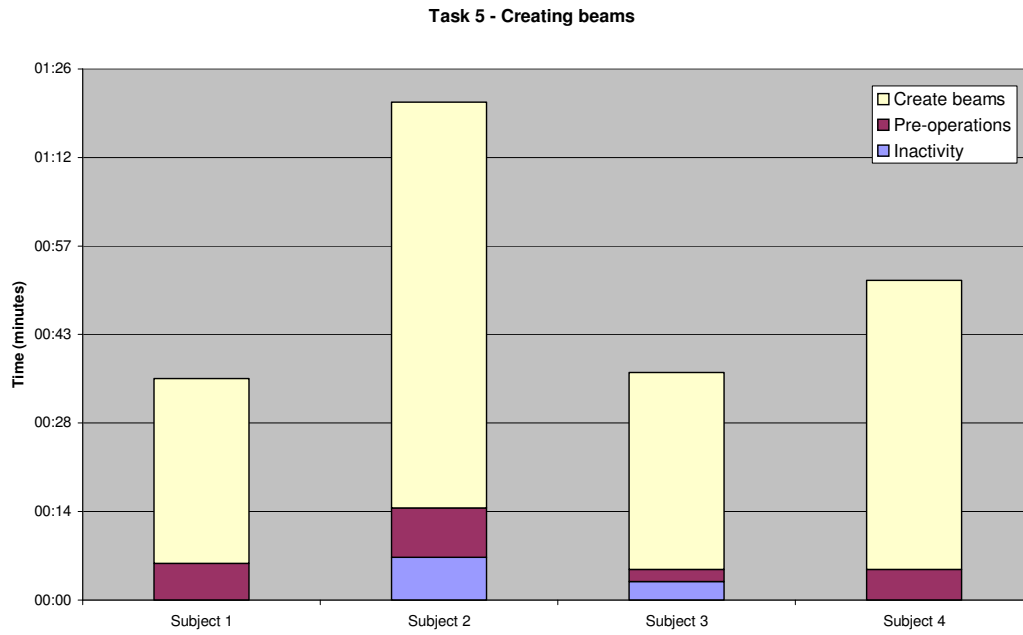


Figure 5.5: User times for task 5

Task 6 - Selecting and deleting beams

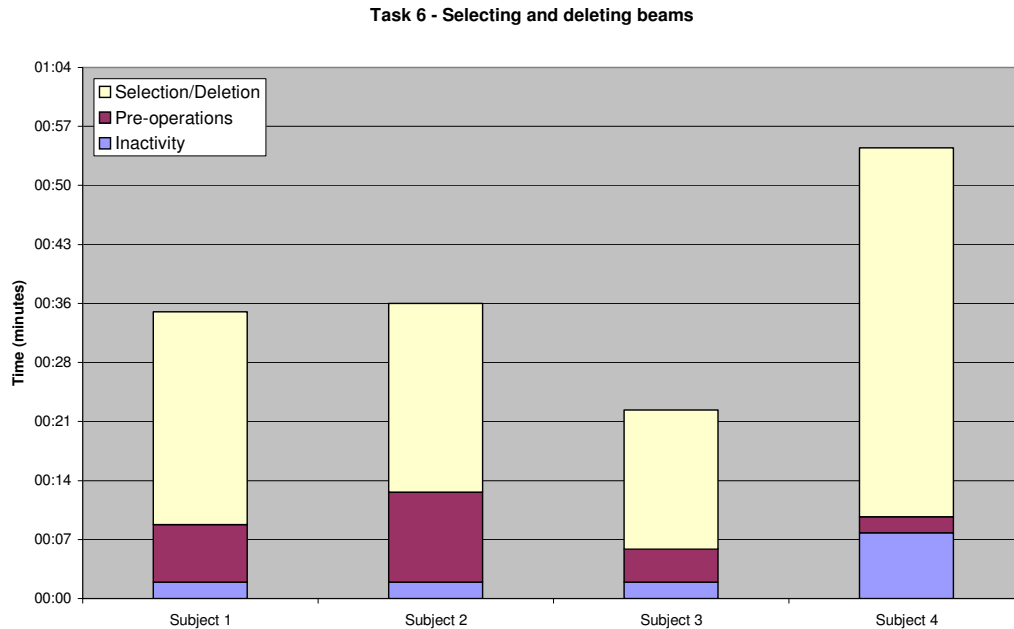


Figure 5.6: User times for task 6

In this task the beams shown in the picture should be deleted. Users quickly select the beams. If the beams are hidden in the 3d view the view is changed. Highlighting of selectable objects is speeding up the selection process. One user tried to delete the axis indicator. Also some problems with using the view functions were noted.

Visible buttons for changing the view should be added to aid user not familiar with how to change the view using the mouse and modifier keys. The axis indicator should probably also be made smaller or more different from the beams.

Task 7 - Creating loads

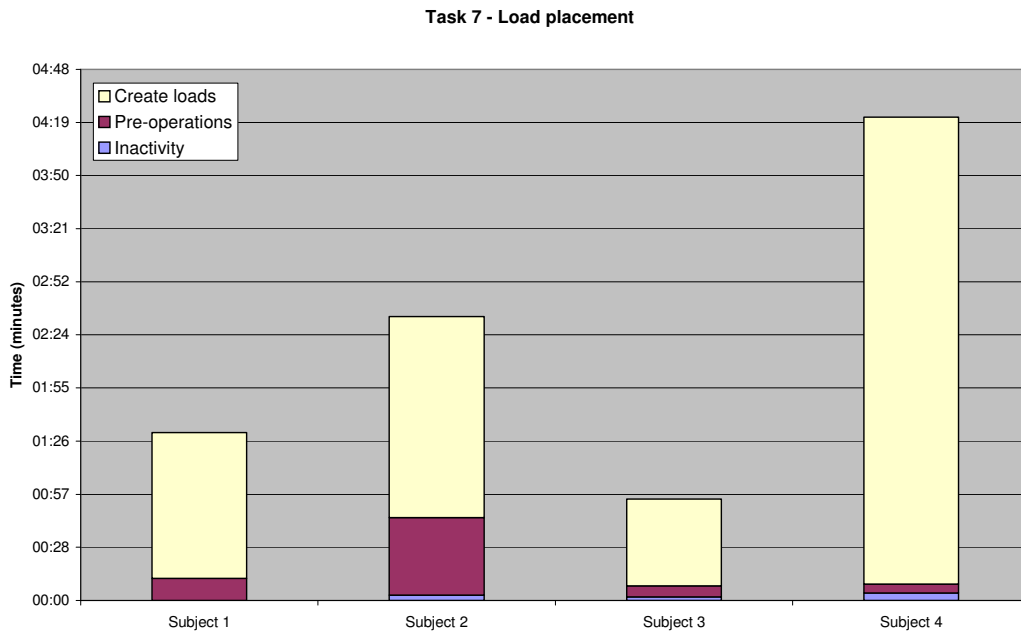


Figure 5.7: User times for task 7

In this task the users should add loads to the structure given. All of the users seems to figure out how to use the load widget. There seems to be a mapping problem in the widget rotational control. Two of the users had difficulties figuring out how to rotate the load. There were also problems when a load was created in the wrong direction, but this was mainly related to the fact that the prototype does not support deleting nodes.

The mapping for the rotational widget should be reviewed, so they are more easily interpreted.

5.2 Heuristic evaluation

In this section the prototype application will be evaluated using the eight golden rules for interaction design according to Shneiderman [4].

Consistency

The prototype takes advantage of consistency by using the same methods for moving and placing objects in 3d. Node creation, move and copy functions all use the same cursors and methods. Cursors are also shown using the same color and graphical design. The Esc-key can be used throughout the different operations in the prototype.

Shortcuts for experienced users

The prototype supports experienced user by providing shortcuts to the commands. By pressing special keys the experienced user quickly can switch between the different states of the application. Also movement in the Z-direction can be reached by holding down the [Shift] key.

Informative feedback

Feedback is provided in the status view of the application. In some cases the feedback could be improved. For example when placing objects in 3d there should be some indicator of the state of the process. Also confirmation dialogs should perhaps be added before objects are deleted. This could be implemented as an option in the application, to let more experienced user delete without confirmation.

Design dialogs to yield closure

The prototype application does not contain dialogs, but the process of creating nodes, copying and moving objects can be seen as a kind of dialogs. As such there should be more feedback of the process, as discussed in the previous section.

Offer error prevention and simple error handling

Many errors in the prototype is prevented by only allowing certain actions to take place, such as only allowing nodes to selected when creating beams. The highlighting scheme used in the object selection process is also helping the user to place the mouse more accurately.

Permit easy reversal of actions

The prototype application only support very limited action reversal. This reversal is in the form of the Esc-key permitting the current operation to be cancelled. This is an area which should be enhanced in the new version of ObjectiveFrame.

Set the user in control

The prototype is designed in such a way that it is always the user that has the control. The concept of "virtual workbench" supports this by letting the application be the "Meccano kit" letting the user assemble models using the building blocks provided.

Reduce short term memory use

Most operation and commands are clearly visible. Most information is available in the user interface. Hint boxes provide additional information on the functionality of the buttons and tools. The user is provided with additional instructions in the status view about what he has to do to complete the operation.

Chapter 6

Conclusions and improvements

ObjectiveFrame has shown potential as an educational software. Unfortunately it was designed to be a general application tool with engineers as the main user group. Used in courses at the architectural and design schools at Lund University has shown that the concepts used in ObjectiveFrame was not suitable for this category of students.

This work has illustrated that ObjectiveFrame can be modified to be a more direct application, suitable for an educational setting. It has also been shown that by doing a simple user testing and a heuristic analysis a lot of information on the problems of a user interface can be found.

Through the information provided in the user test and the heuristic evaluation the following points will be improved in the final version of ObjectiveFrame.

State information Some kind of indicator should be added to display in what state the current operation is in. In the node creation process, information on which step the application is currently in should be clearly visible together with information on how to complete the step.

View handling The view handling should be improved by supporting view manipulation using visible buttons in the application, in addition to using the mouse and modifier keys.

Improved axis indicator A more distinguishable axis indicator should be designed so that it won't be mistakenly taken for a beam object.

Selection The process of how selection of objects should be done. No clear method can be determined by the results in the evaluations.

Confirmation dialogues Critical operations should be preceded by optional confirmation dialogues.

Mapping of rotations The mapping of rotation angles when creating loads should be improved, to make the load placing a more intuitive process

Appendix A

Test instructions and tasks

The following pages show the actual instructions and tasks used in the usability evaluation.

Allmän information

Detta test syftar till att utvärdera ett antal metoder för gränsnitt i 3D för inmatning av noder, stänger och laster.

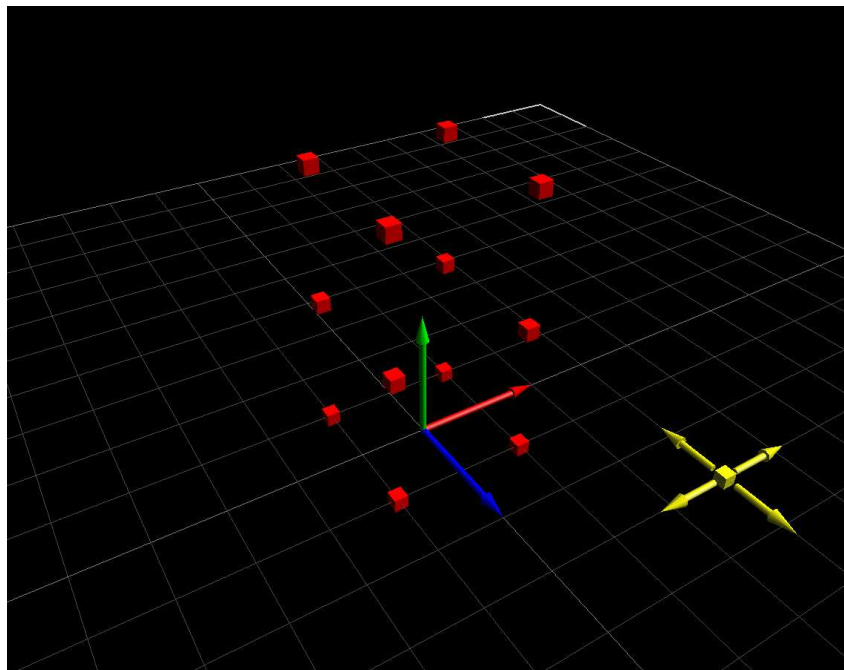
Testet är frivilligt och du kan när som helst avbryta testet utan att känna att du behöver ge någon förklaring.

Testet består av ett antal uppgifter som skall lösas. Varje uppgift är oberoende av det föregående och prototypen kommer att återställas till ett givet utgångsläge för varje uppgift.

För att vi bättre skall kunna förstå hur du tänker när du använder prototypen kan det vara bra om du ”Tänker högt”, dvs berättar hur du resonerar och vad du kommer att göra. Testledaren svarar gärna på frågor av allmän natur, däremot kommer frågor om uppgiften ej att besvaras förrän du inte kan komma vidare.

Uppgift 1

Skapa noder enligt följande figur. Placeringen av noderna behöver ej vara exakt.



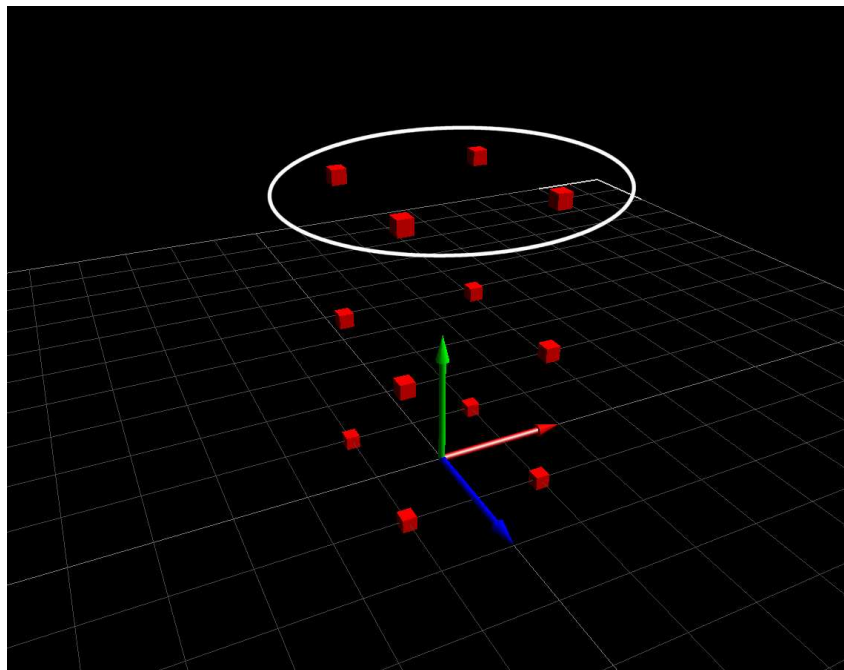
Instruktioner

3D-vyn kan manipuleras genom att använda mittknappen och flytta musen. Med **[Shift]** och **[Ctrl]** tangenterna kan ytterligare funktion nås enligt följande tabell.

Tangenkombination	Förklaring
Mittknapp	Roterar vyn
Mittknapp + [Shift]	Zoomar in och ut i vyn
Mittknapp + [Ctrl]	Panorerar vyn

Uppgift 2

Ta bort de noder som är markerade med en vit cirkel i följande figur.



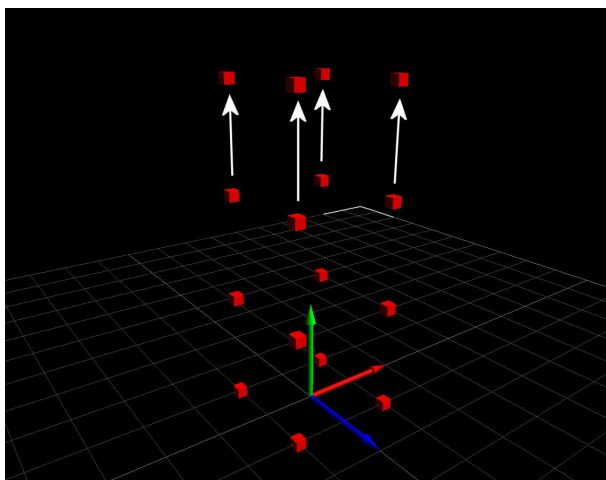
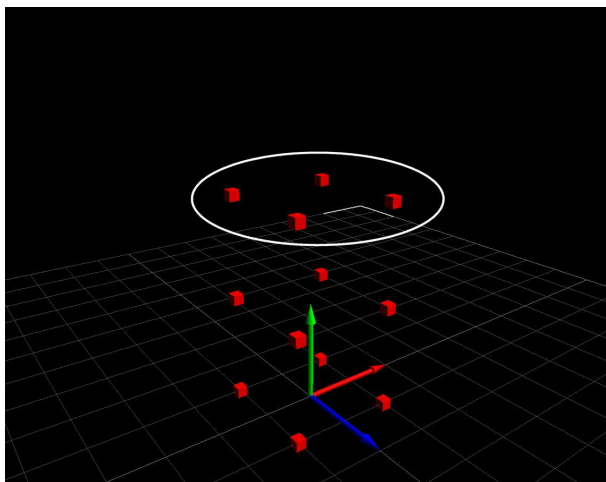
Instruktioner

3D-vyn kan manipuleras genom att använda mittknappen och flytta musen. Med **[Shift]** och **[Ctrl]** tangenterna kan ytterligare funktion nås enligt följande tabell.

Tangenkombination	Förklaring
Mittknapp	Roterar vyn
Mittknapp + [Shift]	Zoomar in och ut i vyn
Mittknapp + [Ctrl]	Panorerar vyn

Uppgift 3

Kopiera noderna markerade med vit cirkel i första bilden till positionen i den andra bilden.



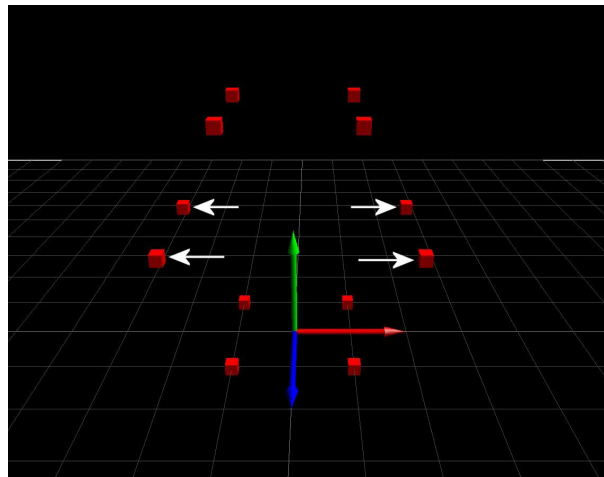
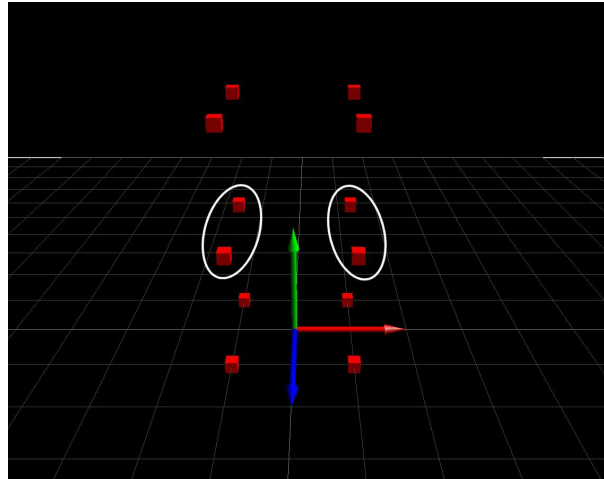
Instruktioner

3D-vyn kan manipuleras genom att använda mittknappen och flytta musen. Med **[Shift]** och **[Ctrl]** tangenterna kan ytterligare funktion nås enligt följande tabell.

Tangenkombination	Förklaring
Mittknapp	Roterar vyn
Mittknapp + [Shift]	Zoomar in och ut i vyn
Mittknapp + [Ctrl]	Panorerar vyn

Uppgift 4

Flytta de markerade noderna i första bilden till positionerna i andra bilden.



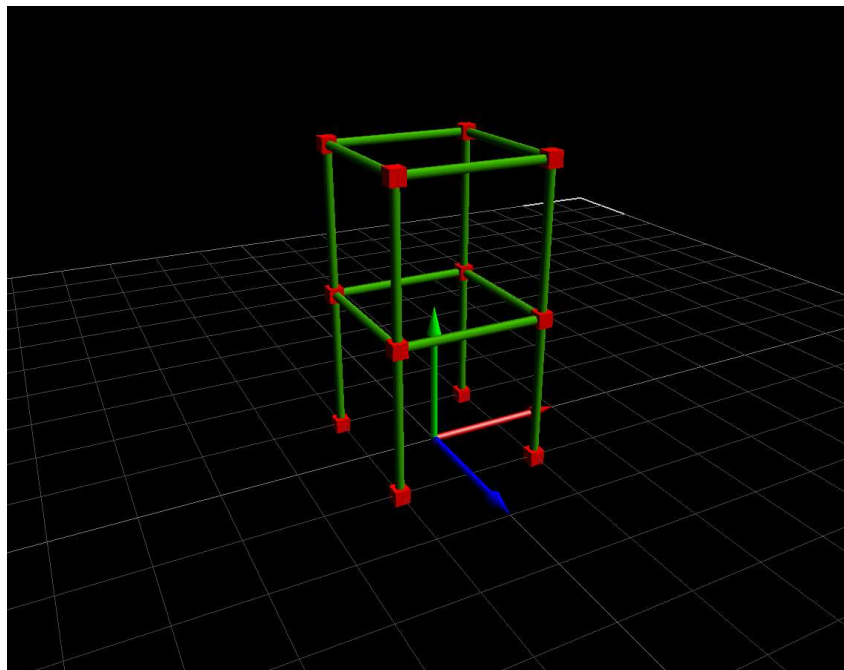
Instruktioner

3D-vyn kan manipuleras genom att använda mittknappen och flytta musen. Med **[Shift]** och **[Ctrl]** tangenterna kan ytterligare funktion nås enligt följande tabell.

Tangenkombination	Förklaring
Mittknapp	Roterar vyn
Mittknapp + [Shift]	Zoomar in och ut i vyn
Mittknapp + [Ctrl]	Panorerar vyn

Uppgift 5

Utifrån de givna punkterna skapa strukturen enligt nedanstående bild.



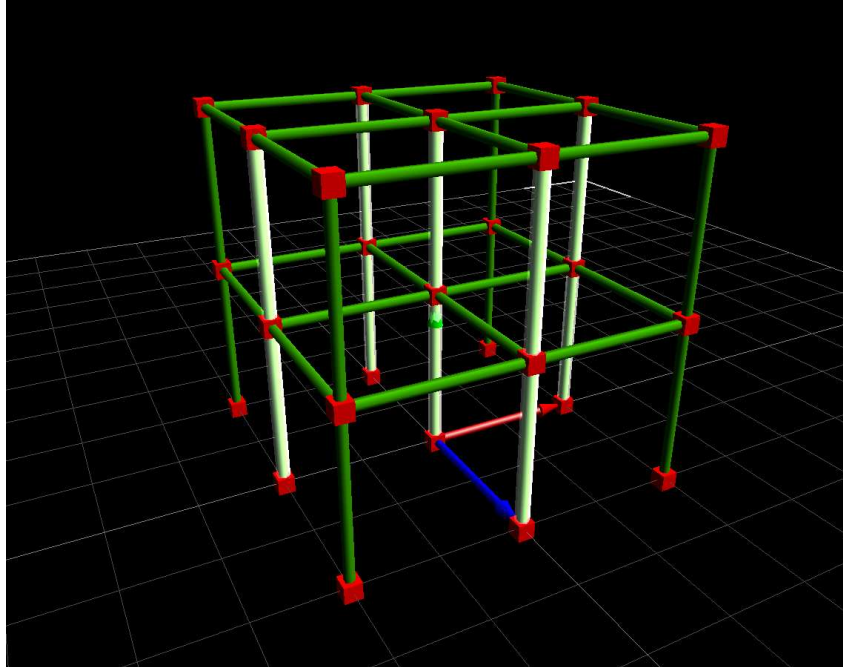
Instruktioner

3D-vyn kan manipuleras genom att använda mittknappen och flytta musen. Med **[Shift]** och **[Ctrl]** tangenterna kan ytterligare funktion nås enligt följande tabell.

Tangenkombination	Förklaring
Mittknapp	Roterar vyn
Mittknapp + [Shift]	Zoomar in och ut i vyn
Mittknapp + [Ctrl]	Panorerar vyn

Uppgift 6

Ta bort the markerade linjerna i strukturen enligt nedanstående bild.



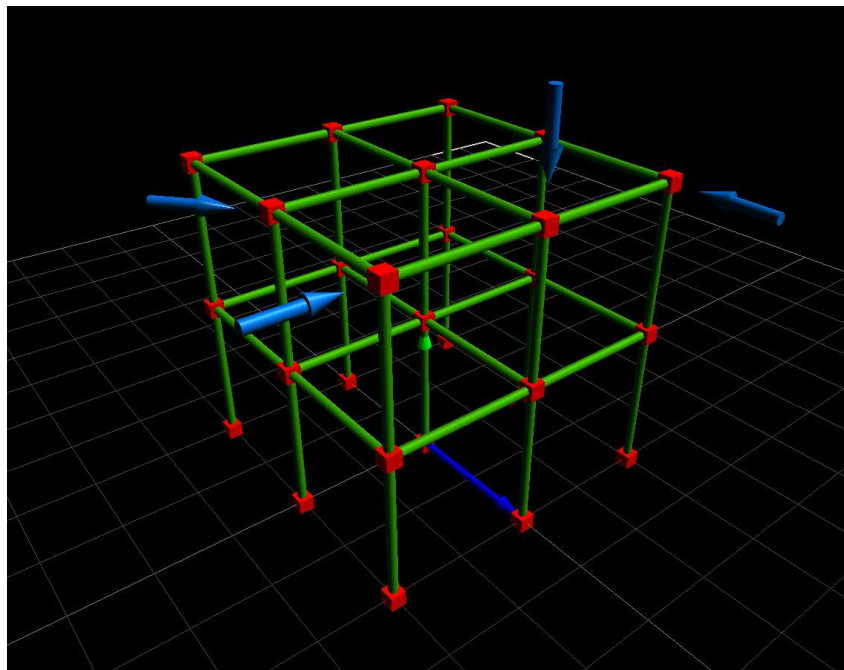
Instruktioner

3D-vyn kan manipuleras genom att använda mittknappen och flytta musen. Med **[Shift]** och **[Ctrl]** tangenterna kan ytterligare funktion nås enligt följande tabell.

Tangenkombination	Förklaring
Mittknapp	Roterar vyn
Mittknapp + [Shift]	Zoomar in och ut i vyn
Mittknapp + [Ctrl]	Panorerar vyn

Uppgift 7

Placera en laster enligt följande figur.



Instruktioner

3D-vyn kan manipuleras genom att använda mittknappen och flytta musen. Med **[Shift]** och **[Ctrl]** tangenterna kan ytterligare funktion nås enligt följande tabell.

Tangenkombination	Förklaring
Mittknapp	Roterar vyn
Mittknapp + [Shift]	Zoomar in och ut i vyn
Mittknapp + [Ctrl]	Panorerar vyn

Bibliography

- [1] The MECCANO Light Red and Green Period,
<http://www.meccanonut.com/lightred/index.htm>, 2003
- [2] The Fast Light Toolkit Home Page, <http://www.fltk.org>, 2003
- [3] The Interactive Visualisation Framework - Ivf++,
<http://www.gorkon.byggmek.lth.se/ivfweb>, 2003
- [4] Shneiderman B., Designing the User Interface - Strategies for Effective Human-Computer Interaction, Addison-Wesley, 1998
- [5] Rendersoft CamStudio 2.0, <http://www.rendersoftware.com/products/camstudio>, 2003

