

AoC Admin

User Manual

Contents

<u>1</u>	<u>INTRODUCTION</u>	<u>3</u>
1.1	AGE OF COMPUTERS HISTORY	3
1.2	AOC ADMIN	3
1.3	AOC GAME OBJECTS	3
<u>2</u>	<u>LOGIN</u>	<u>6</u>
<u>3</u>	<u>COMMON</u>	<u>7</u>
3.1	MENU BAR	7
3.1.1	FILE	7
3.1.2	VIEW	8
3.1.3	HELP	8
3.2	STATUS BAR	9
3.3	SAVE AND DISCARD BUTTONS	9
3.4	ADD AND REMOVE BUTTONS	10
<u>4</u>	<u>EDITOR VIEW</u>	<u>11</u>
4.1	LEVEL EDITOR	12
4.1.1	LEVEL TREE	12
4.1.2	LEVEL EDITOR	13
4.1.3	CREATING MAPS	14
4.2	QUESTION EDITOR	15
4.2.1	QUESTION TREE	15
4.2.2	QUESTION EDITOR	16
4.3	TEXT EDITOR	18
4.3.1	TEXT TREE	18
4.3.2	TEXT EDITOR	19
<u>5</u>	<u>ADMIN VIEW</u>	<u>20</u>
<u>6</u>	<u>SQL TRACE</u>	<u>22</u>
<u>7</u>	<u>CONFLICT VIEW</u>	<u>23</u>

1 Introduction

This chapter contains a short introduction to Age of Computers, AoC Admin and the objects used throughout these applications. The rest of this document contains detailed descriptions of the windows and functions found in AoC Admin.

1.1 *Age of Computers History*

Age of Computers (AoC) is an online multiplayer game used in teaching of computer fundamentals at the Norwegian University of Science and Technology (NTNU). It supplements traditional auditorium lectures by a rich set of problems in a computer game like environment. The story of the game is linked to the historical periods (epochs) of computers, and the content is organized in rooms or places on a map. A chat window for each historical period is used for communication with other students and teaching assistants. The first version of AoC, AoC-1, was used in a class with 250 students the fall semester 2003 in the course TDT4160 "Datamaskiner Grunnkurs" (Computer Fundamentals). The response from the students was very positive, and new versions of the game were tested in the same course in the 2004 and 2005 fall semester. The project continues in year 2006, with AoC now at version 3.

1.2 *AoC Admin*

AoC Admin is a tool for *Age of Computers* designed to make administration of the game easier and more intuitive. The application provides support for administrating users, levels and game objects in an easy-to-understand way.

As explained, AoC was originally created as a tool for use in the course "Datamaskiner Grunnkurs". However, one of the motivators for creating AoC Admin was AoC's potential for use in other areas both inside and outside computer related environments. Being as it is today, AoC is more of a framework than a game itself, as all the contents in the game may be adapted to other settings.

1.3 *AoC Game Objects*

This section contains short descriptions of the different objects found in Age of Computers:

Level

The *Levels*, or *Maps*, are the landscape that the users move around. The levels are filled with objects that the users may either pick up (*Texts*, *Questions*) or walk through (*Teleports*, *Messages*).

Teleport Source

The *Teleport Sources* are visualized as red “spirals” on the Level. Whenever a user enters a Teleport Source, and has completed the required *Question Categories*, he is moved to the Teleport Source’s corresponding *Teleport Destination*. This transfer may be on the same Level, or it may be between different Levels. The Teleport Source’s text is displayed in Age of Computers’ chat console when the user enters it.

Teleport Destination

The *Teleport Destinations* are visualized as blue “spirals”. A user may only enter a Teleport Destination through a *Teleport Source*. If he tries to walk directly through a Teleport Destination, nothing happens. A Teleport Destination may have several Teleport Sources leading to it. The Teleport Destination’s text is displayed in Age of Computer’s chat console when the user enters it through a Teleport Source.

Message

The Messages are visualized as signs. Whenever a user walks over a Message, the Message’s text is displayed in Age of Computer’s chat console. The Messages are well suited for providing the players with hints as to how to make progress in the game.

Question Category

The *Question Categories* are visualized as question marks. A yellow question mark signals a Category that is not yet completed, while a green question mark signals a completed Category. Each Question Category contains a number of Questions, where a given number of these questions are required to answer correctly in order to complete the Category. When walking onto a Question Category in the map, one of the Questions in that Category is displayed, and the player is required to answer before moving on.

Question

There are four different *Question* types: *Multiple Choice Questions*, *Input Questions*, *Dark Questions* and *Auto Generating Questions*. All Questions have to be assigned to a *Question Category*. The players can see their correctly answered Questions in the Question Book. Each Question has a value, which is added to the player’s score when answered correctly.

Question Group

The *Question Groups* are provided to make possible a logical and readable structure among the *Question Categories* in the game. Each Question Group can contain several Question Categories and other Question Groups which gives the questions a hierarchical tree-structure.

Fact

A *Fact* is one of the two *Text* objects in the game. It is visualized by a scroll with an ‘F’. The Facts are meant to provide some kind of fact that is useful in its settings. Facts that have the “*Fact book*” Text Group as one of their ancestors are shown in the Fact Book in the game.

History

The other *Text* object is *History*. The History is visualized by a scroll with an ‘H’. The Histories are meant to provide relevant historical background. The Histories having the “*History book*” Text Group as one of their ancestors are shown in the History Book in the game.

TextGroup

The *TextGroups* are similar to the *QuestionGroups* in that they provide a hierarchical tree-structure to the *Texts*. Each TextGroup may contain several Texts or other TextGroups. However, one TextGroup can only contain a number of Texts *or* a number of other TextGroups.

2 Login



Figure 1: Login Frame

The Login-frame is the first screen to appear when starting AoC Admin. In order to get past this screen a database server and a web server must be running. For information on how to do this, consult the server manual.

Username

The user needs to have access to both the web server and the database server in order to log on the application.

Password

The passwords must be the same for both the database- and the web server.

Database server

The database server should be entered in the format <Server name>/<Database name>. The drop-down list contains previously used database servers. This list will be empty when first using the application. The default value for the database server is the last used server.

Web server

The web server should be entered in the format <Server name>/. The drop-down list contains previously used web servers. This list will also be empty the first time. The default value for the web server is the last used server.

Login

If all the information is correctly filled in, and the values are valid, clicking this button leads to the main application starting up using the provided servers.

Cancel

Clicking this button closes the application.

3 Common

This chapter describes the functions common throughout the application.

3.1 Menu Bar

The menu bar is located at the top of the application. It consists of three drop-down menus: *File*, *View* and *Help*. The functions of these menus are explained in the following sections.

3.1.1 File

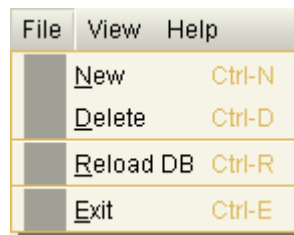


Figure 2: File Menu

New

Creates a new object. The types of objects that can be created are defined by the current view. See section 3.4.

*Short key: **Ctrl+N***

Delete

Deletes the currently selected object, both from the application and the database. See section 3.4.

*Short key: **Ctrl+D***

Reload DB

Saves all changes, before reloading all the information from the database. This command should be performed regularly, to avoid possible conflicts if more than one person is working with the same database at any time.

*Short key: **Ctrl+R***

Exit

Saves all changes, and closes the application.

*Short key: **Ctrl+E***

3.1.2 View

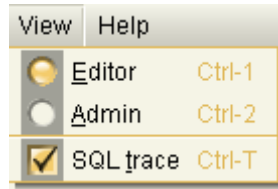


Figure 3: View Menu

Editor / Admin

Switches between the Editor- and Admin views (see chapters 4 and 5). Default value is Editor.

*Short keys: **Ctrl+1** / **Ctrl+2***

SQL trace

Sets the SQL trace window (see chapter 6) visible or invisible. Default value is on.

*Short key: **Ctrl+T***

3.1.3 Help

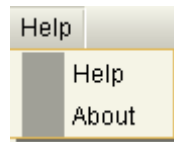


Figure 4: Help Menu

Help

Displays this user manual.

About

Displays information about the current version of AoC Admin.

3.2 Status Bar

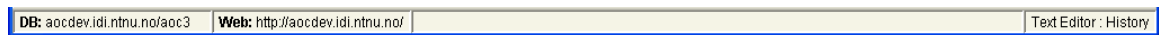


Figure 5: Status Bar

The status bar is located at the bottom of the application. Its three fields show the following information:

Database server

The name of the database server currently in use.

Web server

The name of the web server currently in use.

Current selection

The current view and the type of object currently displayed.

3.3 Save and Discard buttons



Figure 6: Save / Discard Buttons

The *Save* and *Discard* buttons are located directly above the status bar, on the right hand side. Their functions are:

Save

Saves all changes and writes the updated objects to the database. This action is also performed when the user selects a new object, or changes the view.

Discard

Discards the changes performed on the selected object. This action only discards changes made since the object was selected, or *Save* was last clicked. Changes in the tree structures (changing level or group) can not be discarded. (Only textual changes can be discarded).

3.4 Add and Remove buttons



Figure 7: Add / Remove Buttons

The *Add* and *Remove* buttons are located directly above the status bar on the left hand side, when the Editor View is selected (that is, when a tree is visible). Their functions are:

Add

Opens a popup menu where the user may select what kind of object he wants to create. The available object types in this list depend on the currently active editor. The new object is assigned as a child of the currently selected group or level in the tree, if valid. Otherwise, it is assigned to a predefined default value.

Remove

Removes the currently selected object both from the application and the database. If the object being removed is a parent group for other objects, these objects are either reassigned to the deleted object's parent or to a default group if the first is not valid. A popup is displayed where the consequences of this deletion are listed.

4 Editor View

The Editor View is the main view in the application, and where the objects used in Age of Computers may be created, modified and deleted. To enter Editor View, select ***View → Editor*** from the menu, or press ***Ctrl+1*** on the keyboard. The Editor View is the default view on startup.

The editor part of the application consists of three major parts: the Level Editor, the Question Editor and the Text Editor. Each of these parts has a tree on its left side, and the editor window on the right side.

In the tree panel, objects can be selected. In addition, the ***Add*** and ***Remove*** buttons can be used to create new objects, or delete existing objects. See section 3.4 for further explanation of these buttons.

In the editor panel, the object currently selected in the tree can be edited. Which information may be edited depends on the object selected and the currently selected editor. If the currently selected object is not an object belonging to the current editor, the editor is grayed out, and editing is disabled.

The following sections explain in detail the functions of the three editors:

4.1 Level Editor

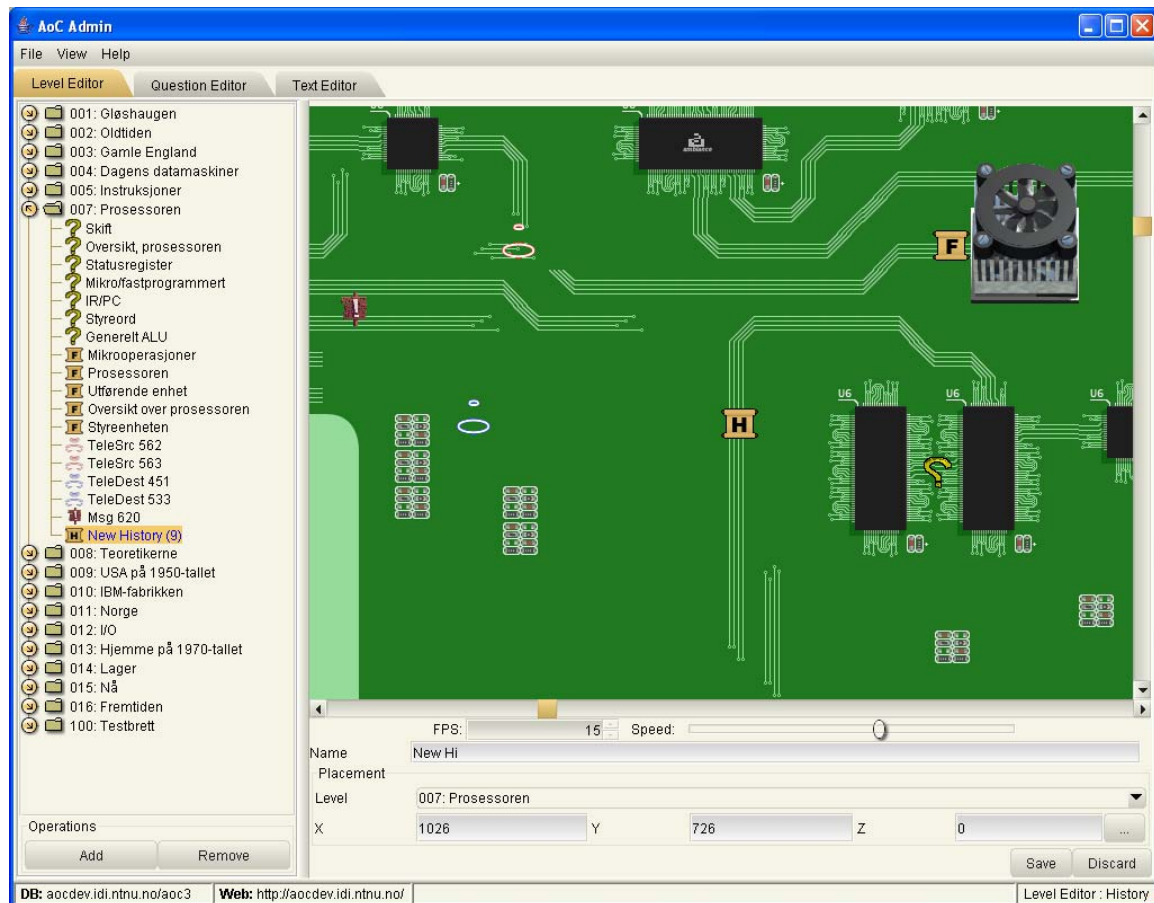


Figure 8: Level Editor

4.1.1 Level Tree

The level tree contains six different objects types that may be selected. Each type has its own icon. These types are:

- **Level** (Shown as folder)
- **Question Category** (Shown as question mark)
- **Fact** (Shown as scroll with an 'F')
- **History** (Shown as scroll with an 'H')
- **Teleport Source** (Shown as two red circles)
- **Teleport Destination** (Shown as two blue circles)
- **Message** (Shown as sign)

Selecting an object in the tree makes this object the active object, and enables it for editing in the editor.

4.1.2 Level Editor

The Level editor consists of several components; some of which are only visible for specific object types, and some of which are visible for all object types. These components are:

Map panel

The map panel consists of a scrollable window showing the map of the selected level. If the selected object is an item on the level, the window will be scrolled towards the selected object on selection. To view the entire map, the user may scroll using the mouse wheel (hold down **CTRL** to scroll horizontally), the arrow keys or the scrollbars to the right and the bottom of the map panel. The user may select an object on the level both by clicking it in the tree, or by clicking it directly in the map.

By right-clicking the mouse somewhere in the map, the user may create a new object which is then placed directly at the point where the user clicked.

Below the map panel, two extra components are located:

- **FPS:** Frames per Second; the speed in which the map panel is redrawn when scrolling. A high FPS value leads to smoother scrolling, but requires more resources from the computer than lower values.
- **Speed:** How fast the map is scrolled (step size).

Information panel

The info panel changes according to the kind of object currently selected. The different fields of the info panel are:

- **Name:** The name of the currently selected object. Applies for *Levels*, *Question Categories*, *Facts* and *Histories*.
- **Map:** The map used for the selected level. Only applies for *Levels*.
- **Text:** The text displayed when the user walks over the object in the game. Applies for *Teleport Sources*, *Teleport Destinations* and *Messages*.
- **Destination:** The Teleport Destination to which the selected Teleport Source leads. Only applies for *Teleport Sources*.
- **Requirements:** The Question Categories that are required to complete before the user is allowed to enter this Teleport Source. Additional Question Categories may be added by clicking the *Add* button and removed by clicking the *Remove* button. Only applies for *Teleport Sources*.

Placement panel

The placement panel is used for all map items. It shows which level the item is located at, and its position on that level. The level can be changed by selecting a new level from the drop-down box, while the position may be changed both by typing in each coordinate manually, or clicking the "..."-button. This leads to a popup being opened, where the new position can be left-clicked, followed by clicking "OK".

4.1.3 Creating maps

AoC and AoC Admin use maps created using the external tool Mappy. This tool can be found at <http://tilemap.co.uk/mappy.php>. To make a new map available in AoC Admin, the FMP-file needs to be put in the /data/maps folder on the web server.

4.2 Question Editor

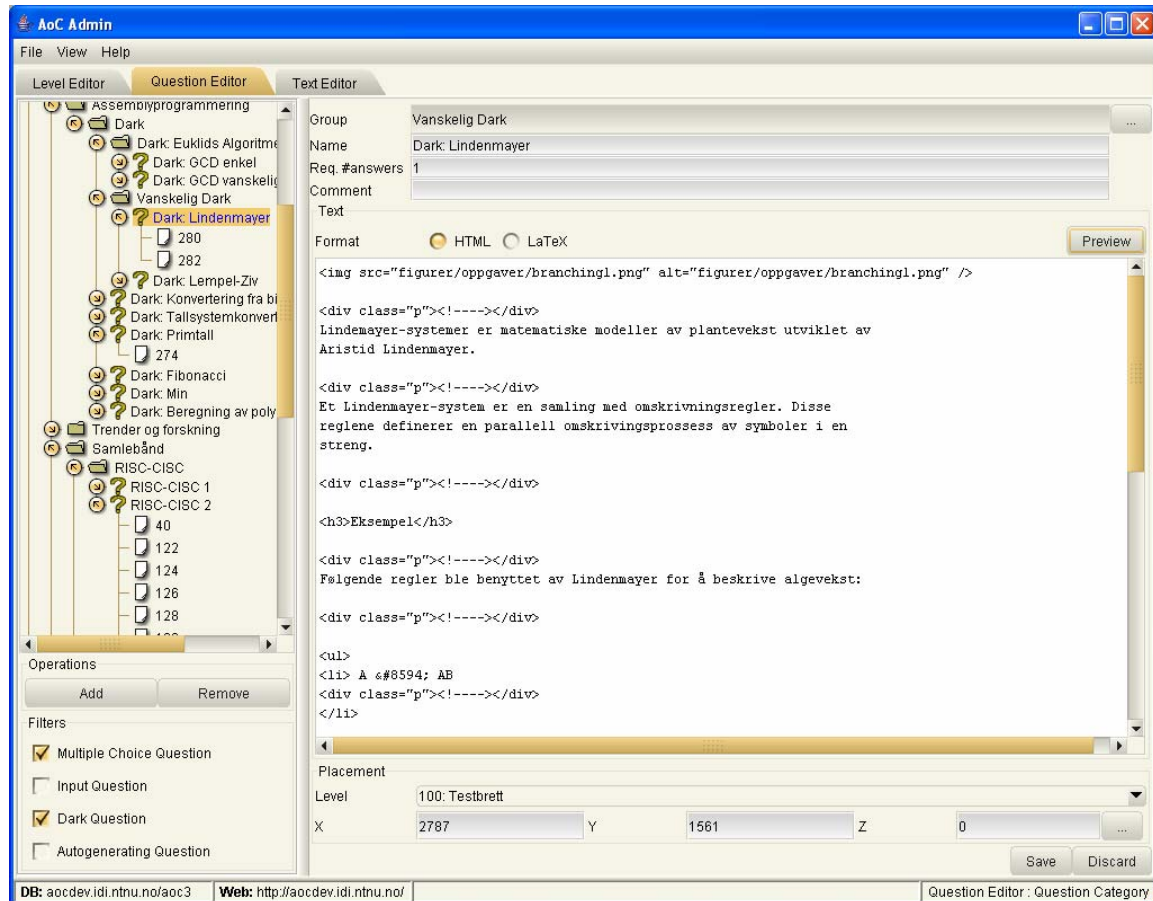


Figure 9: Question Editor

4.2.1 Question Tree

The question tree contains six different object types that may be selected. These six types are:

- **Question Group** (Shown as folder)
- **Question Category** (Shown as question mark)
- **Multiple Choice Question** (All questions shown as papers)
- **Input Question**
- **Dark Question**
- **Auto Generating Question**

Selecting an object in the tree makes this object the active object, and enables it for editing in the editor.

Filters

The question tree also, in addition to the “*Add*” and “*Remove*” buttons, contains four different filters. Unchecking one of the filters leads to all questions of that type being removed from the tree, while checking the filter leads to them being visible again. By default all filters are turned off (all questions are visible).

4.2.2 Question Editor

The question editor has six different ways of showing information, depending on the type of object selected in the tree. Each of these ways is listed in the following:

Question Group

When selecting a *Question Group*, the available fields are:

- **Group:** The parent group of the selected group (empty if the selected group is the root). The parent group can be changed by clicking the “...” button, selecting the new parent group and clicking “OK”.
- **Name:** The name of the selected *Question Group*.
- **Description:** A text describing the selected object.

Question Category

When selecting a *Question Category*, the available fields are:

- **Group and Name:** The same as for *Question Group*.
- **Req. #answers:** The number of correct answers on this category’s questions in order to make the category completed.
- **Format:** Can be either *HTML* or *LaTeX*. When selecting one or the other of these, the corresponding text is displayed. When *HTML* is selected, the “*Preview*” button may be clicked, and the text is previewed as it will be in the game. (Preview of *LaTeX* is not supported). *HTML* written in this category will be appended at the start of all questions assigned to this category in the game.
- **Placement:** Shows the level and position of the text. May be changed as described in section 4.1.2.

Question

All four question types have some fields in common. These are:

- **Category:** The *Question Category* to which the question is assigned. The parent category can be changed by clicking the “...” button, selecting the new parent category and clicking “OK”.
- **Value:** The number of points given to players who answer the question correctly.
- **Comment:** A comment to the question. May have multiple lines.

The fields unique for each question type are:

Multiple Choice Question

- **Format:** Same as for *Question Category*. The *HTML* field is the one used in the game.
- **Alternatives:** A table showing all the alternative answers to this question. One and only one alternative can be correct. The alternative text fields can be entered to change the texts. Click the checkbox next to the alternative to make it the correct alternative. The “*New*” button adds a new alternative to the bottom of the table, whereas the “*Delete*” button deletes the selected alternative(s).

Input Question

- **Format:** Same as for *Multiple Choice Question*.
- **Answer:** The correct answer to the question.

Dark Question

- **Question (Format):** Same as for *Multiple Choice Question*.
- **Code:** Java code that is used to validate the player’s code. Should implement the methods *getNumberOfTests()*, *getArchitecture()* and *implementTest()*.
- **Test Code:** Dark (Assembly) Code used to test whether the answer suggested by the player is correct. This code is available to the player and is meant to help the user test his code without having to run the full test.
- **Suggested Solution:** A suggested solution to the question.

Auto Generating Question

- **Code:** The code generating the question. Should be written in Java, and implement the methods *generateParameters()*, *getQuestionText()* and *checkSolution()*.
- **Suggested Solution:** A suggested solution to the question.

For more information on how to create Dark- and Auto Generating Questions, consult the AoC3 source code. For further questions contact the AoC3 technical staff.

4.3 Text Editor

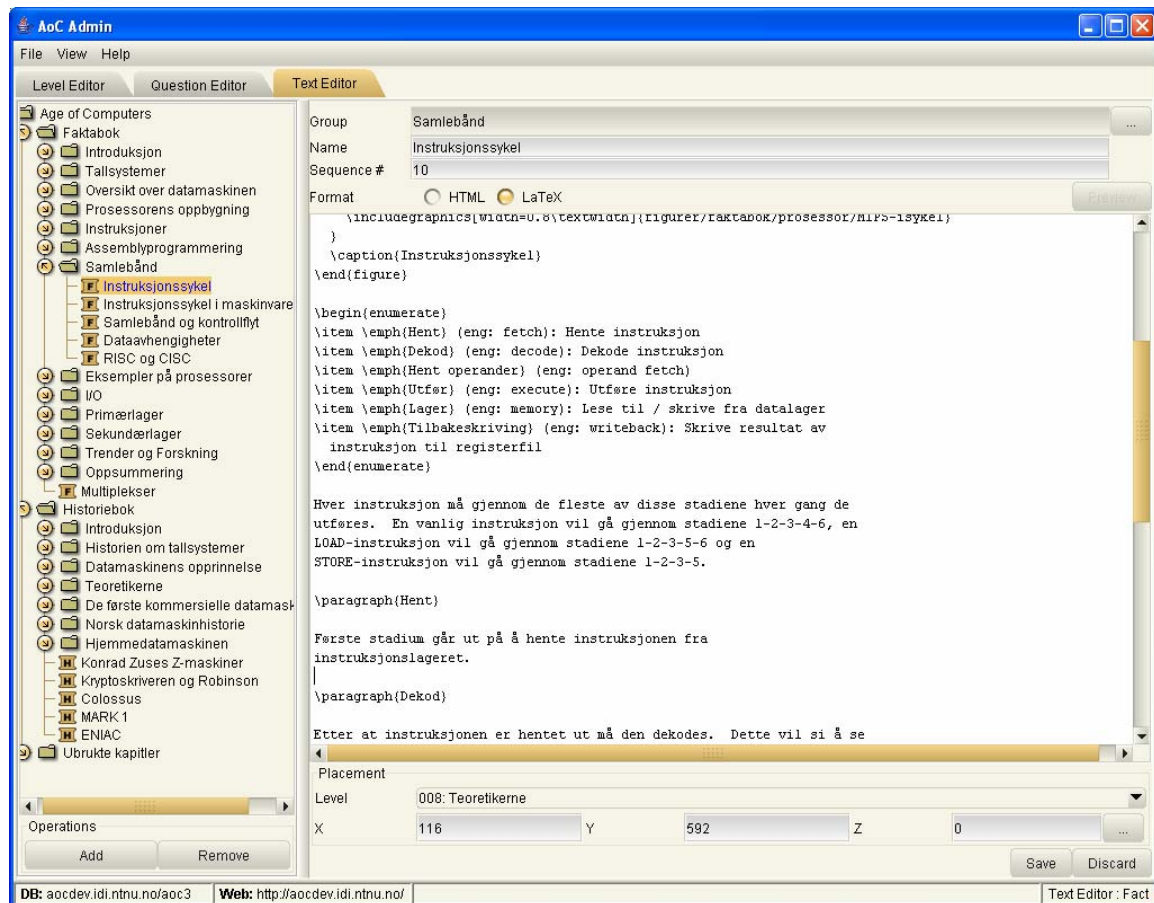


Figure 10: Text Editor

4.3.1 Text Tree

The text tree contains three types of objects. These are:

- **Text Groups** (Shown as folders)
- **Facts** (Shown as scrolls with an 'F')
- **Histories** (Shown as scrolls with an 'H')

Selecting an object in the tree makes this object the active object, and enables it for editing in the editor.

The items listed below the “Factbook” group defined in the database are the items displayed in the game’s *Fact Book*, while the items listed below the “Historybook” group are the items displayed in the game’s *History book*. These two groups are by default named “*Factbook*” and “*Historybook*”, respectively. Items placed outside these two sub-trees will not be displayed in the game.

4.3.2 Text Editor

The text editor has two different ways of showing information, depending on what kind of object is selected in the text tree. These ways are as follows:

Text Group

When selecting a *Text Group*, the following fields are available:

- **Group:** The parent group of the selected group (empty if the selected group is the root). The parent group can be changed by clicking the “...” button, selecting the new parent group and clicking “OK”.
- **Name:** The name of the selected group.
- **Sequence #:** This number determines the order of the text groups in the tree when the tree is built. This applies both for AoC Admin and in the game, where Text Groups having the same parent group are sorted in ascending order. New Text Groups have a default value of 0 on creation. This value should be changed immediately to control the order in which the Text Groups are listed.

Text

Text objects are either *Fact* or *History* objects. When selecting one of these, the following fields are available:

- **Group, Name and Sequence #:** Same as for *Text Group*.
- **Format:** Can be either *HTML* or *LaTeX*. When selecting one or the other of these, the corresponding text is displayed. When *HTML* is selected, the “*Preview*” button may be clicked, and the text is previewed as it will be in the game. (Preview of *LaTeX* is not supported). The *HTML* field is the one used in the game.
- **Placement:** Shows the level and position of the text. May be changed as described in section 4.1.2.

5 Admin View

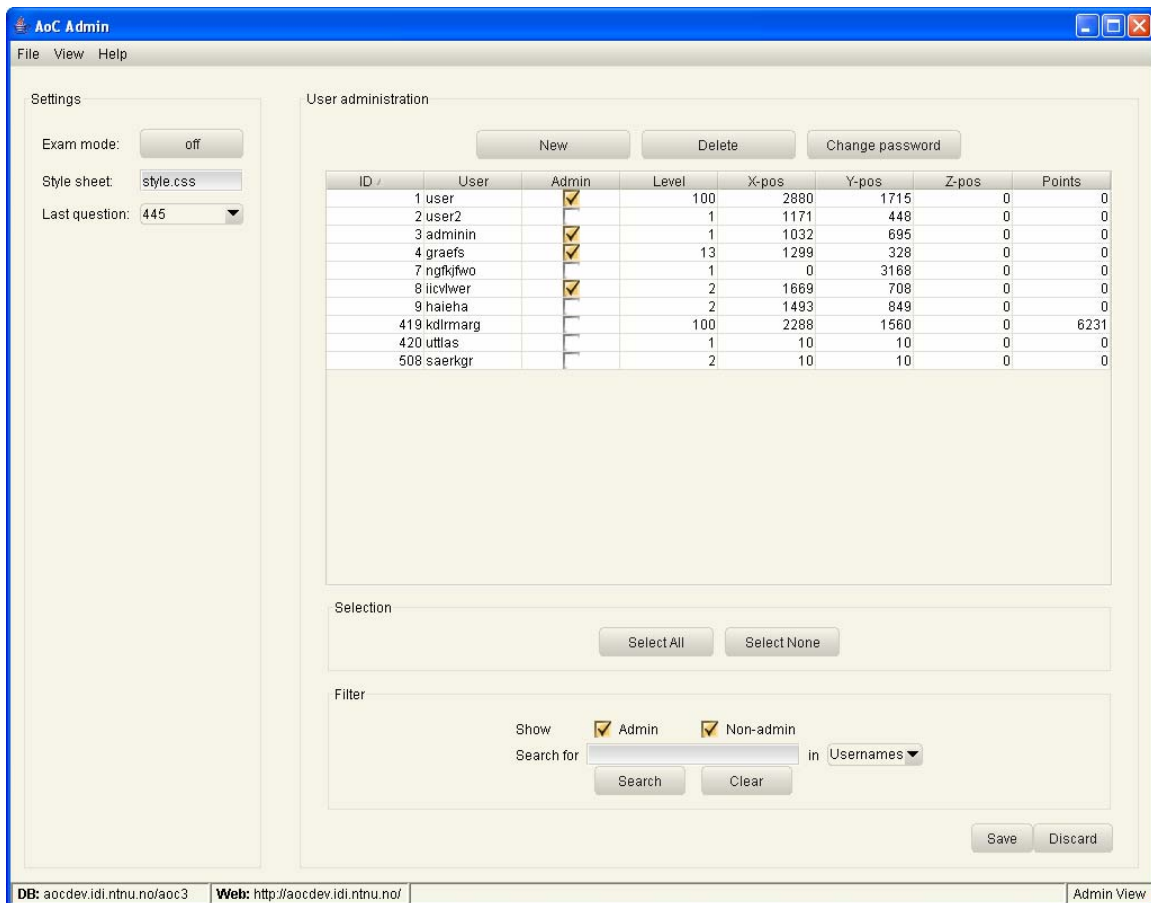


Figure 11: Admin View

The admin view consists of two main parts. Each of these parts is described in the following sections:

Settings

The settings section has three fields:

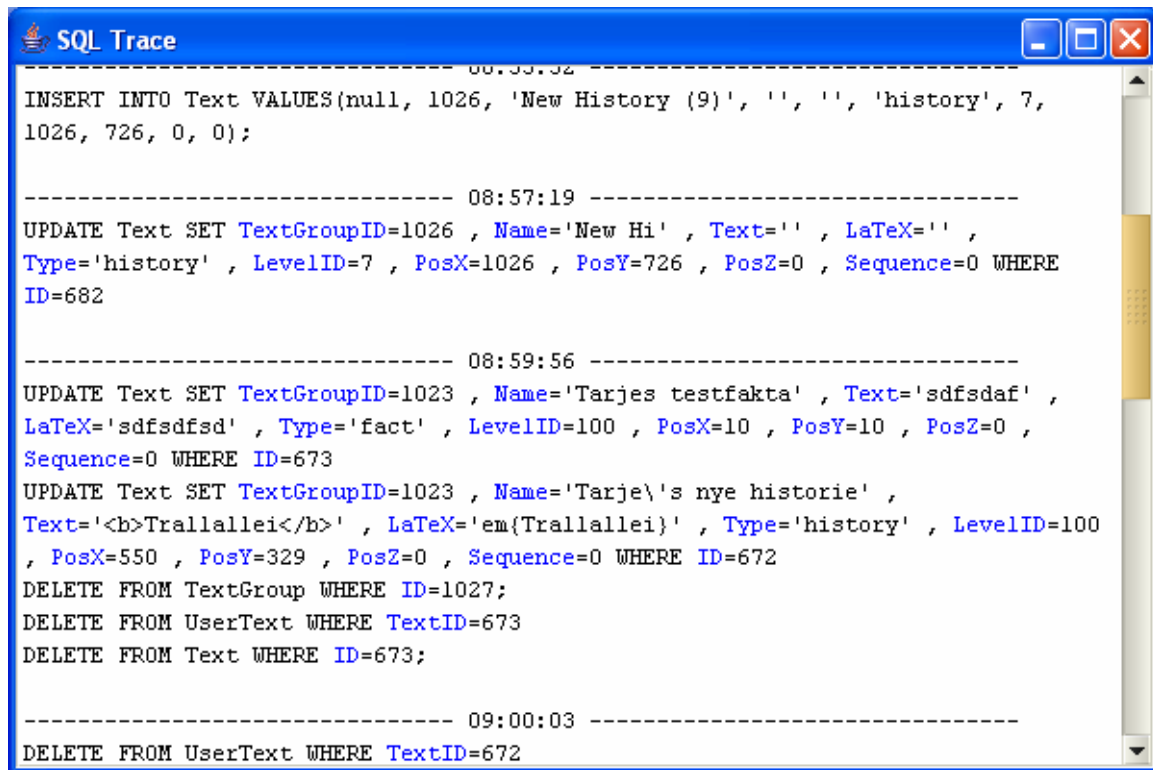
- **Exam mode:** When exam mode is turned on, all registered players who have completed the game will have access to all questions and answers.
- **Style sheet:** The style sheet containing the information on how *HTML* is to be displayed in the game. If this file is not specified, or the file specified does not exist, plain style will be used.
- **Last question:** This is the final question in the game. This is the question that has to be completed in order for a player to make use of the exam mode when turned on.

User administration

The components in the user administration section have the following functions:

- **User Table:** This table lists information about all the currently registered users in the database. The table can be sorted by all the columns by clicking the top of the wanted column. All values except *ID* and *Points* can be editing by selecting the wanted cell and entering the new information.
- **New:** This button adds a new user to the bottom of the table with default attributes. These attributes can be altered immediately after creation.
- **Delete:** This button deletes the selected user(s) from both the application and the database.
- **Change password:** When clicking this button, a popup is displayed where a new password can be entered for the selected user. The password must be entered (identical) two times before clicking “OK”.
- **Select All:** When clicking this button, all the users in the table are marked as selected.
- **Select None:** When clicking this button, all the users in the table are unselected.
- **Filter:** The filters can be used to find users fulfilling some criteria by typing in a search text, selecting a search field and clicking the “Search” button. The search can be reset by clicking the “Clear” button.

6 SQL Trace

The image shows a screenshot of a window titled "SQL Trace". The window has a blue title bar with standard Windows window controls (minimize, maximize, close) on the right. The main area of the window contains a text log of SQL queries. The queries are separated by horizontal lines and timestamps. The first query is an INSERT statement. The second is an UPDATE statement with a timestamp of 08:57:19. The third is another UPDATE statement with a timestamp of 08:59:56. This is followed by two more UPDATE statements and three DELETE statements. The final query is a DELETE statement with a timestamp of 09:00:03. The text is monospaced and color-coded, with keywords like INSERT, UPDATE, DELETE, and WHERE in blue, and values in black.

```
----- 08:55:32 -----
INSERT INTO Text VALUES(null, 1026, 'New History (9)', '', '', 'history', 7,
1026, 726, 0, 0);

----- 08:57:19 -----
UPDATE Text SET TextGroupID=1026 , Name='New Hi' , Text='', LaTeX='',
Type='history' , LevelID=7 , PosX=1026 , PosY=726 , PosZ=0 , Sequence=0 WHERE
ID=682

----- 08:59:56 -----
UPDATE Text SET TextGroupID=1023 , Name='Tarjes testfakta' , Text='sdfsda' ,
LaTeX='sdfsdfsd' , Type='fact' , LevelID=100 , PosX=10 , PosY=10 , PosZ=0 ,
Sequence=0 WHERE ID=673
UPDATE Text SET TextGroupID=1023 , Name='Tarje's nye historie' ,
Text='<b>Trallallei</b>' , LaTeX='em{Trallallei}' , Type='history' , LevelID=100
, PosX=550 , PosY=329 , PosZ=0 , Sequence=0 WHERE ID=672
DELETE FROM TextGroup WHERE ID=1027;
DELETE FROM UserText WHERE TextID=673
DELETE FROM Text WHERE ID=673;

----- 09:00:03 -----
DELETE FROM UserText WHERE TextID=672
```

Figure 12: SQL Trace

The SQL Trace is an external window making it possible to watch all information sent to the database. Whenever an object is created, updated or deleted, the corresponding SQL query is shown in the SQL Trace window.

For long attributes, like *Text* and *LaTeX*, the displayed values in the SQL Trace are cut down to a maximum of 30 characters in order to make the trace easily readable. The values written to the database are the same as shown in the AoC Admin main window.

If something goes wrong during an SQL Query, a red error message is shown in the SQL Trace. The last query performed before the error will then most likely have succeeded.

The SQL Trace can be shown or hidden through the *View* → *SQL Trace* menu in the Menu bar (see section 3.1.3).

7 Conflict View

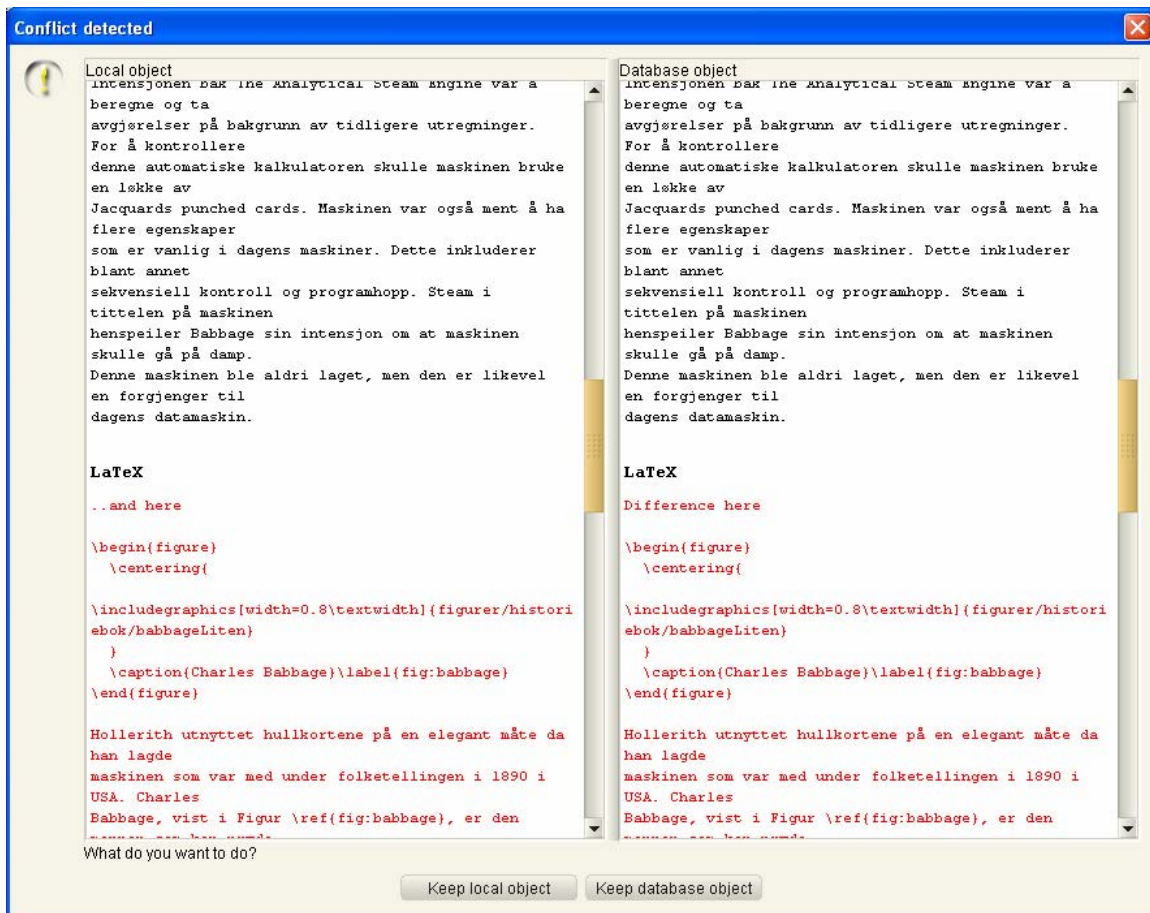


Figure 13: Conflict View

If two or more users are simultaneously working with AoC Admin, a conflict may arise if they make changes to the same object and both save the new object to the database. If this happens, a *Conflict View* window is displayed, where one or the other of these objects may be stored.

The left column shows the local object - the values the current user was trying to store in the database. The right column shows the object currently stored in the database, which has been changed since it was last loaded.

The differing attributes of the object is shown in red, so that it is easy to locate the conflicting values. Clicking the “*Keep local object*” stores the left hand values to the database, while clicking “*Keep database object*” keeps the values in the database and loads the database values into the local object to prevent a similar conflict on the next save.

The Conflict View requires the user to select one or the other of the conflicting objects. This means that information from the object not selected will be lost (merging is not supported). For this reason, frequent use of the “*Reload DB*” command (section 3.1.1) is recommended.