

# **The <e-Adventure> platform**

## *User's guide*

### Authors:

The <e-Adventure> team  
<e-UCM> research group  
[www.e-ucm.es](http://www.e-ucm.es)

### Latest version:

1.3RC-2

### Updated:

April 2011

**<e-Adventure>**

<http://e-adventure.e-ucm.es>

**<e-UCM>**  
*e-learning group*

## License



<e-Adventure> is freeware: can be used, redistributed, integrated in your Project (event for commercial purposes) and/or modified under the terms of the *GNU Lesser General Public License*, published by the *Free Software Foundation*, either version 3 or newer.

<e-Adventure> is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. The full license terms are available at:

<http://www.gnu.org/licenses/lgpl.html>.



This document is registered in Safe Creative (<http://www.safecreative.org>) under the terms of license “*Creative Commons Attribution-NonCommercial-NoDerivs 3.0*”. A full version of this license can be obtained at: <http://creativecommons.org/licenses/by-nc-nd/3.0/legalcode>

According to this license, this user’s guide can be shared (copied, distributed and transmitted) under the following conditions:

- Attribution: You must attribute the work in the manner specified by ) the <e-UCM> group, form Universidad Complutense of Madrid (the author) (but not in any way that suggests that they endorse you or your use of the work.
- Non-commercial: this user’s guide can’t be used for commercial purposes.
- No Derivative Works: You may not alter, transform, or build upon this work.

## Acknowledgements

<e-Adventure> version 1.3-RC2 has been partially funded by the following institutions:

- Spanish Ministry of Science, through the national research project TIN2010-21735-C02-02
- Spanish Ministry of Industry, Tourism and Trade, through the Avanza 2008-2011 program (TSI-020110-2009-170, TSI-020312-2009-27).
- European Commission, through the “Life Long Learning Programme-LLP” and the 7<sup>th</sup> Framework program (505469-2009-LLP-ES-KA3-KA3MP y FP7-ICT-2009-5-258169).
- Complutense University of Madrid, (research group nº 921340).
- Regional Government of Madrid, through the e-Madrid Project S2009/TIC-1650.

## Index of contents

License	3
Acknowledgements	3
<i>Index of contents</i>	4
<i>Index of figures</i>	7
<b>1. &lt;e-Adventure&gt; basics</b>	<b>9</b>
1.1. About <e-Adventure>	9
1.2. About this document	9
<b>1.3. Starting out with the editor: basic project management and running games</b>	<b>10</b>
1.3.1. Create a new project	11
1.3.2. Open an existing project, a game stored as an .ead file or a recently opened project	11
1.3.3. Exporting a project as an .ead file	12
1.3.4. Importing an existing game as a new project	12
1.3.5. Running a .ead game in the engine or from the editor	13
<b>1.4. Interaction in &lt;e-Adventure&gt; games</b>	<b>14</b>
<b>2. My first &lt;e-Adventure&gt; game</b>	<b>15</b>
<b>2.1. Chapters</b>	<b>16</b>
2.1.1 Importing Chapters	16
<b>2.2. Scenes and cut-scenes</b>	<b>16</b>
2.2.1. Adding a new scene	16
Background image (required)	19
Foreground mask (optional)	19
Background music (optional)	20
2.2.2. EXAMPLE: Defining the assets of a scene	20
2.2.3. Connecting scenes: adding an exit	22
2.2.4. Scene initial position	25
<b>2.3. Items</b>	<b>26</b>
2.3.1. Adding a new item	26
2.3.2. Interacting with items: actions	29
Examine	29
Grab	29
Use	29
Use with	29
Custom	30
Give to	30
Drag to	30
2.3.3. Information	31
2.3.4. EXAMPLE: Adding an item to a scene	32
<b>2.4. Set-items</b>	<b>34</b>
2.4.1. Creating a new set-item	34
2.4.2. Information	35
2.4.3. Adding a set-item to a scene	35
<b>2.5. Characters</b>	<b>35</b>



2.5.1. Create a new character	36
2.5.2. Character dialog configuration	37
2.5.3. Adding characters to the scene	38
2.5.4. Conversations	38
2.5.5. EXAMPLE: Editing a conversation	42
2.5.6. The player	44
<b>2.6. Books</b>	<b>44</b>
2.6.1. EXAMPLE: Create and edit a simple content book	44
2.6.2. EXAMPLE: Create and edit a formatted text book	47
<b>2.7. Cutscenes</b>	<b>49</b>
2.7.1. Slidescenes	49
2.7.2. Videoscenes	51
2.7.3. Connecting cutscenes with other scenes	52
<b>3. Extending basic games: Advanced features</b>	<b>54</b>
<b>3.1. Conditions and effects</b>	<b>54</b>
3.1.1. EXAMPLE: A simple condition	54
3.1.2. Variables	56
3.1.3. Global states	57
3.1.4. Condition edition dialog	57
3.1.5. What can be done with conditions?	59
3.1.6. Activating and deactivating flags: Effects	60
3.1.7. Setting the value of a variable	61
3.1.8. Other effects	62
3.1.9. EXAMPLE: Clarification about effects “consume object” and “generate object”. Grab objects and place them in different scenes.	63
3.1.10. Macros	65
<b>3.2. Organization of the element references in the scene</b>	<b>66</b>
3.2.1. Layers	66
3.2.2. Conditions	67
3.2.3. References list	67
3.2.4. Elements preview	67
<b>3.3. Export as a Learning Object (LO)</b>	<b>68</b>
<b>3.4. Export games as JAR (runnable Java programs)</b>	<b>70</b>
<b>3.5. Adaptation and evaluation profiles</b>	<b>71</b>
3.5.1. Assessment profiles	71
Regular rules	72
Timed rules	73
Report	74
3.5.2. Adaptation profiles	75
3.5.3. Choosing the active profiles of the chapter	76
<b>3.6. Active areas</b>	<b>77</b>
<b>3.7. Barriers</b>	<b>78</b>
<b>3.8. Player movement</b>	<b>79</b>
<b>3.9. Timers</b>	<b>81</b>
<b>3.10. Custom actions</b>	<b>83</b>
<b>3.11. Built-in art resources edition tools</b>	<b>85</b>
3.11.1. Animations editor	85

3.11.2. HTML editor	88
<b>3.12. Polygonal exits and active areas</b>	<b>89</b>
<b>3.13. Other features: error dialog and zoom panels</b>	<b>90</b>
Error dialogs	90
Zoom in panels	90
<b>4. Menu options</b>	<b>92</b>
<b>4.1. File menu</b>	<b>92</b>
<b>4.2. Edit menu</b>	<b>92</b>
<b>4.3. Adventure menu</b>	<b>92</b>
4.3.1. Check adventure consistency	92
4.3.2. Edit adventure data	93
4.3.3. Visualization sub-menu: customize GUI elements	94
4.3.3.1. Cursors	94
4.3.3.2. Action buttons	95
4.3.3.3. Inventory	96
4.3.4. Visualization sub-menu: select graphic configuration	96
4.3.5. Change adventure mode (Convert to 1st/3rd person)	97
4.3.6. Delete unused assets	97
<b>4.4. Chapters menu</b>	<b>97</b>
<b>4.5. Run menu</b>	<b>98</b>
<b>4.6. Configuration menu</b>	<b>98</b>
<b>4.7. About menu</b>	<b>98</b>

## Index of figures

Figure 1. Starting dialog of the <e-Adventure> editor.....	10
Figure 2. Choosing a project. a) shows the tab allowing for the selection of a project in the file system; and b) shows the selection of a recently opened project.....	11
Figure 3. Exporting a game as an .lead file. By default, the Exports folder will be selected.....	12
Figure 4. The <e-Adventure> engine start dialog.....	13
Figure 5. Example of a contextual interaction menu in an <e-Adventure> game.....	14
Figure 6. Edition tools in <e-Adventure>.....	15
Figure 7. Initial view of the <e-Adventure> editor, once a new project is created or an existing one opened.....	15
Figure 8. Scene edition shows a preview of all the scenes in the chapter. By default games include an empty scene.....	17
Figure 9. Scene edition panel, with the “Documentation” tab selected.....	18
Figure 10. Scene edition panel, with the “Appearance” tab selected.....	19
Figure 11. Background image (a) and mask (b). The back parts of the mask mark the parts of the background image that will be painted in the front of other elements and the white parts those that will be painted behind.....	20
Figure 12. Asset selection dialog for the scene background.....	20
Figure 13. Foreground mask selection dialog.....	21
Figure 14. Scene edition panel in the “Appearance” tab, after assets were selected for both the background and the foreground mask.....	22
Figure 15. Scene edition panel, selecting the next scene when adding an exit.....	23
Figure 16. Scene edition panel while adding a new exit. The exit can be dragged and re-sized as needed.....	24
Figure 17. Scene edition panel when previewing the scenes in a chapter. This shows all scenes in the game and the relation between scenes created as a consequence of exits from one to another.....	25
Figure 18. Scene edition panel while on the “Player movement” tab.....	26
Figure 19. Item edition panel in the “Appearance” tab with no selected asset.....	27
Figure 20. Item edition panel in the “Appearance” tab, when the item has an image attached.....	28
Figure 21. View/Edit image dialog.....	28
Figure 22. Item edition panel, “Actions” tab. This tab shows the list of actions and, after click the add button, the different kinds of actions that can be selected.....	30
Figure 23. Item edition panel, “Information” tab.....	31
Figure 24. Scene edition panel, “Element references” tab just after adding a new element reference.....	32
Figure 25. Scene edition panel at the “Element references” tab. After moving and scaling the “Book”, it now sits correctly over the table in the scene.....	33
Figure 26. “Set items” general view, with just one set-item in the chapter.....	34
Figure 27. Set-item edition panel in the “Appearance” tab after setting the image.....	35
Figure 28. Character edition panel in the “Appearance” tab allows for the edition of the characters animations.....	36
Figure 29. Character edition panel in the “Dialog configuration” tab. This shows the dialog to choose the color of the main font color.....	37
Figure 30. Dialog nodes. (a) represents the initial node in the conversation and (b) any other dialog node.....	38
Figure 31. Options node.....	39
Figure 32. End or finish node.....	39
Figure 33. Conversation edition panel with a node selected.....	39
Figure 34. Conversation line edition.....	40
Figure 35. Option panels for different nodes in the graph. (a) shows the options available for dialog nodes and end nodes, while (b) shows the options for option nodes.....	40
Figure 36. Edition of a dialog node.....	41
Figure 37. Edition of an option node.....	42
Figure 38. Creating loops in the conversation edition panel.....	43
Figure 39. A preview of a conversation.....	43
Figure 40. Simple content book edition panel in the “Book content” tab.....	44
Figure 41. The “Appearance” tab in the simple content book edition panel.....	45
Figure 42. Simple content book edition panel, after adding a title element.....	46
Figure 43. Simple content book edition panel, after adding an image paragraph.....	46

Figure 44. Simple content book edition panel, after resorting the elements in the book .....	47
Figure 45. A web page about raspberries.....	48
Figure 46. Preview of a formatted text book .....	49
Figure 47. Slidescene edition panel .....	50
Figure 48. Cutscene edition panel after adding a slide. The slide is the opening title for a game .....	51
Figure 49. Audio and video formats tested in <e-Adventure>.....	52
Figure 50. Cutscene edition panel in the “Cutscene end configuration” tab.....	53
Figure 51. Adding a new flag .....	55
Figure 52. Adding a new condition using an existing flag .....	56
Figure 53. Adding a condition to an action based on the value of a variable .....	57
Figure 54. Condition edition dialog (showing the contextual menu of a condition).....	58
Figure 55. Creating a complex expression with the condition edition dialog.....	59
Figure 56. An element with several appearances with different conditions .....	60
Figure 57. Editing the effects of an action, selecting the “Activate” effect.....	61
Figure 58. Effects to set the value of a variable or increase its value in 3 units .....	61
Figure 59. Table with all the effects available in <e-Adventure>.....	63
Figure 60. EXAMPLE: Pick up and place objects in different scenes. Adding to active areas to the scene "Restaurant." .....	64
Figure 61. EXAMPLE: Pick up and place objects in different scenes. Adding effects to “Book” object. ....	65
Figure 62. Edition of the layer of element references.....	66
Figure 63. Example of using layers to adjust depth in the scene .....	67
Figure 64. LTSC-LOM meta-data editor for Learning Objects.....	69
Figure 65. Assessment profile edition panel.....	72
Figure 66. Edition of a regular assessment rule.....	73
Figure 67. Edition of a timed assessment rule .....	74
Figure 68. A sample assessment report.....	74
Figure 69. Adaptation profile edition panel after adding a new rule .....	75
Figure 70. Adaptation rule edition panel, “LMS State” tab .....	76
Figure 71. Adaptation rule edition panel, “Game state” tab .....	76
Figure 72. Static initial state edition panel in an adaptation profile.....	76
Figure 73. Choosing the adaptation and assessment profiles in the chapter edition panel .....	77
Figure 74. “Active areas” tab in the scene edition panel .....	78
Figure 75. “Barrier” edition tab, in the scene edition panel.....	79
Figure 76. “Player movement” tab in the scene edition panel .....	80
Figure 77. Editing the influence area of an element reference in the “Element references” tab.....	81
Figure 78. Edition of advanced features, “Timers” tab.....	83
Figure 79. Example of normal and over images for a custom action button .....	84
Figure 80. Adding a new custom action .....	85
Figure 81. Animation editor .....	86
Figure 82. Properties of a frame in the animations editor.....	87
Figure 83. Properties of a transition in the animations editor .....	88
Figure 84. HTML editor .....	88
Figure 85. Creation of polygonal exits in a scene.....	89
Figure 86. Error dialog .....	90
Figure 87. Zoom controls .....	91
Figure 88. Check adventure consistency dialog.....	93
Figure 89. Adventure data dialog .....	94
Figure 90. User interface customization dialog, “cursors” tab .....	95
Figure 91. User interface customization dialog, “buttons” tab .....	95
Figure 92. User interface customization dialog, “inventory” tab.....	96
Figure 93. Graphic configuration dialog.....	97
Figure 94. Running a game in debug mode .....	98
Figure 95. Suggestions and comments dialog.....	99

## 1. <e-Adventure> basics

---

### 1.1. About <e-Adventure>

<e-Adventure> is a platform for the development of classic adventure computer games with educational purposes (although other types of 2D games can also be produced). This platform was developed by the <e-UCM> research group (<http://www.e-ucm.es>) at the Complutense University in Madrid. “Adventure games” (sometimes referred to as graphic or *point-and-click* adventures) include games such as the *MonkeyIsland™* or *Myst™* sagas, original titles of the genre. The platform is specially focused on this genre because it is considered one of the most suitable for educational purposes.

<e-Adventure> 1.3 RC2 is the latest version of the platform currently available. It can be downloaded at <http://e-adventure.e-ucm.es>. This web site also includes sample games and more information regarding the platform.

<e-Adventure> runs both in Windows (XP, Vista & 7) and Mac OS X (Leopard and Snow Leopard). Besides, a multi-platform version is available and should be suitable for Linux based systems. Windows users are strongly advised to use version 1.6.0\_18 of the Sun Java runtime environment if problems arise (this is freely available online at <http://www.java.com>). Mac users are encouraged to have their systems up-to-date.

Every <e-Adventure> distribution includes both the game engine and the editor. The game editor is the application that allows for the creation of games, while the game engine is used to run previously created games.

When running the <e-Adventure> installer three new folders will be added to the system user folder (e.g. “My documents” in Windows): *Projects*, *Exports* and *Reports*. The first folder is the default place where new projects are created; the second one is the default place for executable game exports to be stored; and *Reports* contains the evaluation reports generated when the games are played.


Please, before you start using <e-Adventure> make sure you update your Java JRE (Java Runtime Environment) if necessary (<http://www.java.com>). <e-Adventure> 1.2 was compiled for JRE 1.6 and it is not warranted to work on previous versions (JRE version 1.6.0\_18 is recommend and can be downloaded at <http://java.sun.com/products/archive/j2se/6u18/index.html>).

If you find <e-Adventure> of your interest, and want to find out about our publications and other investigations of our research group, you can visit our web site: <http://www.e-ucm.es>.

### 1.2. About this document

This document aims to be a complete user guide, including descriptions of every feature available in <e-Adventure>. Although features of the engine and editor are described, this document focuses on the use of the editor for the creation of new educational video games. We tried to cover every available feature, but some of them might be left out. We apologize for any inconvenience.

This is not the only documentation available about <e-Adventure>. You could find shorter tutorials and other learning materials on our website (<http://e-adventure.e-ucm.es>). You can also access the contextual

help pages that are embedded in the <e-Adventure> editor application. Just click on any info button like this  and context-sensitive information will be displayed.

This guide was prepared to be read from start to end, taking the reader through the different parts of the editor in an order that can be easily followed and that usually goes from the easiest to the hardest details. However, this guide can also be used as reference and for that a detailed table of contents is included which allows for a quick solution of doubts about the platform.

### 1.3. Starting out with the editor: basic project management and running games

To start using the <e-Adventure> platform, you should start up the editor. This will show the starting dialog (Figure 1). From this dialog we can create new adventures or open existing ones.

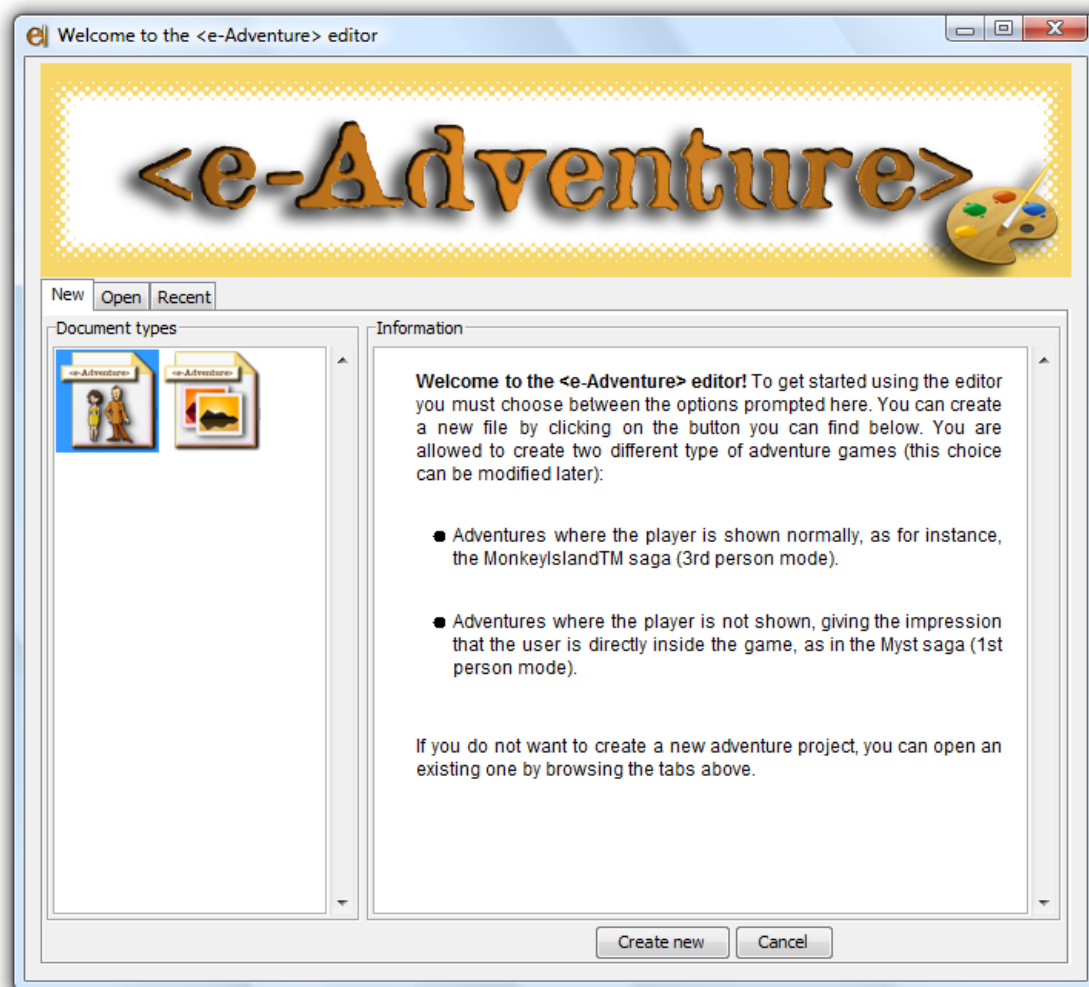


Figure 1. Starting dialog of the <e-Adventure> editor

### 1.3.1. Create a new project

<e-Adventure> games are managed as “Adventure projects” in the editor, which are nothing more than an empty .eap file and a folder that contains all the data of the game. These games can not be played in the engine, and for that they should be later exported as .ead files.

To create a new project we should click the “Create New” button. A dialog will appear asking for a name and destination (by default the *Project* folder in the user folder) for the project folder. The folder name should be made up of numbers, letters, white-spaces and [, ], (, ), \_, - symbols. Other characters, such as /, \, |, @, & or \$ are not allowed in the project folder name.

Two different game types can be selected, each represented by its own icon:

*Third person games:* These games, similar to the *MonkeyIsland*<sup>TM</sup> saga, have a visible player representation. The player is shown as an avatar in the game, which he/she can control. The movement of the player’s avatar requires time (the avatar must walk from one place to the other) and the text is shown directly above its head (just like in a comic book).

*First person games:* The player has no avatar in these games, thus exploring the game in the first person and with instant consequences for the different actions. These games are more alike to the *Myst*<sup>TM</sup> saga. This sort of game is usually better suited for photo-realistic games, in which photographs are used as the setting of the game.

### 1.3.2. Open an existing project, a game stored as an .ead file or a recently opened project

The start dialog also allows for the selection of an .eap file (or a project folder) to be opened (Figure 2, a), besides showing a list of recently opened project to select from (Figure 2, b). By default, the dialog will show the projects available in the *Projects* folder. This folder is destined to store all the projects that are created with the editor, but the user can chose to create or place projects in a different folder.

Once the file is selected, clicking the “open” button will start the editor with the selected game project. If the user tries to open an .ead file (a game already exported from the editor) the system will automatically import the game in the folder the user will be asked to choose.

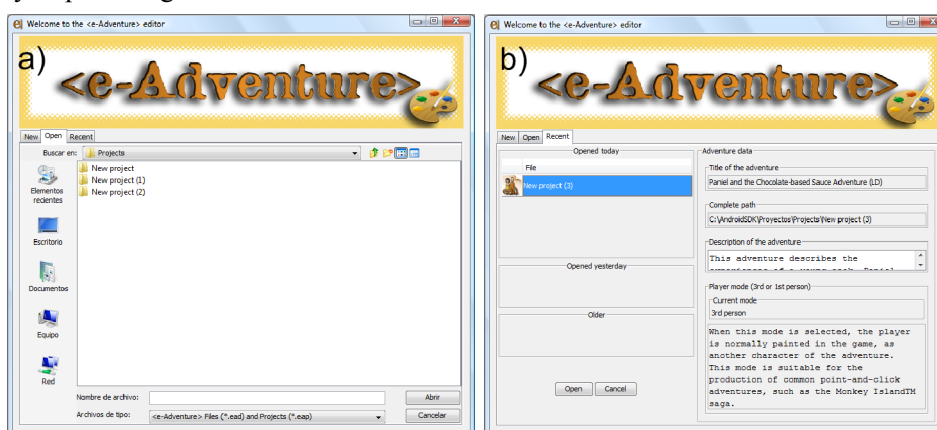


Figure 2. Choosing a project. a) shows the tab allowing for the selection of a project in the file system; and b) shows the selection of a recently opened project.



### 1.3.3. Exporting a project as an .ead file

The editor works with game projects, however, a project is made up of several different files and can make it hard to distribute the game among students. This is the reason why the game engine uses .ead files, which are zip files with all the game resources. To export a game project as an .ead file the following steps must be followed:

1) Check that the adventure is valid. This can be done from the “Adventure” menu in the menu-bar, choosing the “*Check consistency*” option. Should the adventure not be valid, the reported errors should be fixed before the adventure can be exported.

2) Export the game as an .ead file. This option can be found in the menu-bar at *File > Export project > Export as EAD file*. This will prompt us to choose a name and place for the game (Figure 3), which, after a few seconds, will be ready to be played in the engine. By default, games will be placed in the *Exports* folder.

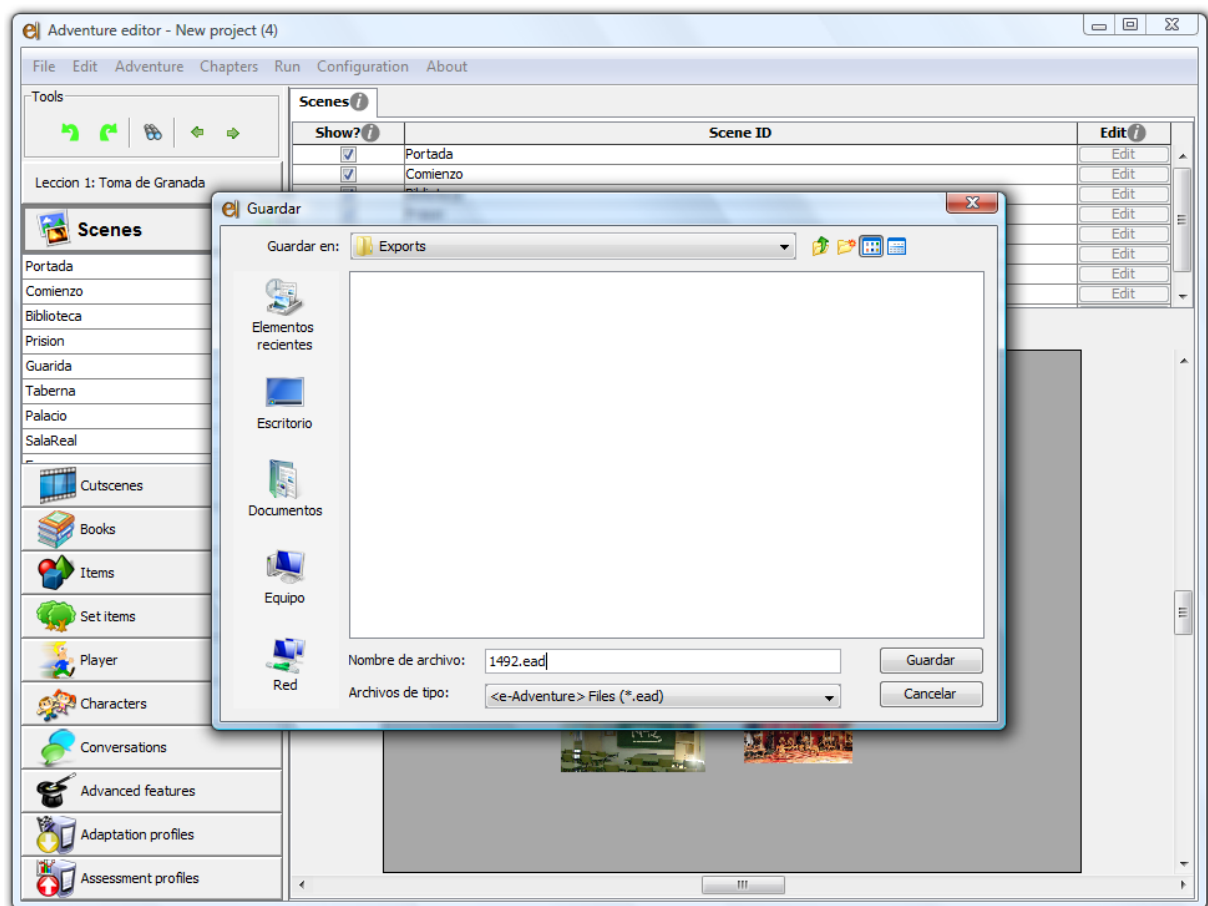


Figure 3. Exporting a game as an .ead file. By default, the Exports folder will be selected

### 1.3.4. Importing an existing game as a new project

Using the *File > Open* menu the starting dialog will be displayed, allowing us to open .eap or .ead files. When an .ead file is selected, we will be prompted to choose a folder where to extract its contents.



This menu can also be used to import games previously exported as Learning Objects (LO) or executable .jar files. To do this, the .zip or .jar extensions should be selected within the dialog.

### 1.3.5. Running a .lead game in the engine or from the editor

As stated earlier, adventure games can be exported as .lead files. These files are the ones supported by the game engine. To run the games in such files, the game engine can be launched and a dialog that allows for the selection of the game will appear (Figure 4). By default, the *Exports* folder will be selected, but this can be changed by click in the “Browse” button. This dialog will show all the .lead files in the selected folder, selecting one of them will show detailed information about the game and double-clicking will launch the game.

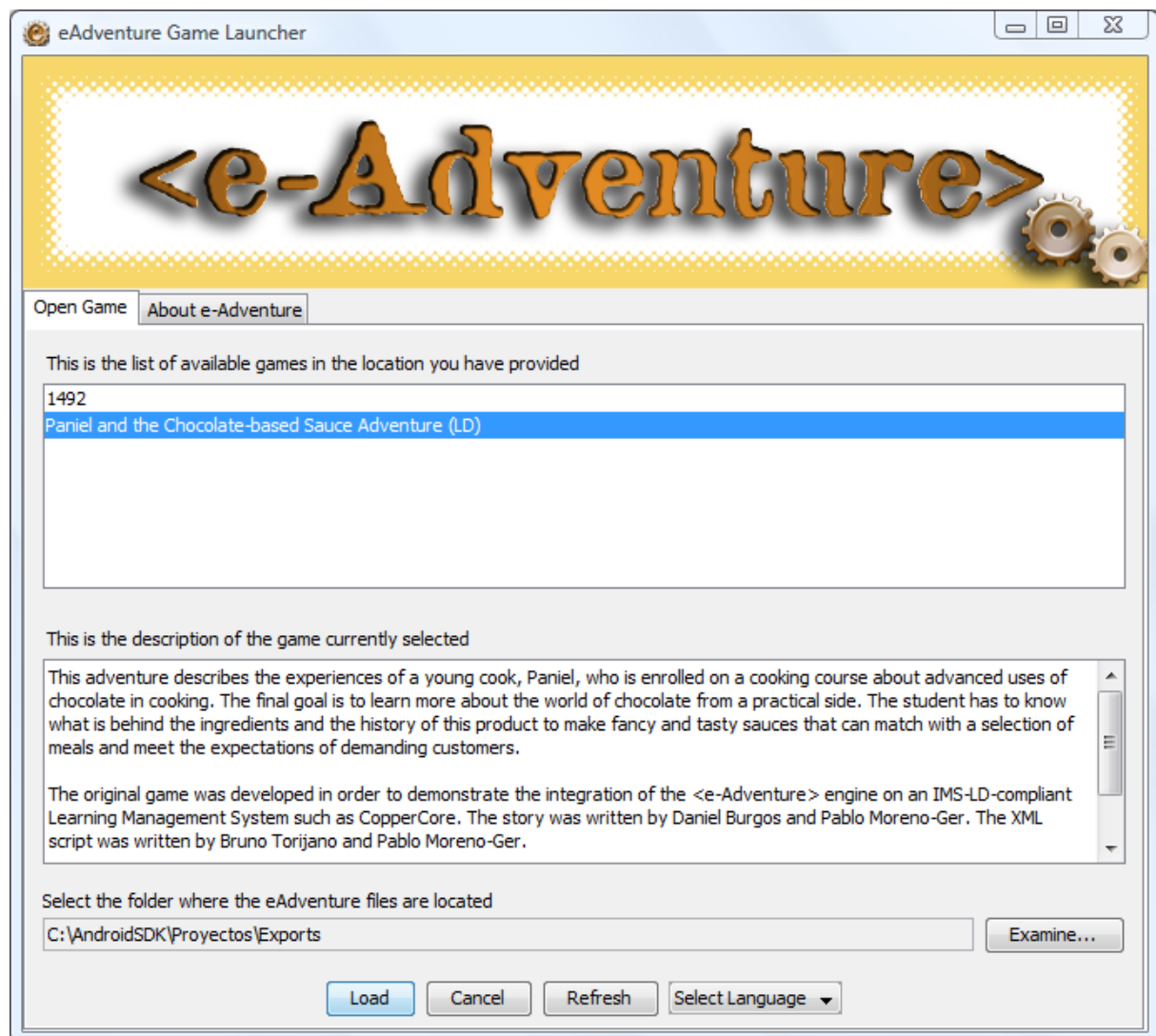


Figure 4. The <e-Adventure> engine start dialog.

## 1.4. Interaction in <e-Adventure> games

Although this document is oriented to explaining the features of the <e-Adventure> game editor, it is important that potential users of the platform get an idea of what the games will look like when the engine runs them. In this regard, how interaction is carried out is an important issue.

When the game is launched the player will perceive the scene as it is. Therefore no further menus or head-ups will be displayed. Players can explore the scene with the mouse. When the mouse pointer is over any interactive element the cursor will change and a brief text may also be displayed.



**Figure 5. Example of a contextual interaction menu in an <e-Adventure> game**

Some elements (e.g. exits, see section 2.2.3) can be interacted by clicking the left button of the mouse. To interact with other elements the player must right-click (or equivalent behavior in Mac systems) on it, leave the mouse left button pressed for a couple of seconds, or make a double click (only in Windows systems). Then a contextual menu with different interaction options will appear. Press on any of the options of this menu to trigger the interaction.

There is a structure called “**Inventory**” where all the objects collected by the player during the game will be stored. This inventory can be accessed by moving the mouse cursor towards the bottom or the top of the scene, although this behavior is configurable (see section 4.3.3.3).

To display the game menu, press the ‘Esc’ key.

## 2. My first <e-Adventure> game

Once a project is selected and the editor opened, the editor window will be visible (Figure 7). We will start by studying the options available in the editor. To the left we find a structure made up of collapsible panels used to organize all the elements in the game, as well as a handy set of edition tools (Figure 6). These tools are: undo, redo, search, back and forward.



Figure 6. Edition tools in <e-Adventure>

When an element is selected in the structure panel, the properties of that element will be editable in the right side panel. This way, the structure panel is used to access the different elements, modify them, delete them and create new ones. To perform some of these operations, a common set of buttons is used to: add new elements (green plus icon), delete elements (red minus icon) and duplicate elements (green square icon). These same icons are used in other contexts in the editor.

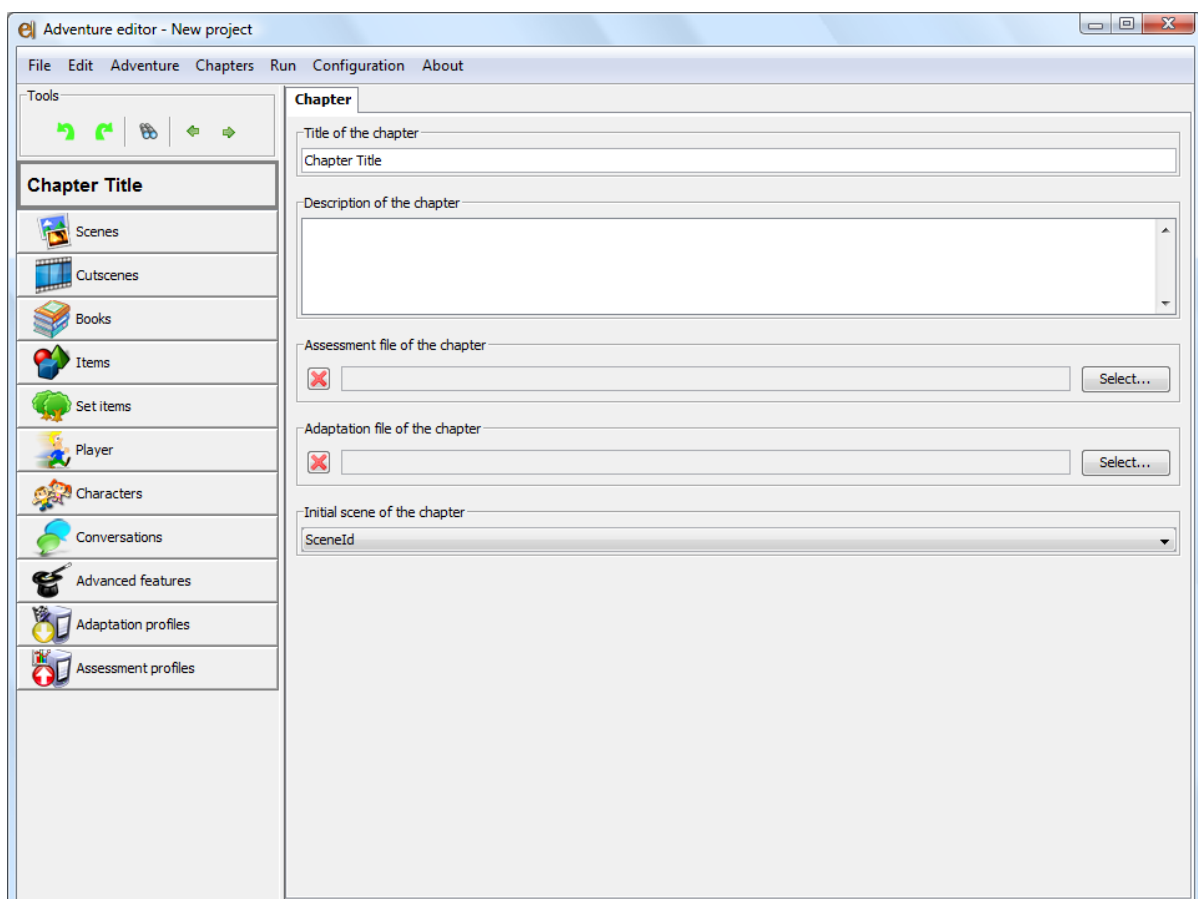


Figure 7. Initial view of the <e-Adventure> editor, once a new project is created or an existing one opened

Now we will proceed to study each element in the structure of an <e-Adventure> game.

## 2.1. Chapters

<e-Adventure> games can be divided into chapters. This allows for games to be fragmented into smaller parts that are easier to manage, design, edit and maintain. Each chapter can be viewed as a totally independent mini-game, or an act in a theatre play. Elements on a chapter are only available within the chapter and not from other chapters in the game.

The editor can only edit one chapter at a time. In the “Chapter” menu (found in the menu-bar) has options to create, import, delete and organize the chapters in the game, as well as to allow for the modification of the current chapter’s name. This menu also has options to modify the *flags* and *variables* available in the chapter, but this will be detailed later on this guide.

Once a chapter is selected, the title, the description and the first (or initial) scene of the chapter can be modified. Besides, the active evaluation and adaptation profiles can be selected, as shown in Figure 7, but these features will be left for the “Evaluation and adaptation” section of this document.

### 2.1.1 Importing Chapters

This feature aims to be a first approach to introduce collaborative edition support in <e-Adventure>. In this manner different authors can distribute the game in chapters, implement them separately, and then merge them into a single game.

Full chapters can be added from other existing <e-Adventure> games. We only need to go to Chapters menu and select Import chapter option.

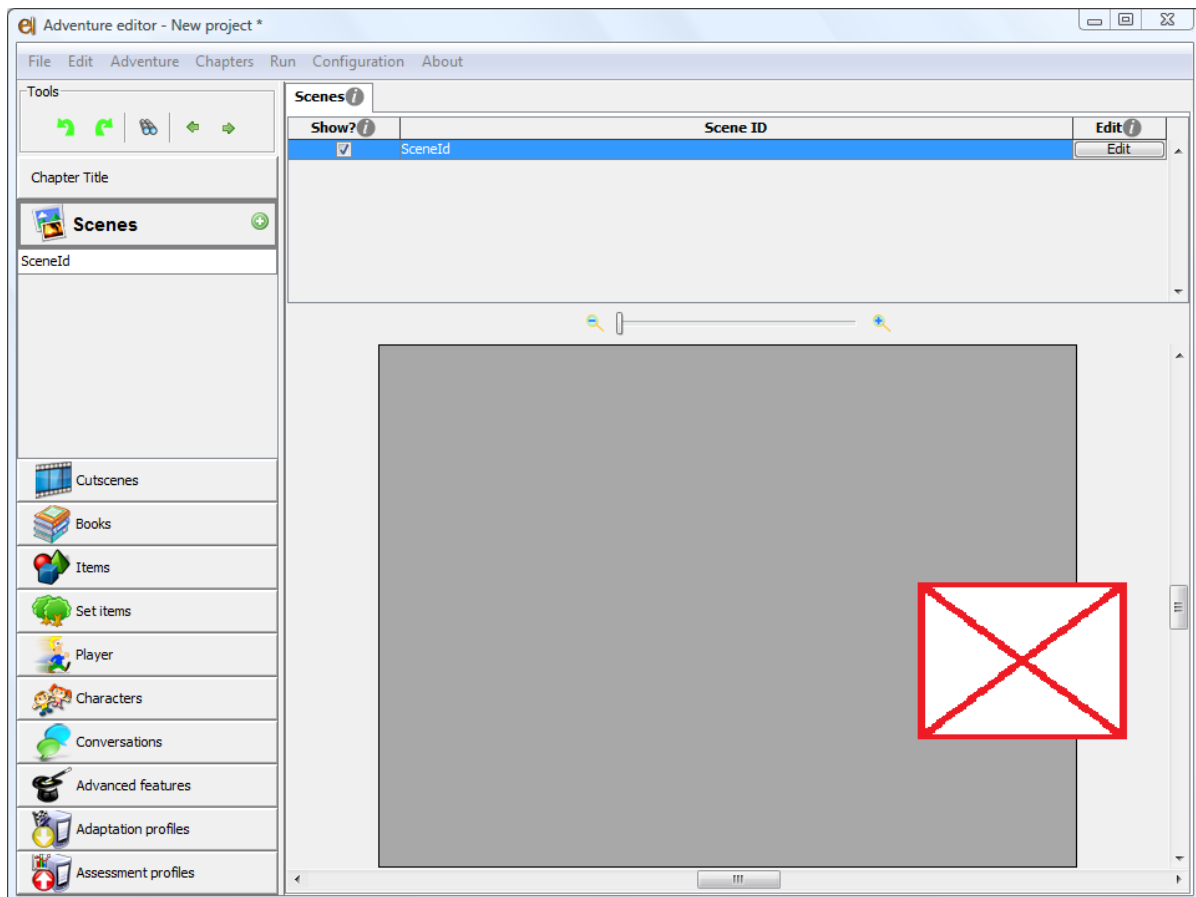
We must navigate through the folders in the pop-up dialog until the desired <e-Adventure> game is found. Then, we select XML file defining the chapter to be imported. Usually, chapter file names are chapterN.xml.

## 2.2. Scenes and cut-scenes

<e-Adventure> chapters are organized into *scenes* and *cut-scenes*. On the one hand, *scenes* are the scenarios or places where the action of the game takes place, that is, where the player interacts with the objects and characters in the game, and these are connected to other scenes by *exits*. On the other hand, *cut-scenes* are used to increase the educational value of games and can be either *Slidescenes* or *Videoscenes*. The first kind is made up of slides or images that are shown using the full screen. The other kind, the *Videoscenes*, display a video using the full screen too.

### 2.2.1. Adding a new scene

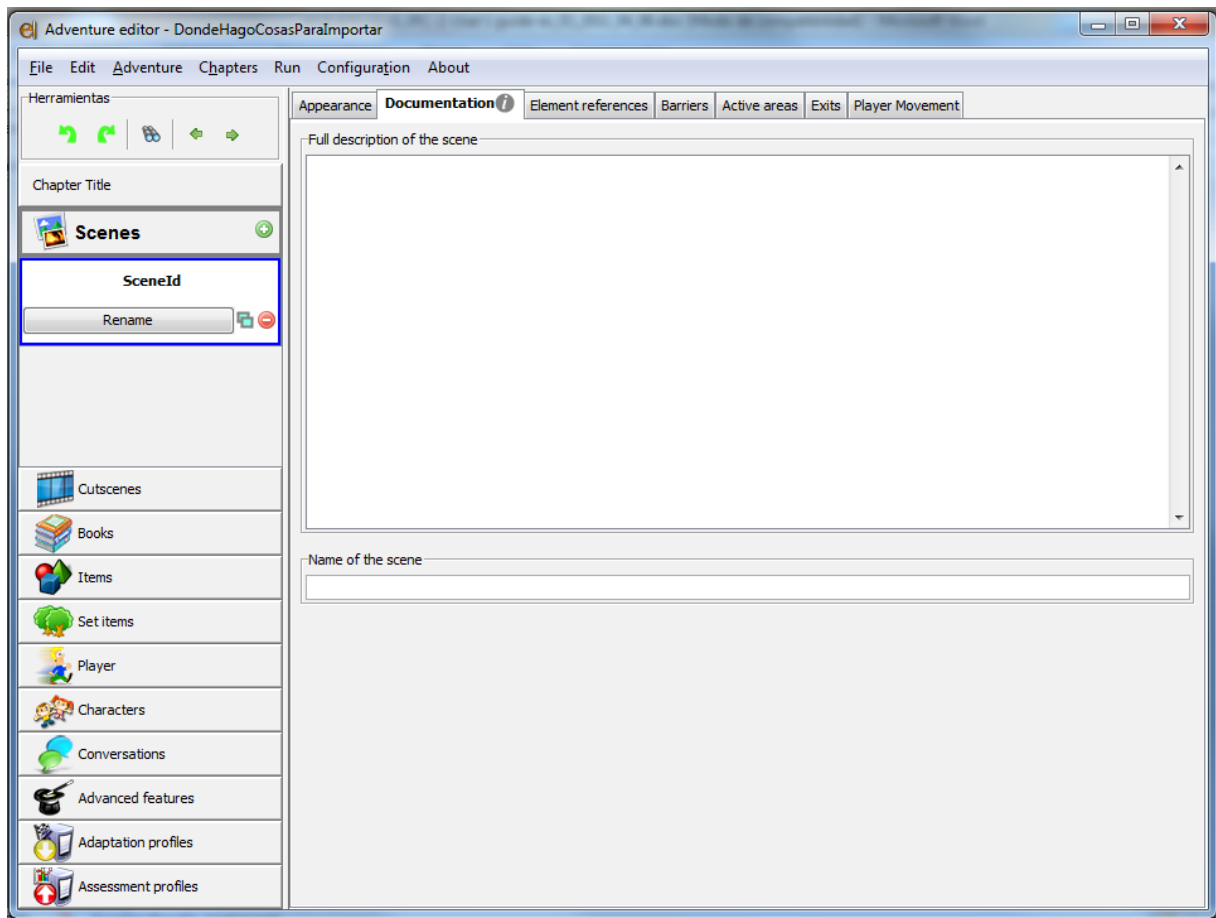
To add a new scene, the “Scene” or “Cut-scene” element in the structure panel must be selected, depending on the kind of scene that is to be added (Figure 8). When the kind of element is selected, the list with all the elements of that kind is displayed. Besides, a button to add a new element (+) appears to the right. Clicking on the button adds a new element with a default identifier (from now on, id). **The id can be modified by clicking in the “Rename” button, which appears when the element is selected, and must be unique, only contain letters and numbers, and start with a letter. This applies to all ids in <e-Adventure>.**



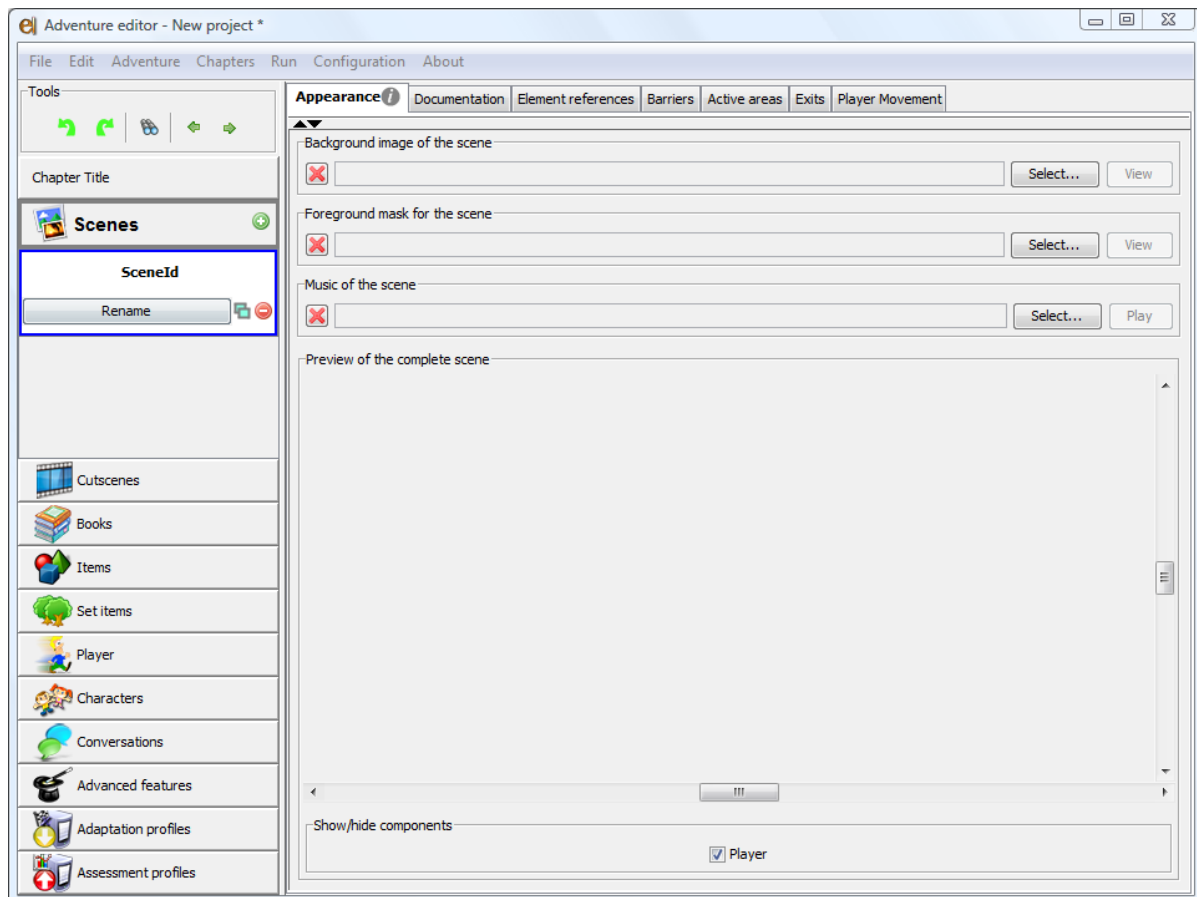
**Figure 8. Scene edition shows a preview of all the scenes in the chapter. By default games include an empty scene**

When a scene is selected in the structure panel, 7 tabs appear in the right side panel (Figure 10). The “Information” tab will appear first and this is common to most <e-Adventure> elements. In this case it allows for the edition of the scene general description and its name. **The element name should not be confused with its id; the former is used in the game while the latter is unique and shown only in the structure of the game.** The name of the element can have white-spaces and use any

character.



**Figure 9.** Scene edition panel, with the “Documentation” tab selected



**Figure 10.** Scene edition panel, with the “Appearance” tab selected

The “Appearance” tab is used to change how the scene is shown (or, more broadly, how it is perceived), and the same is true for any other element in <e-Adventure> that has an “Appearance” tab. For the scene appearance, three resources or assets can be edited but only one (background image) is required while the other two are optional. The contents and meanings of the other tabs will be studied later in this guide.

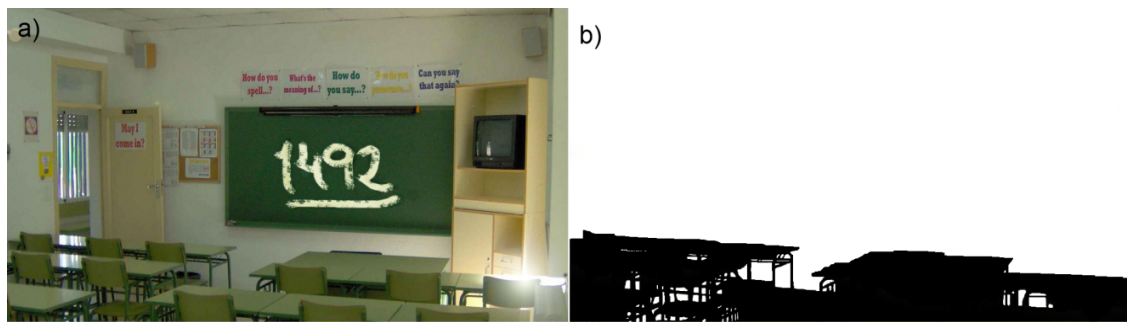
### Background image (required)

This is the most important asset of the scene, given that it represents the actual scene in the game. These images must be at least 800x600 pixels in size. However, although all scenes must be exactly 600 pixels in height, the width of the scene can be bigger. This way, in third person games, the scene will move along with the player’s avatar when it reaches the border of the screen. In first person adventure, a cursor (button) will appear to each side of the screen allowing the user to move the background as needed during the game. **Images for the background, as for any other asset in <e-Adventure> can be: PNG, JPEG (or JPG), BMP, WBMP or non-animated GIF.**

### Foreground mask (optional)

The foreground mask is a black and white image that identifies which parts of the background image should be drawn in front of the objects, characters and the player and which parts should be drawn behind. Figure 11 shows a background image and its corresponding foreground mask.





**Figure 11. Background image (a) and mask (b).** The back parts of the mask mark the parts of the background image that will be painted in the front of other elements and the white parts those that will be painted behind

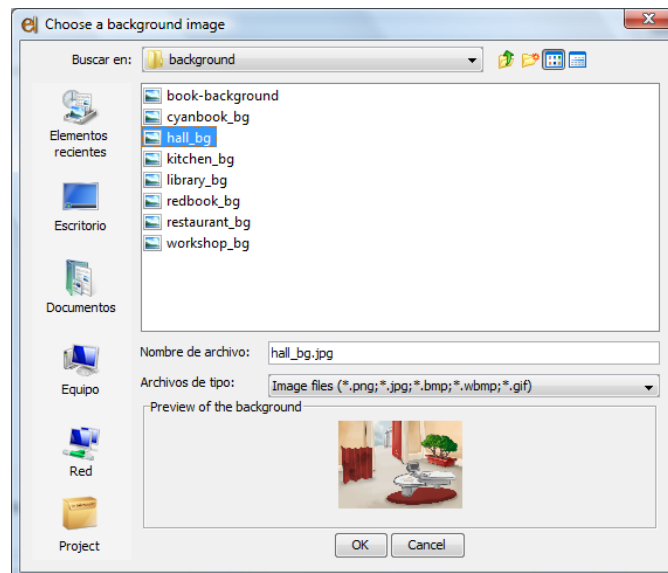
Foreground masks are obsolete and their use is not encouraged, as they are rather complex to use and present many limitations. This behavior can be reproduced by the use of *set items* as explained later in this guide, which at the same time provide a greater degree of flexibility.

### Background music (optional)

This is a music track that will play in a loop while the scene is on screen. It supports both MP3 and MIDI audio.

### 2.2.2. EXAMPLE: Defining the assets of a scene

To add a background to a new scene, we start out by clicking the “Select” button next to the asset. Clicking this will make a dialog to select an image, like the one in Figure 12, visible.



**Figure 12. Asset selection dialog for the scene background**

This kind of screens (the dialog) is used to select a valid resource for the file system (an image in this case). It must be noted that background images must be at least 800x600 as previously established. The lower part of the dialog shows a preview of the selected image. By clicking OK, the image asset will be automatically added to the project.



Besides, in this dialog a button is preset which makes it possible to have direct access to the resources of the same type already in the project. This button is marked as “Project” and appears on the lower left side of the window.

Once the image has been added the foreground mask can be selected. As shown in the previews image, the background has the table and the computer in the front. When a character or the player’s avatar is there, those elements should be draw in front. This is done by choosing a foreground image created specifically for this background as shown in Figure 13.

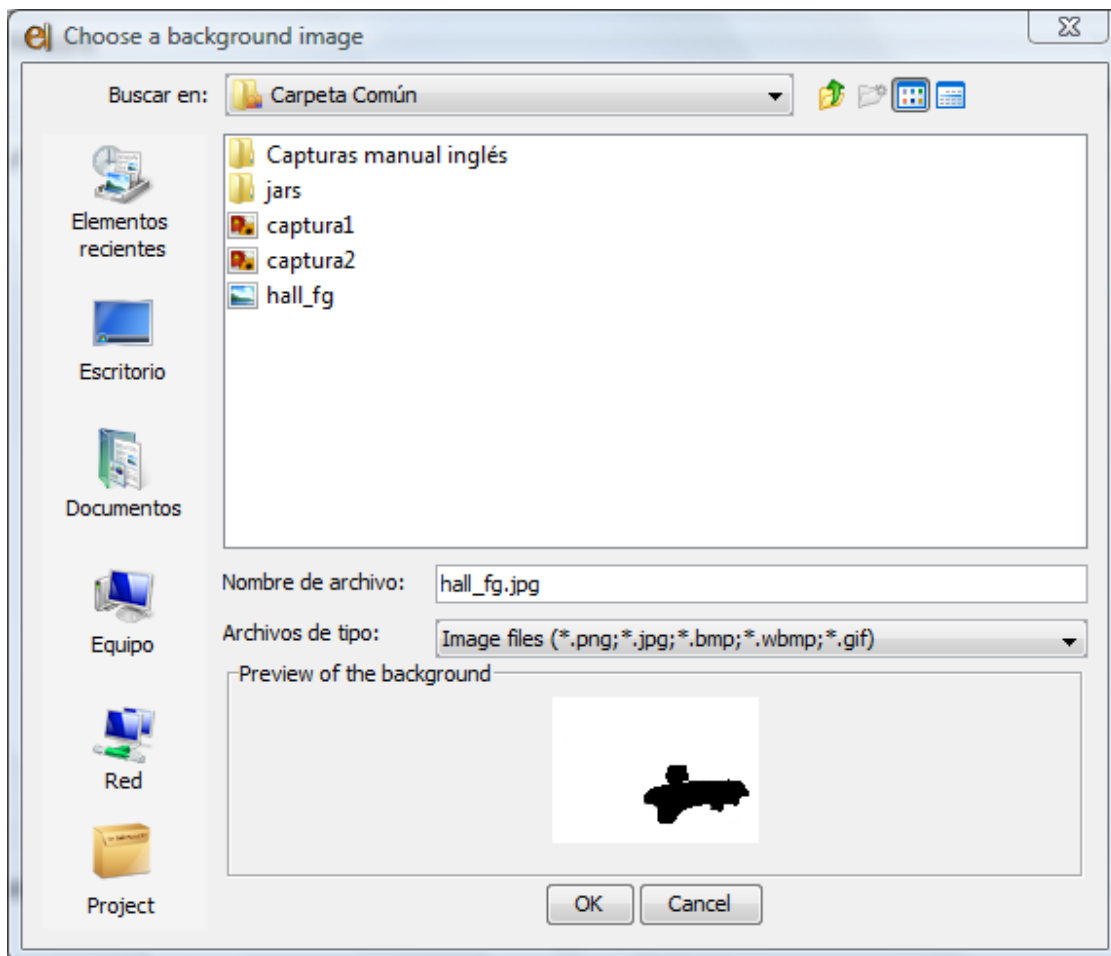
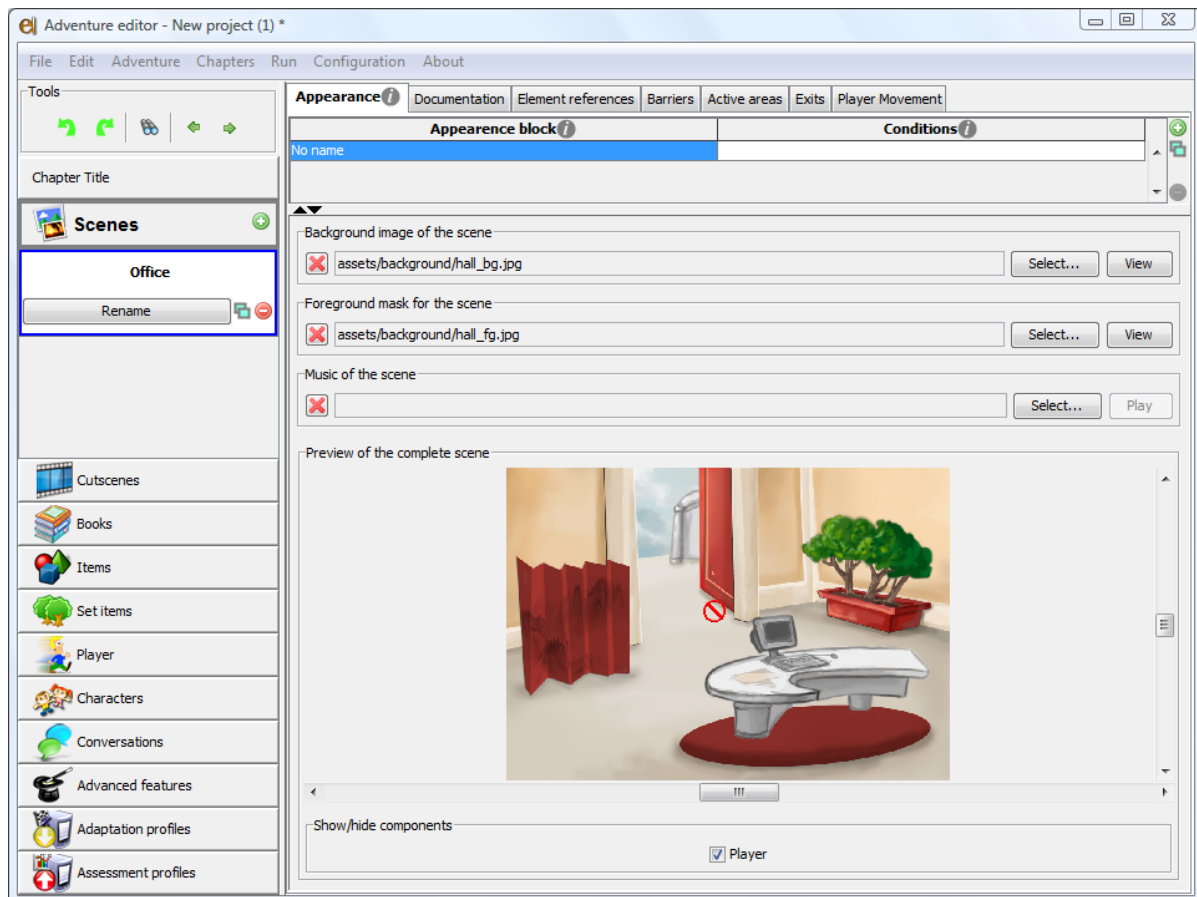


Figure 13. Foreground mask selection dialog

After these steps are completed we can see the final result in Figure 14. It should be noted that red crossed-out circle in the image appears because we are editing a third person game with no assets defined for the player avatar (which would otherwise appear there).



**Figure 14.** Scene edition panel in the “Appearance” tab, after assets were selected for both the background and the foreground mask

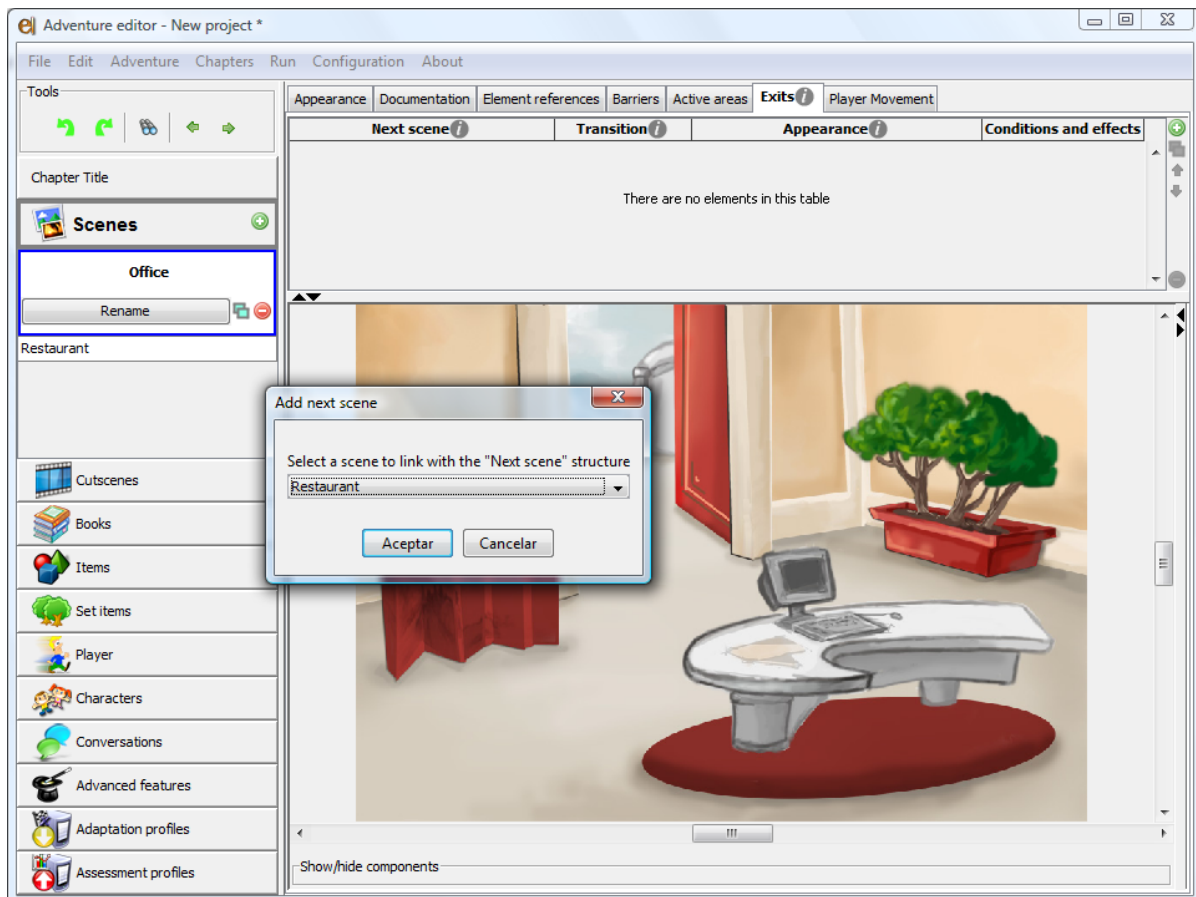
Other elements can also be defined in a scene, such as *exits*, *active areas*, *barriers* and *trajectories*, and *references* (to *characters*, *items* and *set items*), each in a different tab. Later we will show how to add all these different elements, but we will start by adding *exits* to the scene.

### 2.2.3. Connecting scenes: adding an exit

If we want to connect different scenes, creating a heterogeneous world, we have to add exits to the scenes in the game. Without exits, the game would mostly take place in the same scene. Exits can be thought of as invisible rectangles which, when the player click on them, this action results in a change of scene (only after the avatar reaches them, in the case of third person games).

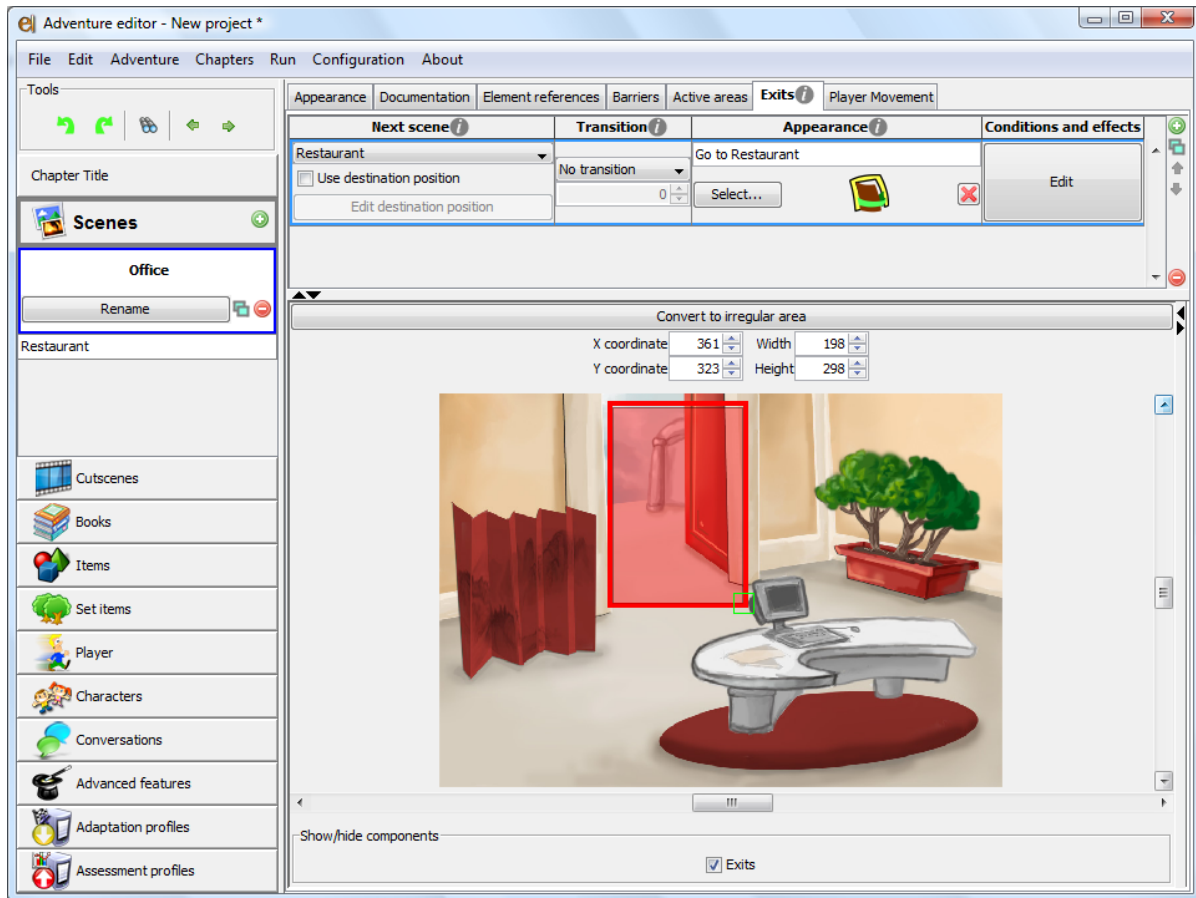
To better understand this concept, let’s suppose we have two scenes, the one we created in the last example (2.2.2) that we called “Office” and a new one called “Restaurant”. As shown in Figure 14, the office had an open door near the center, which we can use to simulate an exit in that area that makes the player go from the office to the restaurant.

The first step is to select the “Exits” tab for the “Office” scene. This shows in the right side panel a list of exists and a preview of the scene. To the right of the table, a + button can be used to add a new exit. When we do, a pop-up dialog will ask where the exit leads to as show in Figure 15. In this case, we choose “Restaurant”.



**Figure 15. Scene edition panel, selecting the next scene when adding an exit**

After selecting a destination scene, a new element is added to the list of exits and the new exit, now selected in the list, is shown in the preview panel of the scene. The highlight red area in the preview corresponds to the selected exit, and it can be dragged anywhere on the scene; and the exit can be re-sized by clicking and dragging the green element in the lower left corner (Figure 16). It must be noted, however, that during the game the exit will be completely invisible.

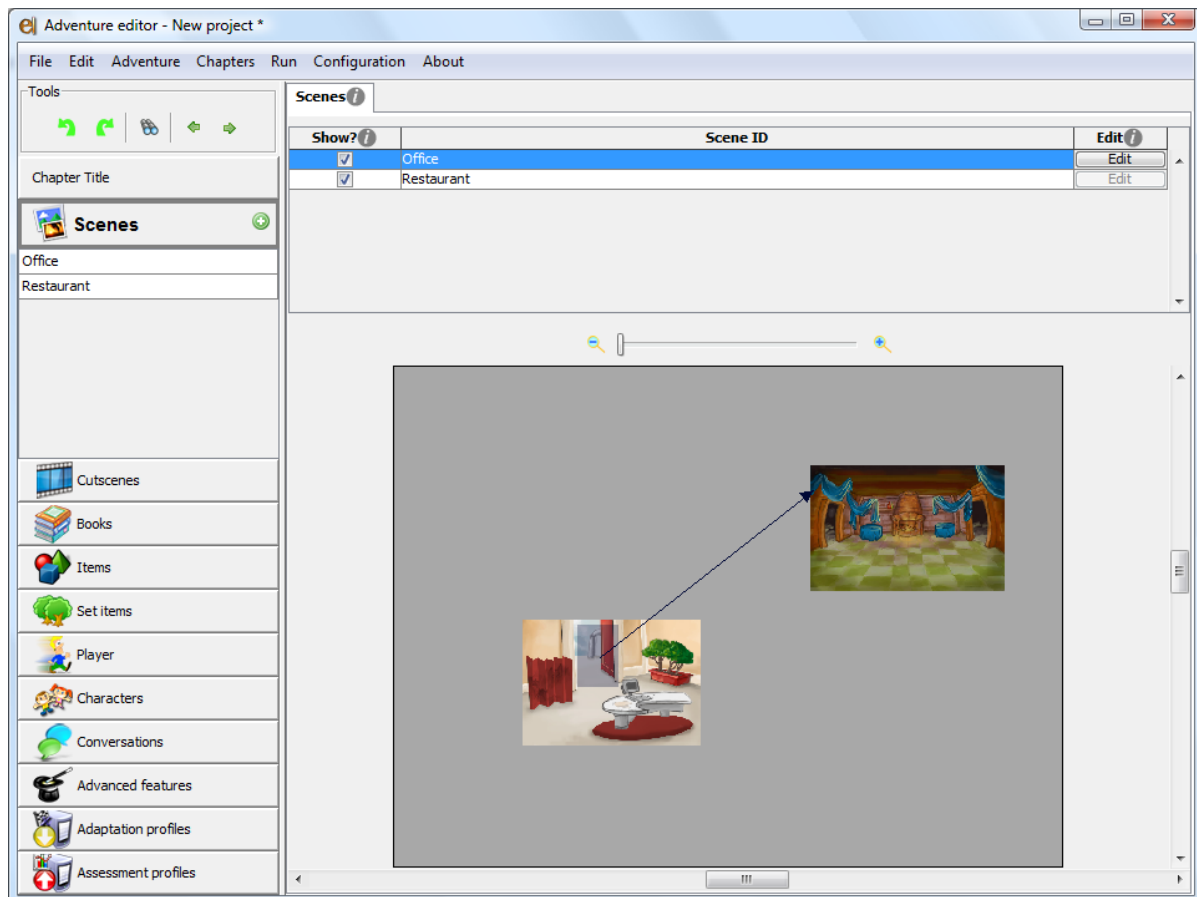


**Figure 16. Scene edition panel while adding a new exit. The exit can be dragged and re-sized as needed**

The name (which will be displayed when the cursor is over the exit) and an image for the cursor when placed over the exit can be modified in the “Appearance” column of the table. This way, in our example, when the player moves the mouse over the exit its name (“Go to restaurant”) will be shown over the cursor, and when the player clicks the scene will change.

Exits (as well as other “rectangular areas”) can also have complex shapes. To do this we must click the “Convert to irregular area” button. This will be detailed later.

If we now go back to the “Scenes” element in the left structure panel we will see the two scenes in our game and the exit that links them represented as an arrow (Figure 17). This helps when editing large games to see every scene in a game and every exit that links it to another.

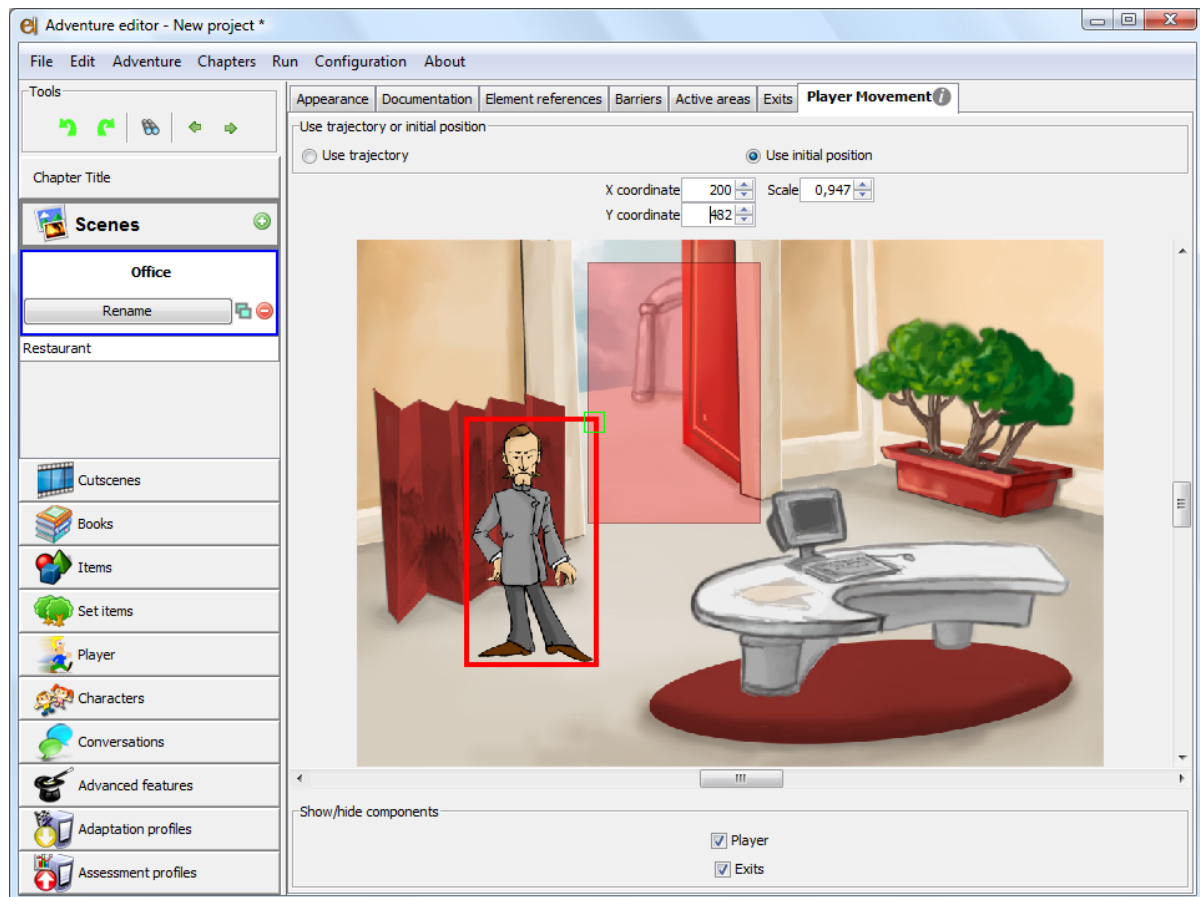


**Figure 17. Scene edition panel when previewing the scenes in a chapter. This shows all scenes in the game and the relation between scenes created as a consequence of exits from one to another**

### 2.2.4. Scene initial position

Before being able to test the game, the player assets must be defined, but this is studied in detail in section 2.5.5. However, it is possible to do that the player would still appear in an apparently random position. This can be changed by defining an initial position, which we will do for the office scene. To do this, we choose the “Player movement” tab while the “Office” scene is selected in the structure panel.

This tab will show a preview of the scene with every element defined in it (Figure 18). In this case, the “Use initial position” choice is selected. The player can be dragged to the desired position in the scene. The player scale in the scene can also be modified by dragging the green element in the upper left corner. At the bottom of the preview panel there are check-boxes for the different kind elements of a scene, deselecting a kind of element will make it disappear from the preview (note that there are only check-boxes for those elements that were defined at least once in the scene). These check-boxes only affect the behavior of the preview and have no consequences in the game-play.



**Figure 18. Scene edition panel while on the “Player movement” tab**

Finally, the entire configuration regarding the placement of the player avatar in the scene is only available in third person games. Besides, an advanced mechanism to define the position of the player in the scene (as well as how to move through it) based on trajectories is also available and studied in section 3.8 of this guide.

## 2.3. Items

New items (or objects) with which the player can interact can be added to <e-Adventure> games. This section describes how to add these items in a game.

### 2.3.1. Adding a new item

Creating a new item is very similar to creating a new scene. It starts out by selecting the “Items” element in the structure panel and click on the add button (+). Click the button will add a new item with a unique default id. This id has the same conditions, and is edited in the same way, as the id for the scene as seen in section 2.2.1.

When the item is selected in the structure panel, the edition process is similar to the one followed for scenes (Figure 19). In the case of objects, the “Appearance” tab only requires that will be used when the item appears in the scene and an icon that represents the appearance of the item, will also appear in the

inventory<sup>1</sup>. Although there are no fixed restrictions for the image of the object, the icon must be 80x48 pixels and can be generated automatically from the image using the “Create icon from image” button.

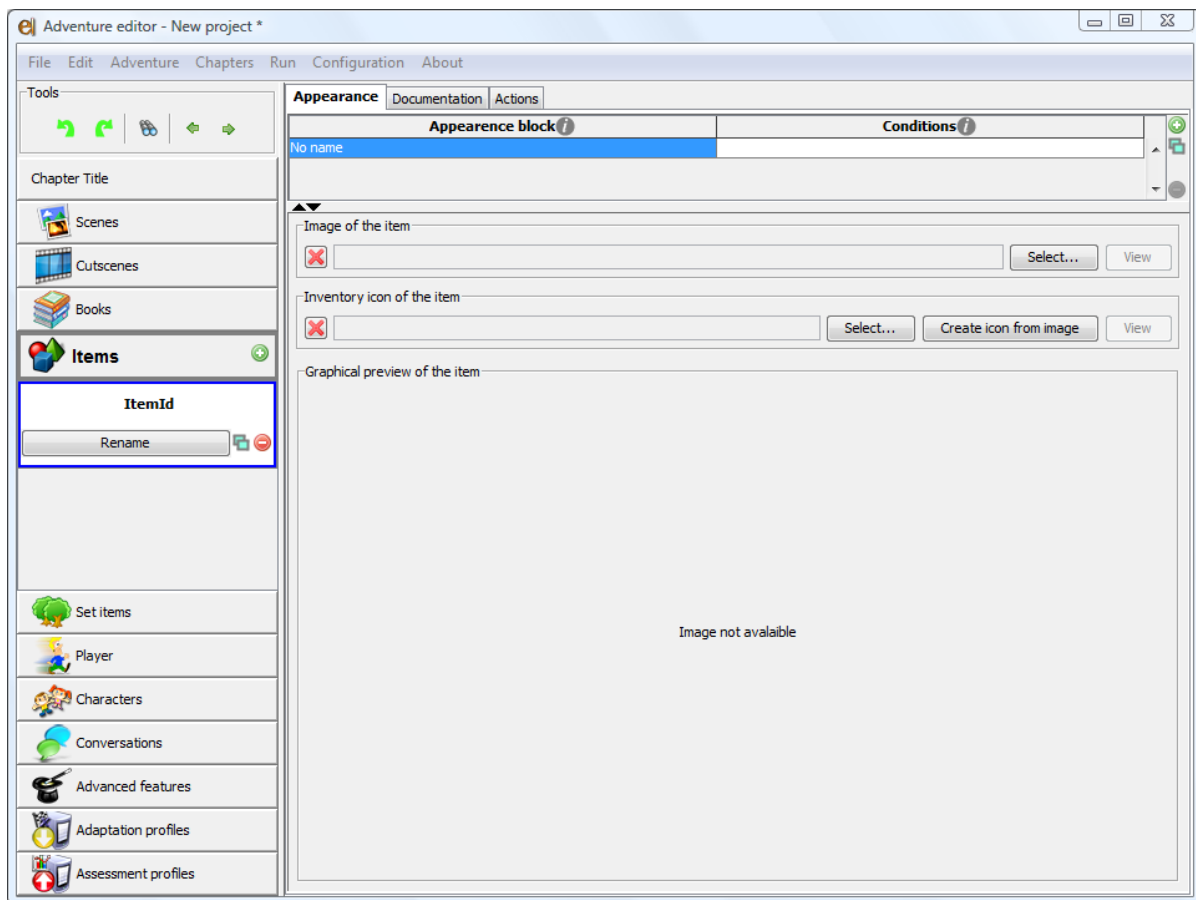
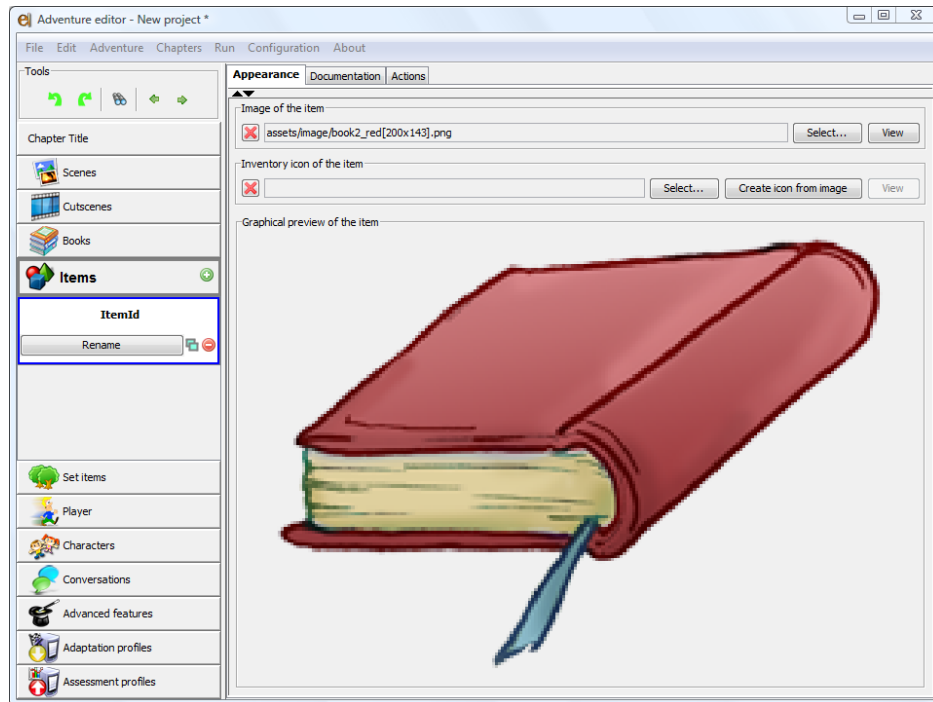


Figure 19. Item edition panel in the “Appearance” tab with no selected asset

For example, if we can choose an image for a new object we see the preview of the object in the lower part of the right side panel (Figure 20). Remember that the id of the element can be changed using the “Rename button” on the left side panel, just as done for the scene in section 2.2.1.

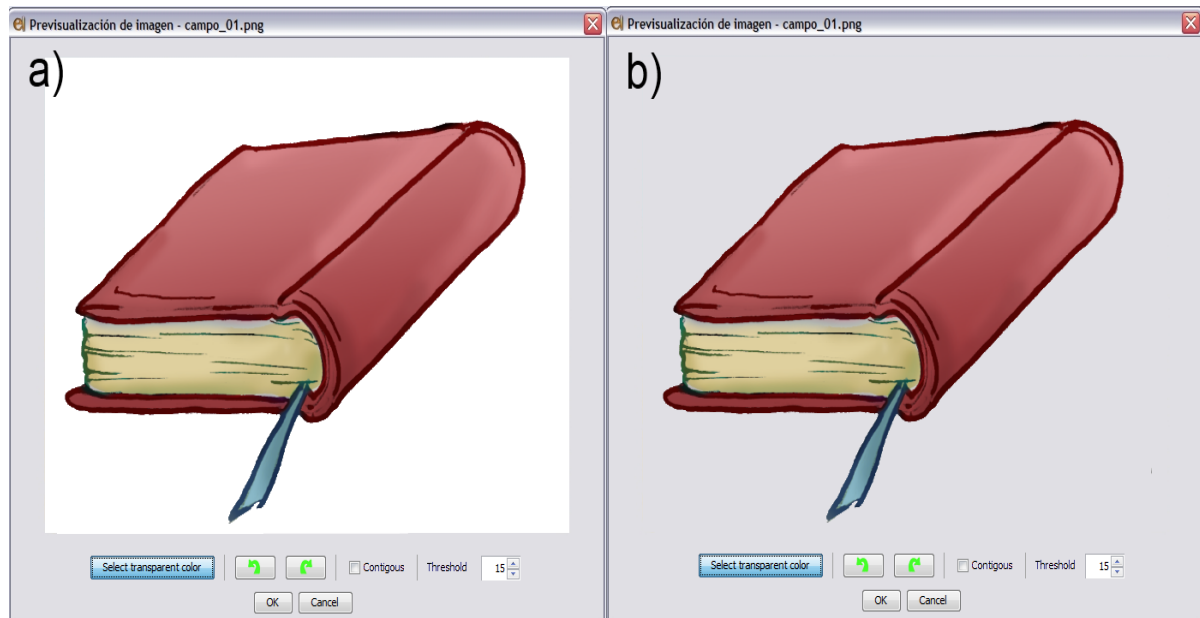
<sup>1</sup>As in traditional graphic adventure games, the player has an inventory where all items are placed once grabbed from the scene. This items can be used later in the game just as if they where in the scene.





**Figure 20. Item edition panel in the “Appearance” tab, when the item has an image attached.**

Moreover, after an image is selected for the item, the “View” button to the left is enabled. This option is available for all image assets. If we click this button a dialog will appear where we can preview the selected image, and it allows from some basic manipulation (Figure 21).



**Figure 21. View/Edit image dialog**

In this view dialog the background of an image can be removed and made invisible in the game. To do this we first click the “Select transparent color” button and then click somewhere in the background of the image. After this, two more settings are available:



- **Threshold:** The threshold gives a measure of how similar the colors should be to the selected one to be made transparent. The higher the threshold, the higher the number of pixels that will be made invisible.
- **Continuous:** If this option is selected, only pixels adjacent to the one selected will be made invisible.

### 2.3.2. Interacting with items: actions

Following the previous explanation it is possible to create a basic object. However, that isn't everything that can be done with an item. If we select the "Actions" tab for an item (for instance, for the recently created "Book" item) a list of actions will be displayed in the right side panel. When an action is selected, its details appear in the panel below the list. A new action can be added by clicking the add button (+).

Besides, when the player left-clicks any item (and this is also true for characters) in <e-Adventure> it results in the player character "reading" its description. This description can be edited in the "Documentation" tab. If the game is in first person, the description will appear near the center of the screen.

The actions available for items (Figure 22) are:

#### **Examine**

When an object is examined, by default, the detailed description of the object is shown in the screen. Even when not added to the item, the "Examine" action will be available to the player. However, the "Examine" action can be reconfigured and this is achieved by adding the action to the list and creating a new behavior for it.

#### **Grab**

Once the "Grab" action is defined for an item the player will be able to grab the object from the scene and place it in the inventory. This item can later be used or combined with another object, as well as given to another character. This default behavior can be overridden by the use of a special effect later described.

#### **Use**

This action defines that the item has a use in itself, that those not requires another character or object. This use can be different for every object and will have unique consequences that must be explicitly defined.

#### **Use with**

This action, although similar to "Use", requires the intervention of another item. When our item is used with the "target" item, the effects of this action will be evident. However, using the item with an item different from the target will have no consequences.

## Custom

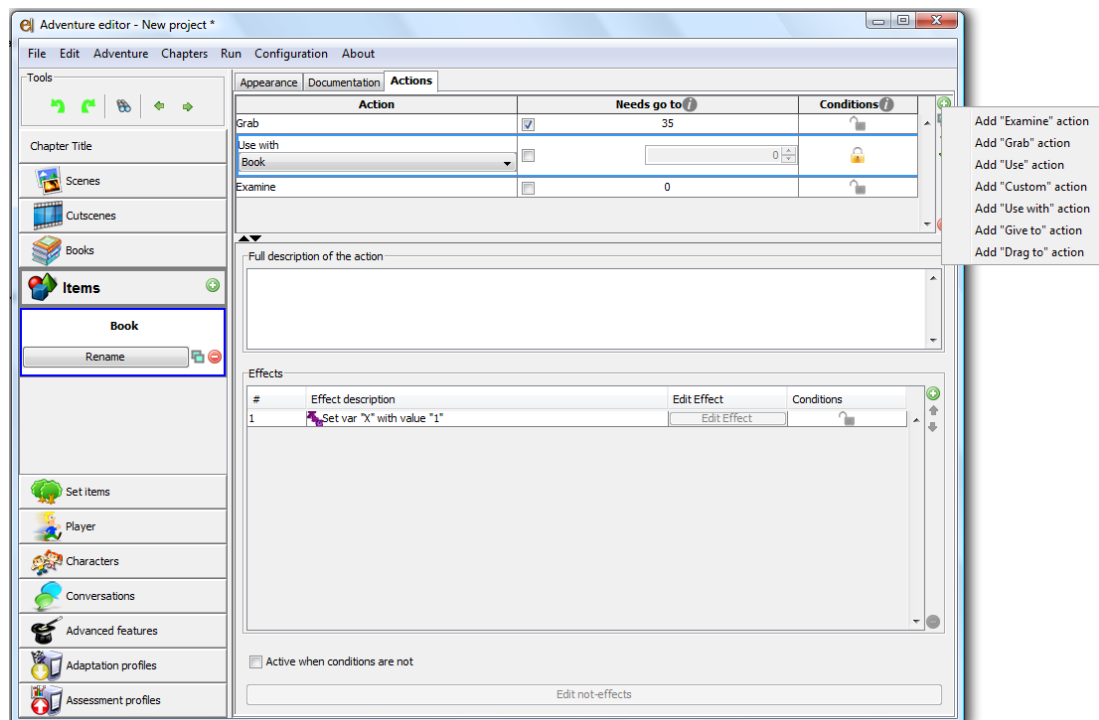
Custom actions allow for a greater number of details to be edited, such as the name or icon of the action and other features of its behavior. These actions are studied in detail among the advanced features of the platform (section 3.10).

## Give to

“Give to” behaves just like the “Use with” action, but the target of this action has to be a character. Besides, this action is only available for items once they have been grabbed and placed in the inventory. By default, after an element is given to a character, it will disappear from the inventory and the other consequences made evident.

## Drag to

This action, defined just like “Use with”, only differs from the former in the way it is used by the player during the game. The *effects* of this action will be triggered when the item is dragged onto the target item or character.



**Figure 22. Item edition panel, “Actions” tab. This tab shows the list of actions and, after click the add button, the different kinds of actions that can be selected**

An infinite number of actions can be defined for the same object. When more than one action of a certain type is defined (for instance, two “Examine” actions) the engine will choose which to use based on the *conditions* (these can be edited in the actions list). The rule followed is that the first action whose conditions are valid is used. This is further explained when the concepts of *conditions* and *effects* are detailed. The actions can be sorted in the list by using the arrows to the right of the list (↓ ↑).

The column named “Needs to reach item”, which is only relevant in third person games, allows for actions to be restricted to when the player’s avatar is able to reach the item in the scene. This allows for

actions such as “Shout to” that can be preformed from anywhere and others (such as “Grab”) that need the player close. If this option is checked, barriers or other obstacles can make it impossible to perform the action.

### 2.3.3. Information

Just like for scenes, the information about items can be modified in the “Information” tab (Figure 23). There are 4 fields that can be modified by the user, and a checkbox:

- Documentation: this field is used to document (i.e. it is not used during the game) the item. Information in this field is stored in the game but only used by the game editor, to provide further information about the item when edited by someone else or at a different time.
- Name: This is the name of the item. This name will be shown to the game user when the mouse cursor is over the item.
- Brief description: This description should be concise and provide an idea of the nature or possible use of the item. The player will be able to see this description by left-clicking the item in a scene.
- Detailed description: This description can be longer, providing additional details about the object. By default, this description is shown to the user when “Examining” the item. However, such default behaviour could be overridden.
- The item returns to its initial position: This checkbox establishes how the item will behave in the case of being dragged. If it is selected, items will return to there original position once the player releases the mouse button.

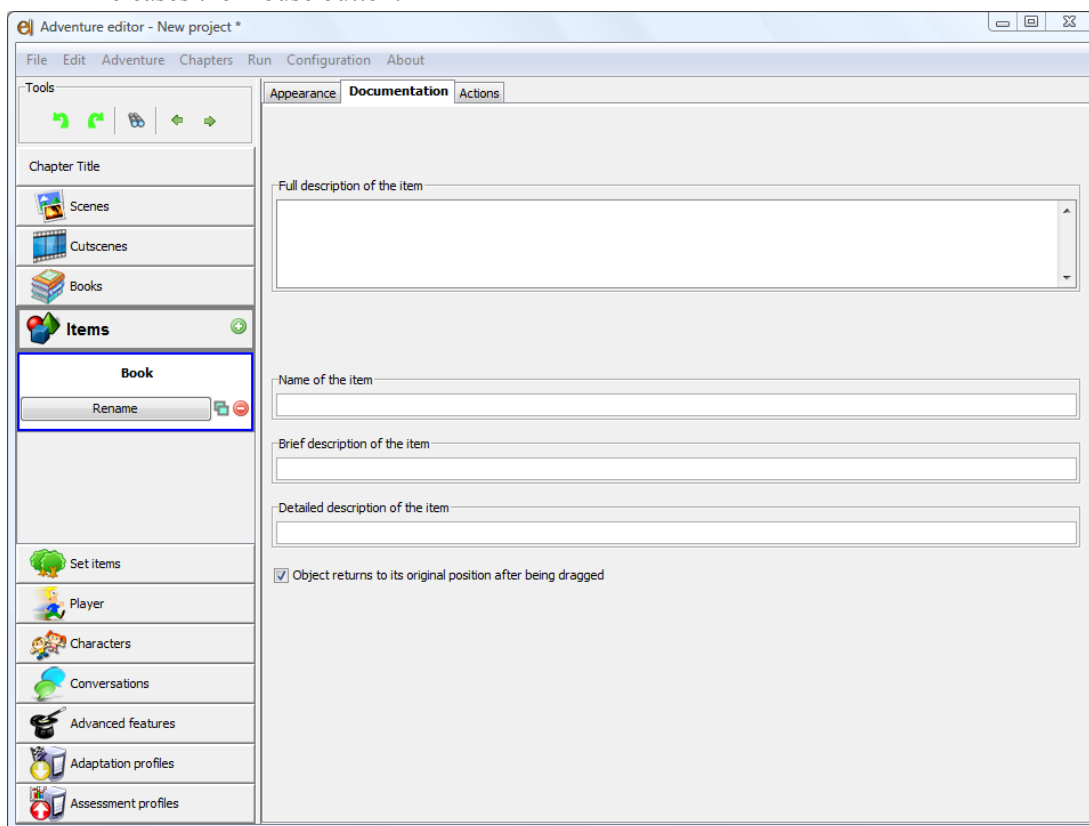


Figure 23. Item edition panel, “Information” tab

### 2.3.4. EXAMPLE: Adding an item to a scene

Once an item has been created, it can be referenced from anywhere in the chapter. For instance, a reference to the item can be added to one or more scenes. This example shows how to add an item to a scene.

The first step is selecting the relevant element from the left panel or structure panel. In this case we will select the scene where we want to place the item and the “Element references” tab of that scene in the right side panel. Now the center panel shows a list of references on top and a preview of the scene at the bottom. Clicking the add button (+) next to the reference list allows us to add a new reference of one of the three element types in <e-Adventure>: items, NPC (Non-Player Characters) and set-items. To add an item to the scene, choose that option. This will prompt a dialog asking for the specific item that we wish to reference by providing a list of elements of the selected type. For this example we will choose the recently created “Book” item y then click the OK button. The item will be added in a default position (Figure 24).

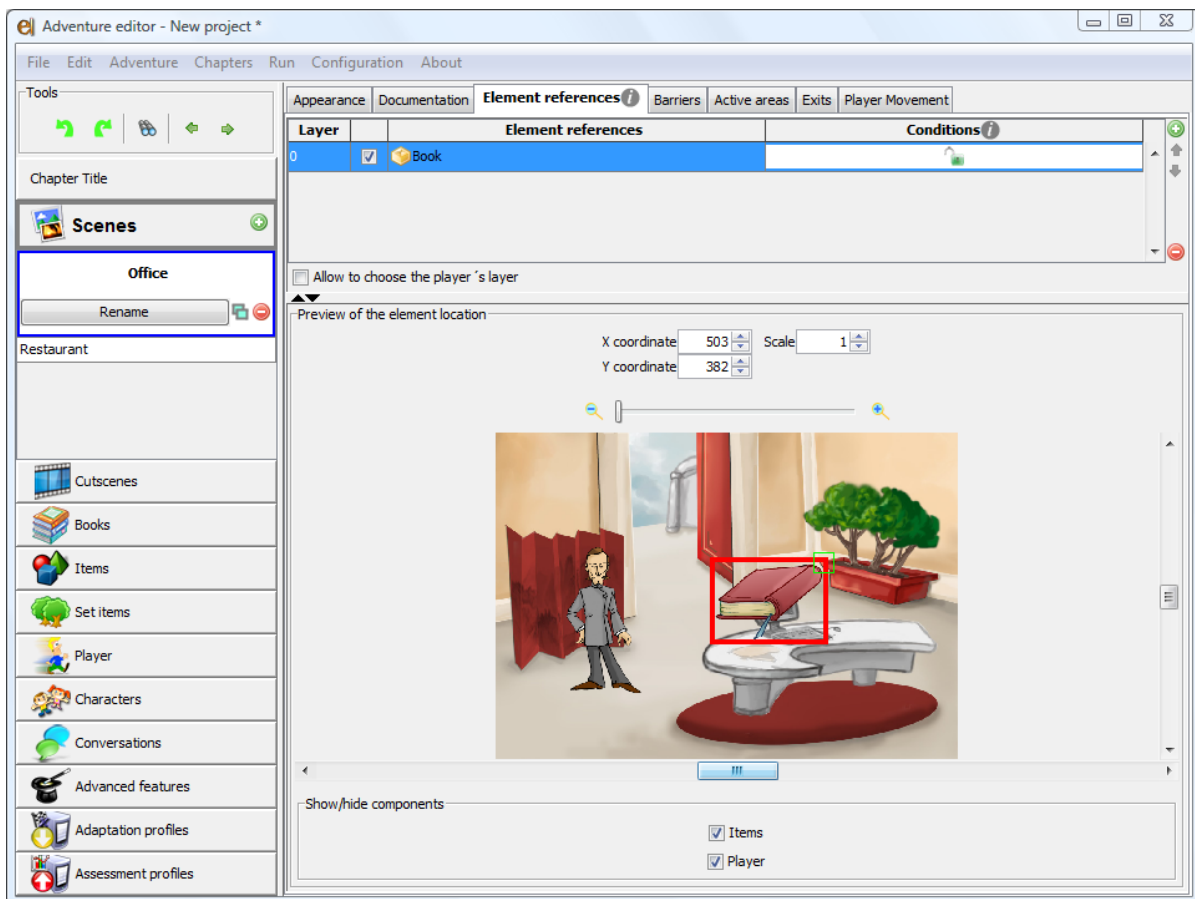


Figure 24. Scene edition panel, “Element references” tab just after adding a new element reference

As most elements in <e-Adventure>, element references can be scaled and moved using drag-and-drop. Changes in the element reference only affect that reference and not the item in itself. Besides, this panel includes fields (just on top of the preview) to modify the position and scale of the references with higher precision. This allows the book to be correctly placed over the table in our example (Figure 25).

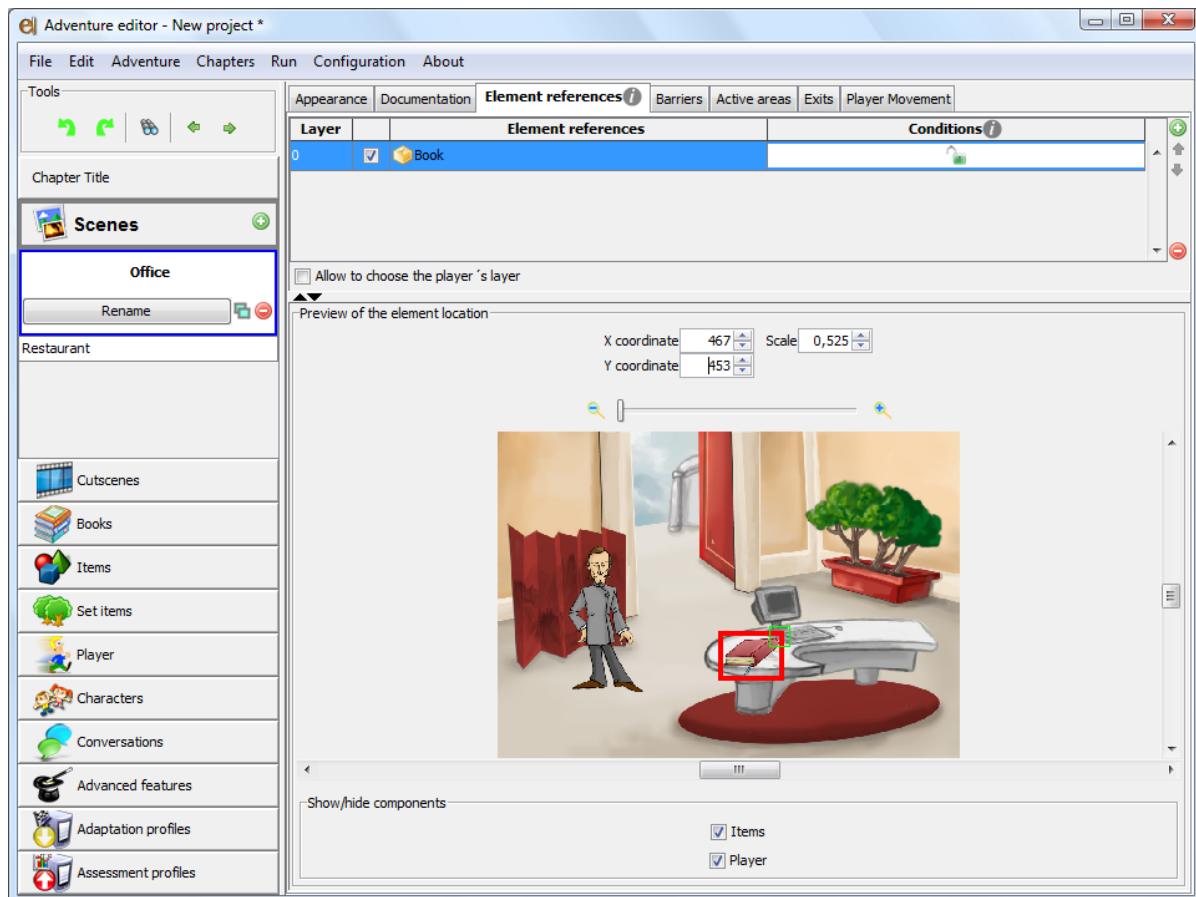


Figure 25. Scene edition panel at the “Element references” tab. After moving and scaling the “Book”, it now sits correctly over the table in the scene

Other settings are available in the reference list:

- Layer: The layer is used to provide a sense of depth. A lower value means that the element is further from the viewer (more information about this can be found in section 3.2).
- Checkbox: This box is used to hide or show the reference in the preview. It has no effect in the game, as seen in section 2.2.4.
- Referenced element: This column shows the actual <e-Adventure> element that is being referenced.
- Condition: This is the condition that establishes when, during the game, will the reference be visible to the player. If the conditions are true, the reference is shown and if they are false the element is hidden. Conditions are edited as any other in <e-Adventure> (see section 3.1).

In scenes with trajectories, this panel also allows for the edition of the “influence area” of an element reference (i.e. the area where the player must be as to be able to interact with the object). This, however, only affects scenes with trajectories and is studied in detail en the “Player movement”, in section 3.8 of this guide.

**Note: Clarification about how “Grab” action Works with the element referents**

The same object can be included in different scenes. As discussed in Section 2.3.2 when the user perform a “Grab” action the game engine removes the object form the scene and adds it to the inventory.

This action affects the whole chapter, that is, the object will not be shown in the chapter even it is referenced in other scenes (more information about in section 3.1.9).

Moreover, when a “Use” and “Grab” actions are defined over the same interactive element the engine will only display the action “Grab” when hiding the action “Use” is not active, it means, when the conditions over “Use” actions are not met.

## 2.4. Set-items

Set-items only provide visual information and cannot be interacted with.

### 2.4.1. Creating a new set-item

Just like for regular items, set-items are added by clicking the add button (+) after selecting “Set items” in the structure panel (Figure 26). The new set-item will have a default id that can be changed by clicking the “rename” button, with the same restrictions as other ids in <e-Adventure> (starts with letter, unique, etc.).

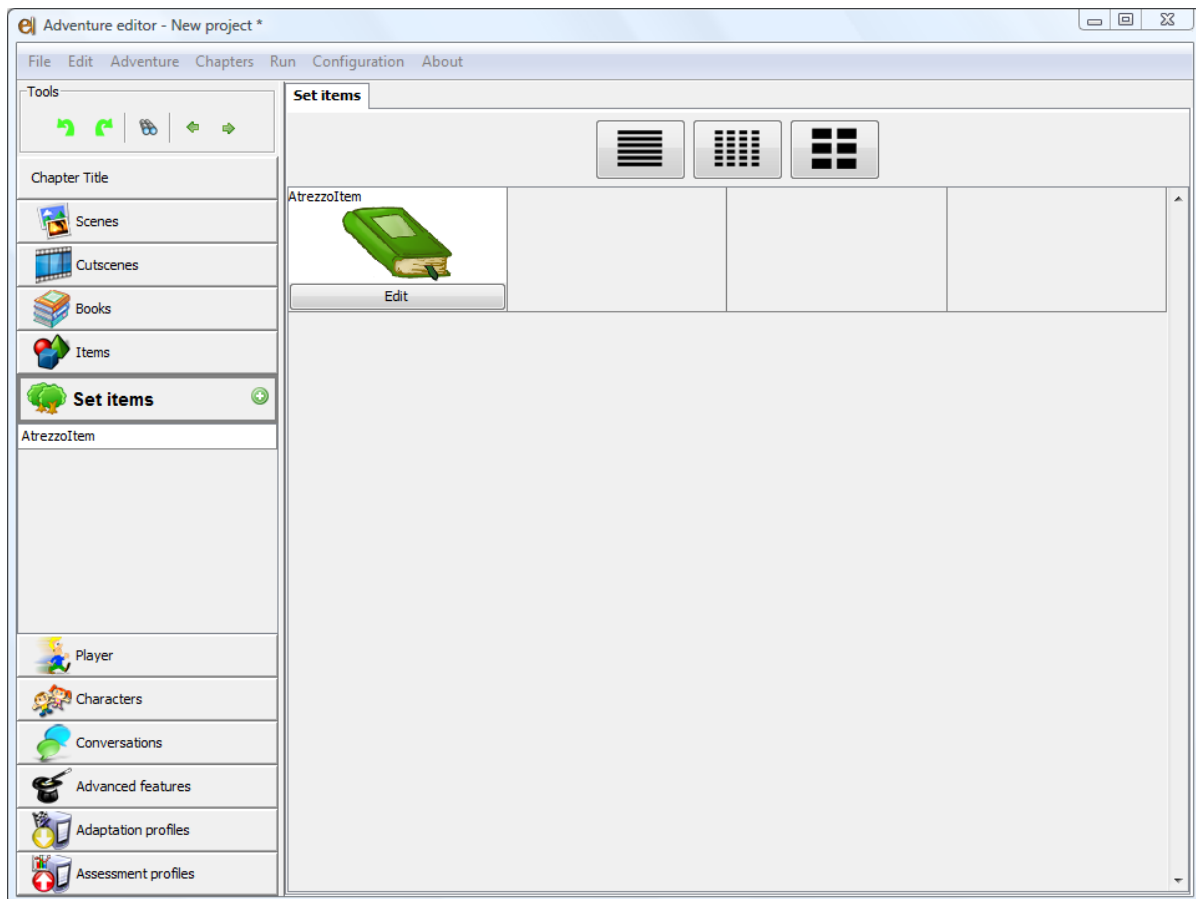


Figure 26. “Set items” general view, with just one set-item in the chapter

The edition panel for set-items is similar to the one for items but with fewer tabs. The “Appearance” tab is used to choose an image for the set-item, but in this case no icon is needed (Figure 27). There are no restrictions to the size the image of the set-item must have.

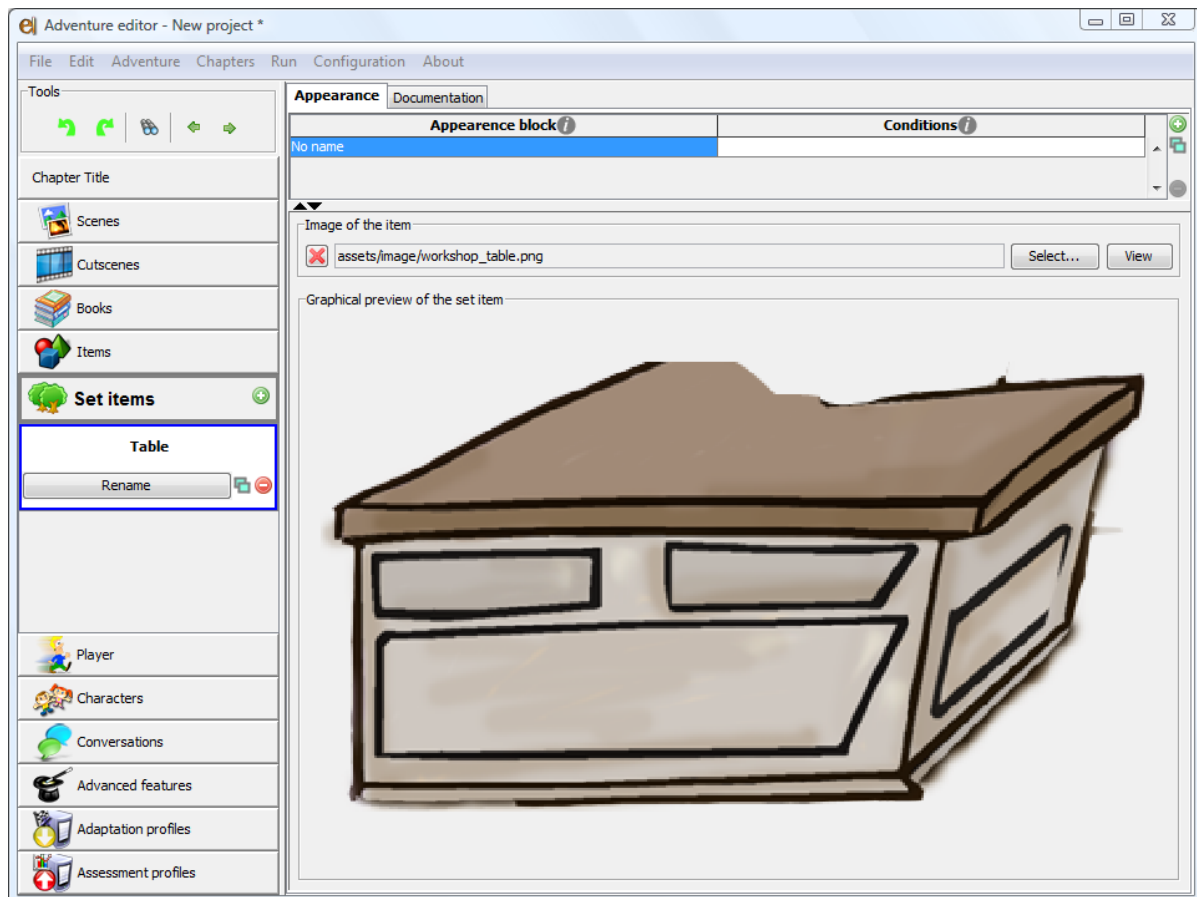


Figure 27. Set-item edition panel in the “Appearance” tab after setting the image

### 2.4.2. Information

Just like for regular items, set-items can be documented in the “Information” tab. There is also a “Name” field, this is not used during the game and is also used for documentation purposes.

### 2.4.3. Adding a set-item to a scene

This process is the same as the one for regular items (see 2.3.4), just that instead of adding an “Item reference” we add a “Set-item reference”. Set-items can be moved and scaled in the same way too.

## 2.5. Characters

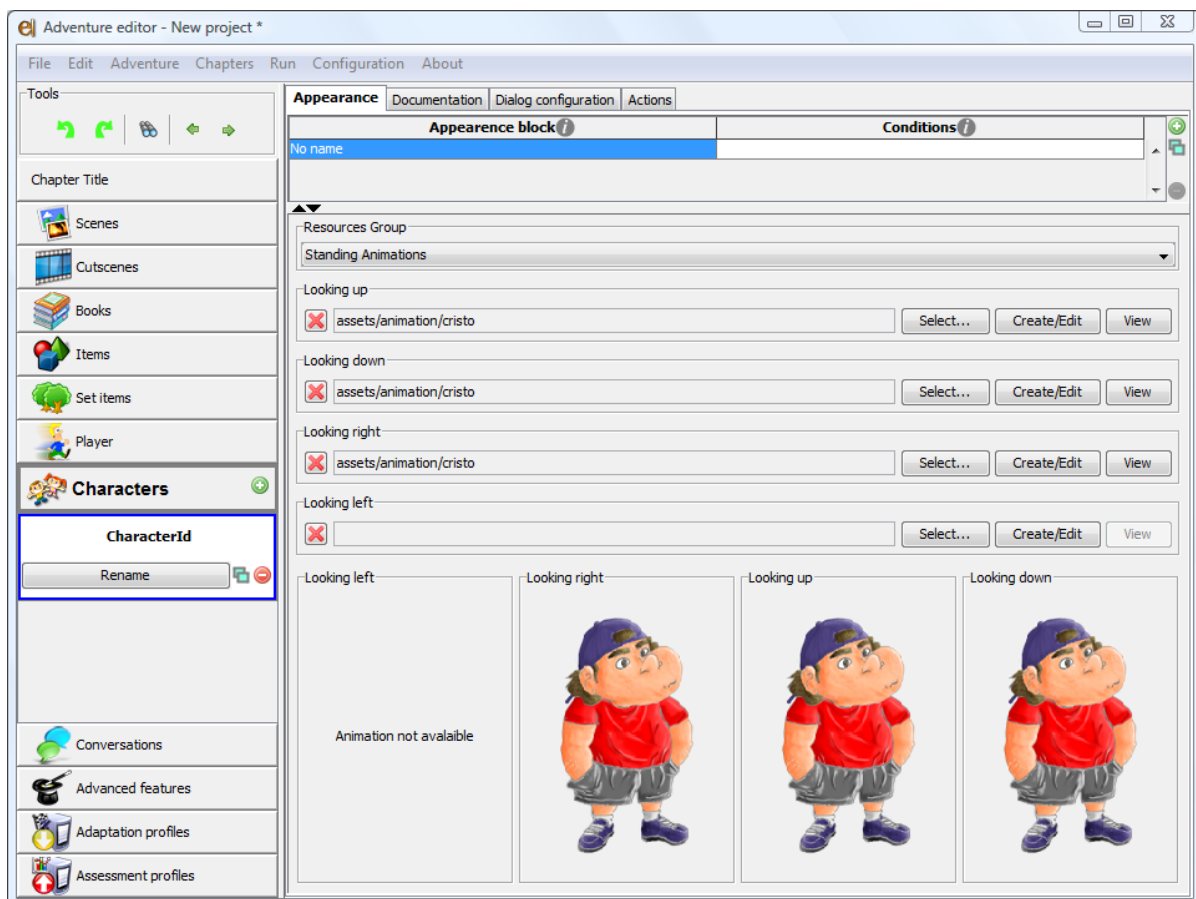
Characters are game elements which, among other things, the player can talk with. Characters (also referred to as NPC or Non-Player Characters) have several differences to the items from the edition point of view. In the first place, characters can be assigned animations, providing them with life like characteristics. Moreover, some actions are different given that characters cannot be grabbed, used or given to, but they can “receive” an item and can be talked to. However, characters can be dragged.



### 2.5.1. Create a new character

New characters are added following a similar procedure to other objects. To do this, see sections 2.3.1 and 2.4.1.

Once the character is added, some difference become apparent. The appearance of this sort of element requires much more resources. This means that different animations must be provided for different situations (talking, grabbing, and walking) and with different orientations (Figure 28).



**Figure 28. Character edition panel in the “Appearance” tab allows for the edition of the characters animations**

Several features of animations in <e-Adventure> must be detailed. Animations are a set of images, drawn one after the other. The basic sort of animation uses JPG or PNG images with the same name but with the \_N suffix, where N is 01, 02, 03, etc. depending on the position of the image in the animation. For example, an animation called “characterSpeaking” made up of 5 PNG images is made up of “characterSpeaking\_01.png”, “characterSpeaking\_02.png”, “characterSpeaking\_03.png”, “characterSpeaking\_04.png” and “characterSpeaking\_05.png”.

A more advanced kind of animation is available, which allows images to be used in a more flexible way and the addition of sound. Animations are automatically converted to this format when clicking the “Create/edit” button and their edition is detailed in section 3.11.1 of this guide.

The animations that must be configured for a character are:



- Standing animations: looking up, looking down, looking right and looking left (this last one is optional)
- Talking animations: speaking up, speaking down, speaking right and speaking left (this last one is optional)
- Using animations: object to the right, object to the left (optional)
- Walking animations: walking up, walking down, walking right and walking left (optional)

When an animation to the left is not provided for an action, a mirrored image of the animation to the right is used.

### 2.5.2. Character dialog configuration

This panel has several options that affect how the player speaks. The first set of options refers to how the text will be displayed in the screen:

- Show speech bubble: If selected, the dialogs of the character will be shown inside a bubble just as in comic books.
- Font front color: this button allows for the configuration of the front (or center) color of the font. Clicking the button will open a color chooser (Figure 29).
- Font border color: this changes the color for the border of the font.
- Bubble background color: this button changes the color for the background of the bubble. During the game, this will be filled with a semitransparent tone.
- Bubble border color: this button changes the border of the bubble.

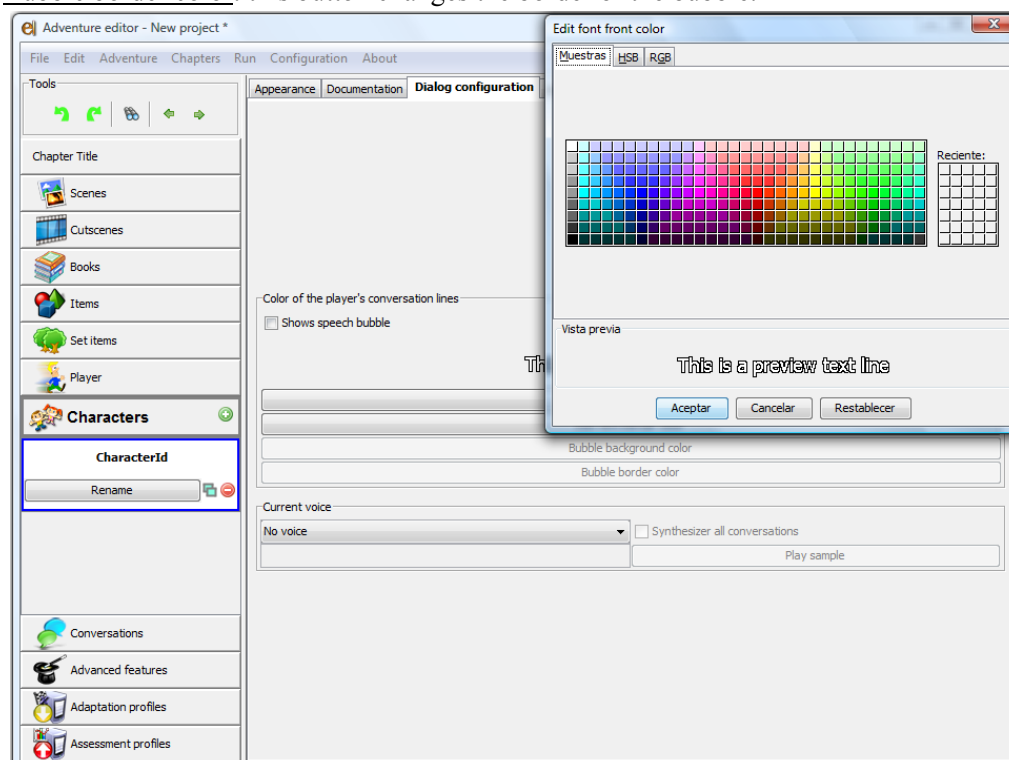


Figure 29. Character edition panel in the “Dialog configuration” tab. This shows the dialog to choose the color of the main font color

The second set of options is used to add a voice, used by the text-to-speech system, when the character speaks. This is used both to improve accessibility and to add audio to the game at a small expense. Checking the “Synthesize all conversation” option will make the character pronounce every line in the game, while if this option is not selected the option will have to be done for each line (see “Conversations” in section 2.5.4).

### 2.5.3. Adding characters to the scene

Characters are added to scenes in the “Element reference” panel. This is done just like for the other elements (see 2.2.3 or 2.3.3).

### 2.5.4. Conversations

Just like the player interacts with items, the player might also interact with characters. However, the kinds of interactions are significantly different. In the “Actions” tab of the characters we find an action specific for characters: “Talk to”. This sort of action requires the existence of conversations in the chapter. Conversations are dialogs between the player and one or more characters, and are created for 3 aims: guiding the player, provide information, or evaluate the player.

More than one character, besides the player, can take part in a conversation. Just like in classic *Lucas Arts™* games, sometimes the player might be presented with a list of options to choose from. These options correspond with the following line the player will utter. The path followed by the conversation will change as a consequence.

Conversations should be bound to a character to be triggered when the player talks to it. To do this, first a new conversation element must be created by selecting “Conversations” in the structure panel and clicking on the add button (⊕). The conversation will then be then get bounded to the character by adding a “Talk to” action in the character’s “Action” tab.

Conversations have a graph structure, made up of a set of nodes and the links between them. All conversations start at an initial node and follow a linear path (one node is “read” after the other). Bifurcations can be added to the conversation, adding special nodes named “option nodes”. Circular paths can also be created to repeat parts of a conversation (e.g. until the correct answer to a question is given).

Nodes in a conversation use different graphic representations, depending on their type, and can be linked to each other. There are three basic node types:

- Dialog nodes. This nodes have dialog lines that will be read by the characters and player in the given order. This nodes are displayed as circle for the case of the initial node and as a rectangle with a curved bottom when a regular node (Figure 30).



**Figure 30. Dialog nodes. (a) represents the initial node in the conversation and (b) any other dialog node**

- Option nodes. This nodes specify the options that the player has to choose from when that point in the conversation is reached. The selected option will be the next line of dialog that the player

reads, and will determine the path followed by the conversation. They are represented with a diamond (Figure 31).



Figure 31. Options node

- End or finish nodes. These are just like the dialog nodes, but they represent the end of the conversation. They are displayed as a rectangle with a triangle on top (Figure 32).



Figure 32. End or finish node

Any node in a conversation can be selected by clicking on it (Figure 33). When a node is selected, its border will be highlighted and an option panel appears next to the node. These options allow for the creation of new nodes of the different types. In the lower part of the panel the lines of the selected node are visible and the character, text and other features of each line can be edited there.

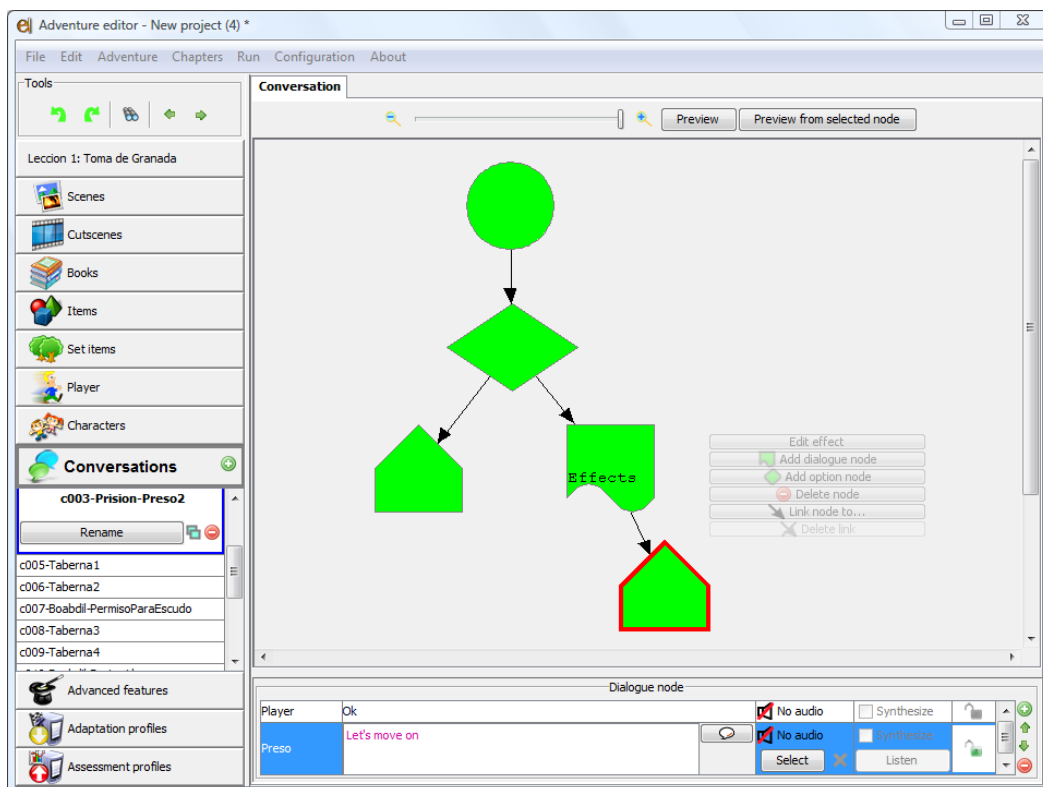


Figure 33. Conversation edition panel with a node selected

When any node in the conversation is selected, the lines of the node will be displayed in the bottom part of the panel. The add (+) and delete (-) buttons can be used to add and delete new dialog lines. The lines can be then edited in the table (Figure 34).



Figure 34. Conversation line edition

The down (↓) and up (↑) buttons next to the lines can be used to sort them. The first column in the table defines who speak the line, among all the characters in the chapter or the player. The second column provides a button to configure an audio track for the line to allow for recorded conversations to be played. The third column provides a checkbox to select the line to be synthesized (only when a voice was configured for the character, as shown in 2.5.2). The last column allows for the definition of conditions for when the line should be used (the definition of conditions is studied in section 3.1).

For every line in the conversation the O button (O) can be used to define the type of bubble to be used when displayed. This way, different feelings can be expressed such as “thinking”, “angry”, etc.

The option panel appears only when a node is selected. The options in this panel vary depending on the type of node and of its position along the graph (Figure 35).

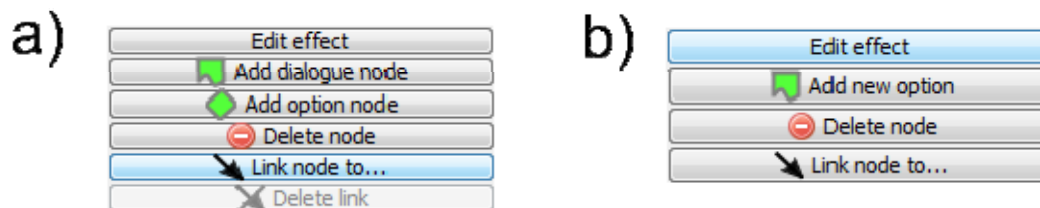


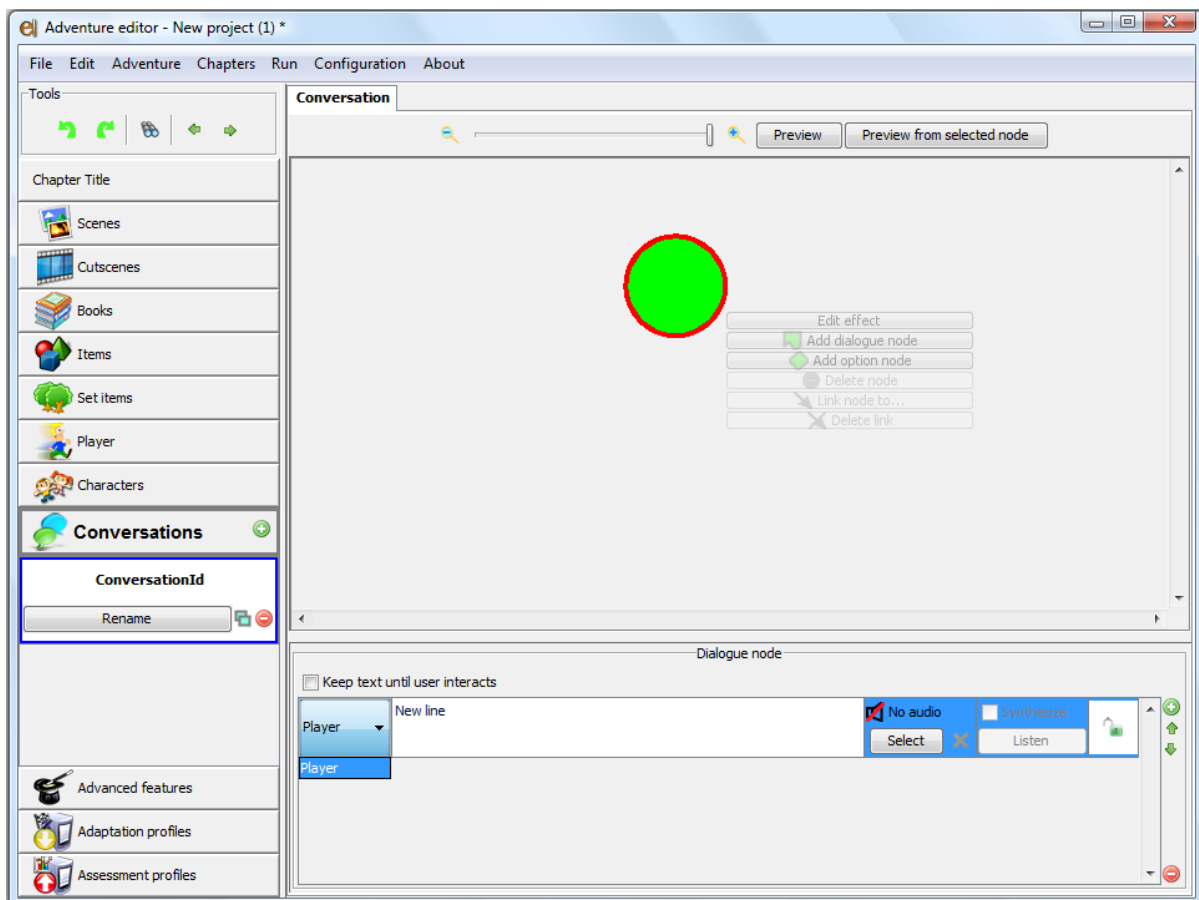
Figure 35. Option panels for different nodes in the graph. (a) shows the options available for dialog nodes and end nodes, while (b) shows the options for option nodes

The options available for a node are:

- Edit effects: this is used to define the block of effects that is triggered after all the lines in the node are read. These are game effects, just like the consequences of actions (see 3.1 for a detail explanation). Nodes with effects will have the word “Effect” printed inside them.
- Add dialog node/Add option node: these buttons add a new dialog or option node just after the currently selected node. For option nodes, this button reads “Add new option” and will add a new dialog node as the result of a choice of the player.
- Delete node: delete the currently selected node and all its children.
- Link node to...: allows to create a link between the currently selected node and an already existing node in the graph.
- Delete link: deletes the link to the following node, and, if necessary, the node and all its children.

Dialog and end nodes have an option to wait for the user input after each line, making the text in the game easier to read. This option can be configured globally or for each node in the different conversations. The global option is activated in the *Adventure > Edit Adventure data...* menu. By checking the “Keep text until user interacts” option the game will wait for the user after each line and the option will not be configurable at a node level. If the option is not selected, however, it will be

configurable at each node by activating the option that will be visible on top of the conversation lines (Figure 36).



**Figure 36. Edition of a dialog node**

Option nodes have different options that can be selected (Figure 37):

- Order options randomly: When selected, the order of the options presented to the player will change each time.
- Show previous question: When selected, the last line of the previous node will be kept on screen while the player answers.
- Show response: When selected, after the user selects an answer, the following line will be the answer itself.

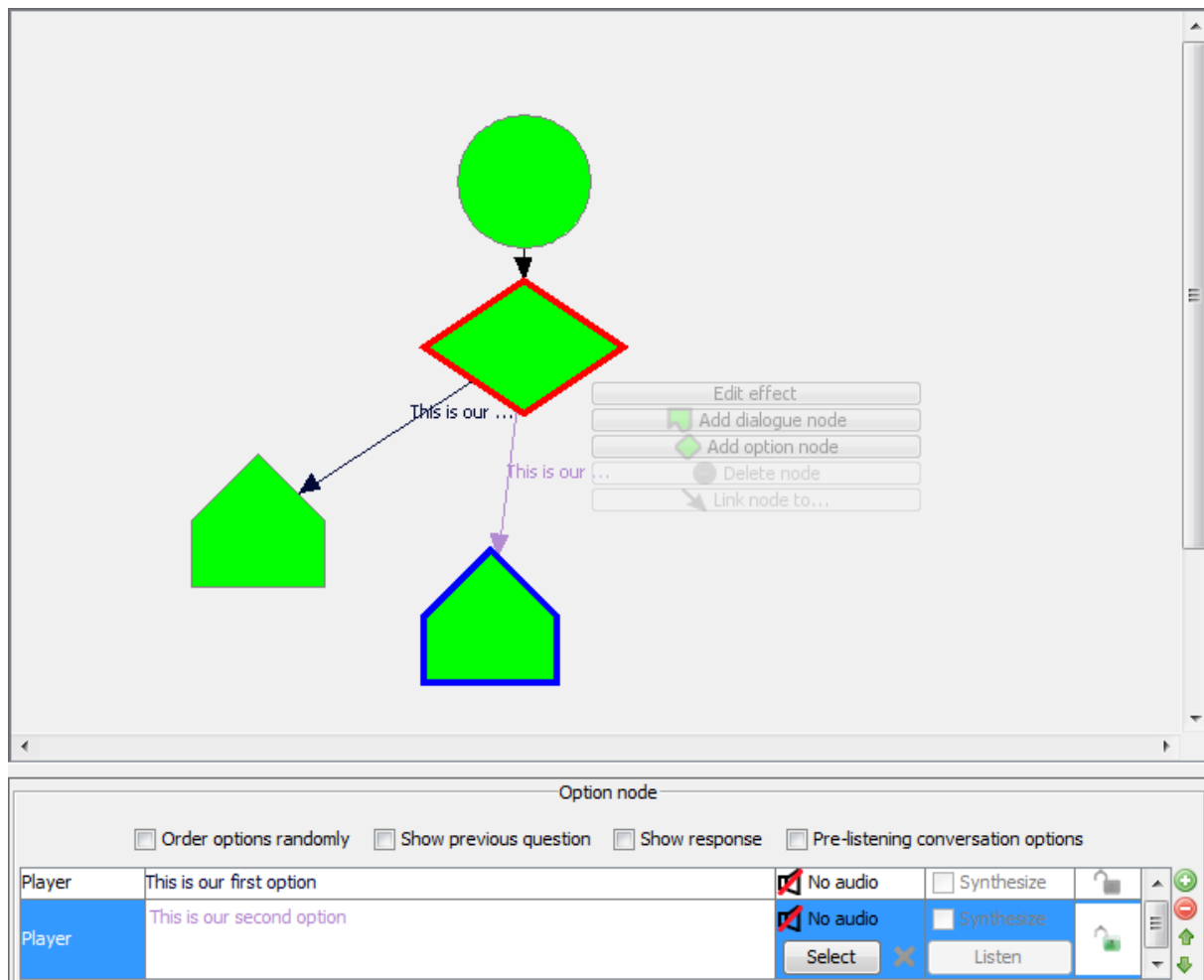


Figure 37. Edition of an option node

### 2.5.5. EXAMPLE: Editing a conversation

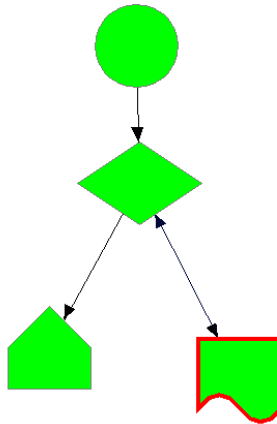
Starting from a new conversation, we select the initial node (this node appears by default in the conversation). We will add two lines to the conversation, the first one read by the player and the second by a character. To choose who read the line, we are presented with a combo box with all possible characters. The text of the line is written directly into the box.

The next step is to present the player with two options. To do this, we select the node and using the option panel we add a new option node. The new node will be automatically linked to the selected node.

Options are added to the option node by choosing the “Add new option” button in the panel. This will, at the same time, create the target node for the option. This node will be a dialog or end node. The option will be added as a line of the option node, which when selected will highlight the corresponding target node. We can use “This is our first option” and “This is our second option” for the text of the options of the node. This will result in a conversation as the one shown in Figure 37.

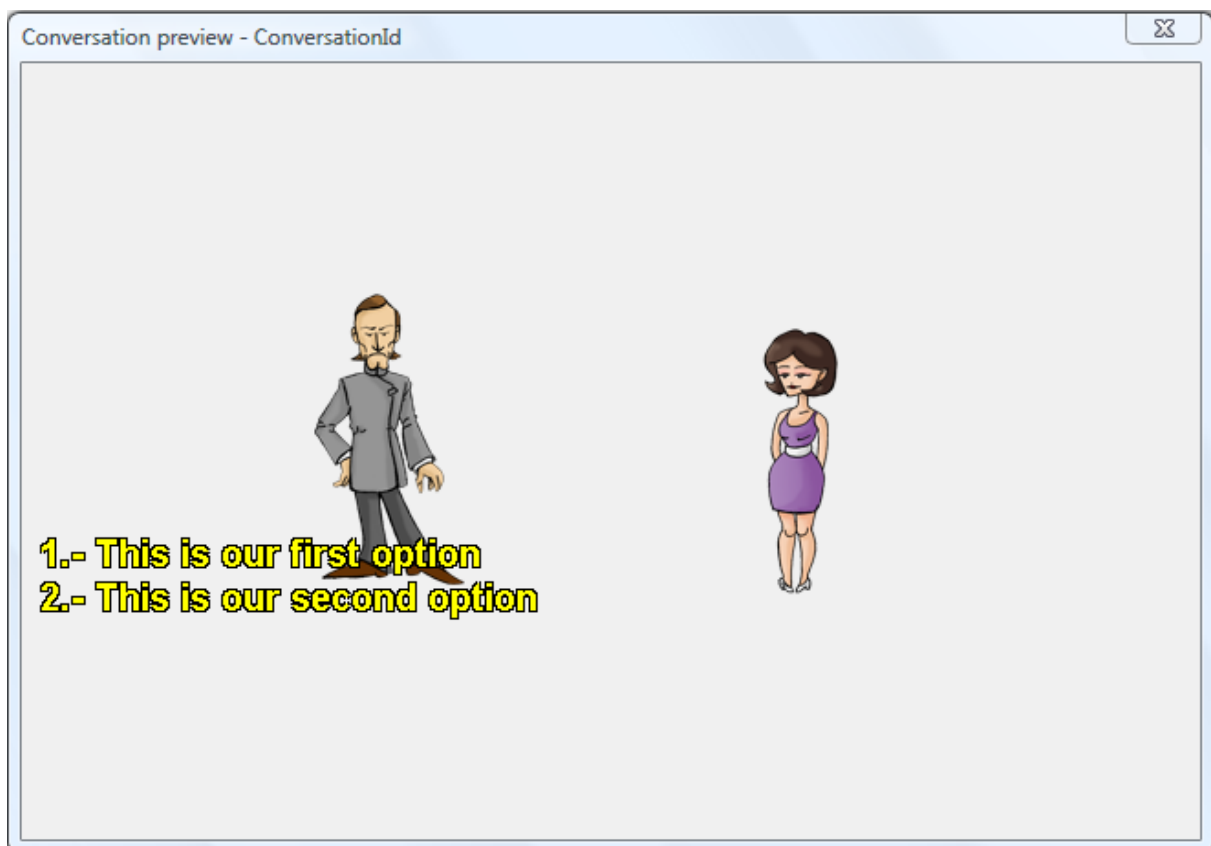
Finally, we will create a loop in the conversation, making the first option the write choice and presenting the user with the question once again if the second option is chosen. To do this, we select the

target node of the second option, click the “Link to node...” button and end by choosing the option node. The result should look something like Figure 38.



**Figure 38. Creating loops in the conversation edition panel**

The conversation can be previewed in full at any time by clicking the “Preview” button at the top (Figure 39). When a node is selected, the “Preview from selected node” is enabled, which allows for the preview of just a part of the conversation.



**Figure 39. A preview of a conversation**



### 2.5.6. The player

The player, in third person games, is the on screen avatar of the real life player. In first person games just a few of the configuration values are available for the player. This is a character that responds to player input and interacts with the different game elements. This player character can be configured just like any other character, except for the fact that it lacks actions.

## 2.6. Books

Books are included in <e-Adventure> as a way to provide large amounts of information in a natural and easy to access manner. Books can be displayed at any time during a game and are of two basic types:

- Simple books: These books are edited directly in the <e-Adventure> editor and are made up of paragraphs, titles, unformatted text, bulleted lists and images. The contents are spread into different pages as needed and are easy to create.
- Formatted text books: These books are used to present documents created by an external edition tool with support for RTF or HTML files, such as Microsoft Word or OpenOffice. The document will be displayed in full screen and the content will not be automatically divided into different pages (this is up to the creator). HTML files can also be created directly within <e-Adventure> as shown in section 3.11.2 of this guide.

### 2.6.1. EXAMPLE: Create and edit a simple content book

The first step is to create a new book of this kind. To do this, we select “Books” in the structure panel and then click the add button (🟢). When asked for the kind of the new book, we choose “Simple content book”. The new book edition panel will appear (Figure 40).

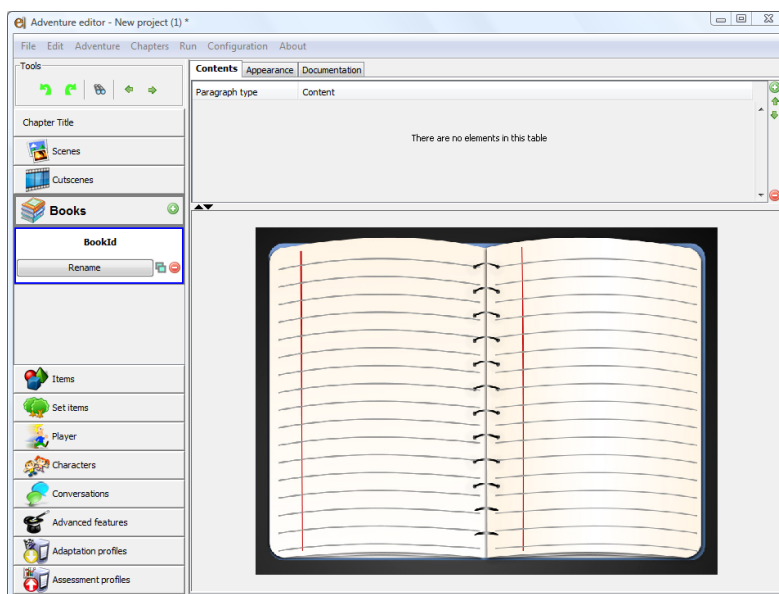
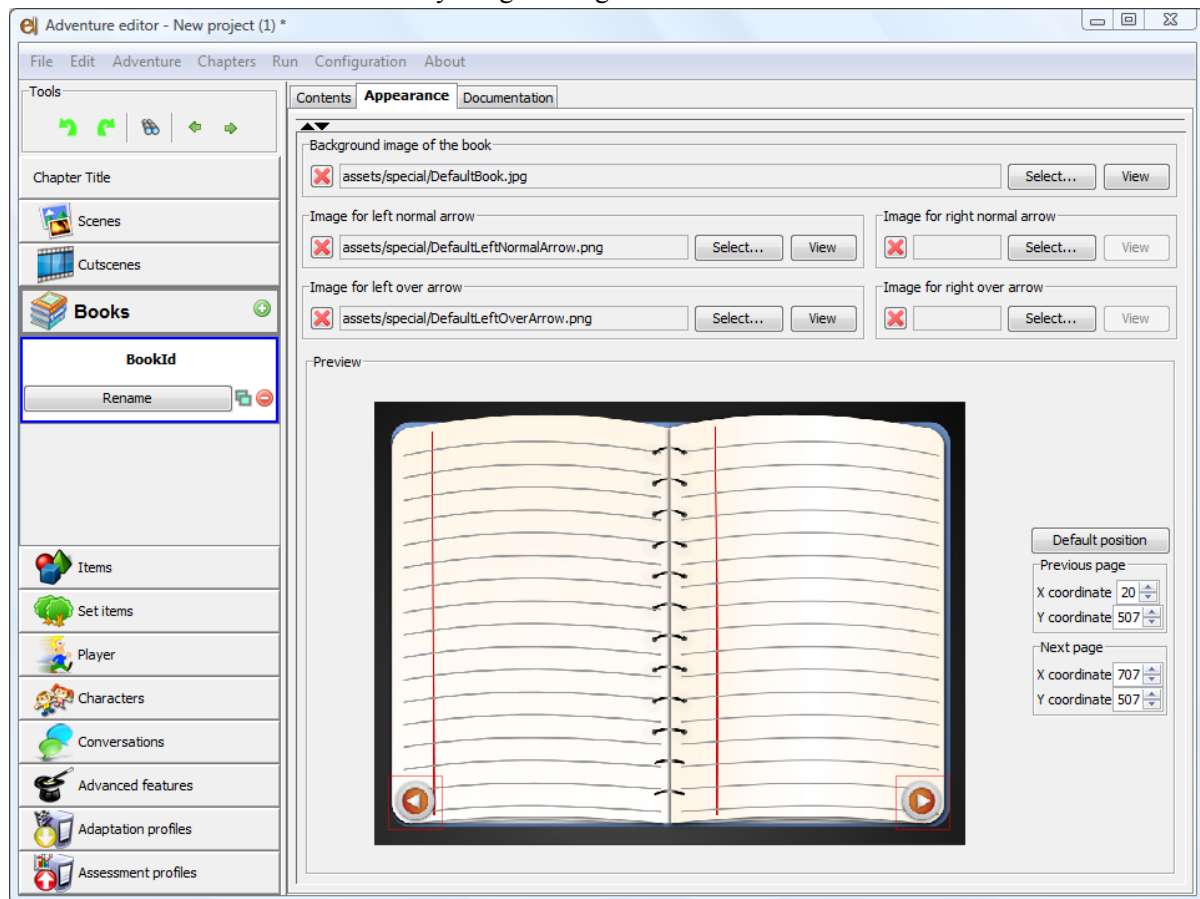


Figure 40. Simple content book edition panel in the “Book content” tab

There are three different tabs in this book edition panel:

- Book content: the contents of the book (paragraphs, images, etc.) are edited in this tab.

- **Appearance:** the looks (i.e. graphic assets) of the book, including the background, arrows, etc. are edited in this tab (Figure 41).
- **Documentation:** this tab allows for the documentation of the book for future reference and the information here is not used by the game engine.



**Figure 41. The “Appearance” tab in the simple content book edition panel**

In this example, we will create a book that has a title, a paragraph and an image. This book will provide information about raspberries. To do this we must first select the “Book content” tab.

The edition panel has two main sections. In the bottom we have a preview of the book that we are creating. At the top we find an ordered list of all the elements (paragraphs, images, etc) that the book has. The buttons to the right of the list can be used to add, remove and sort these elements. To add a title we click the add button (+) and choose “Title” from the options that appear. We then modify its contents directly within the list, change it to “Raspberries” (Figure 42).

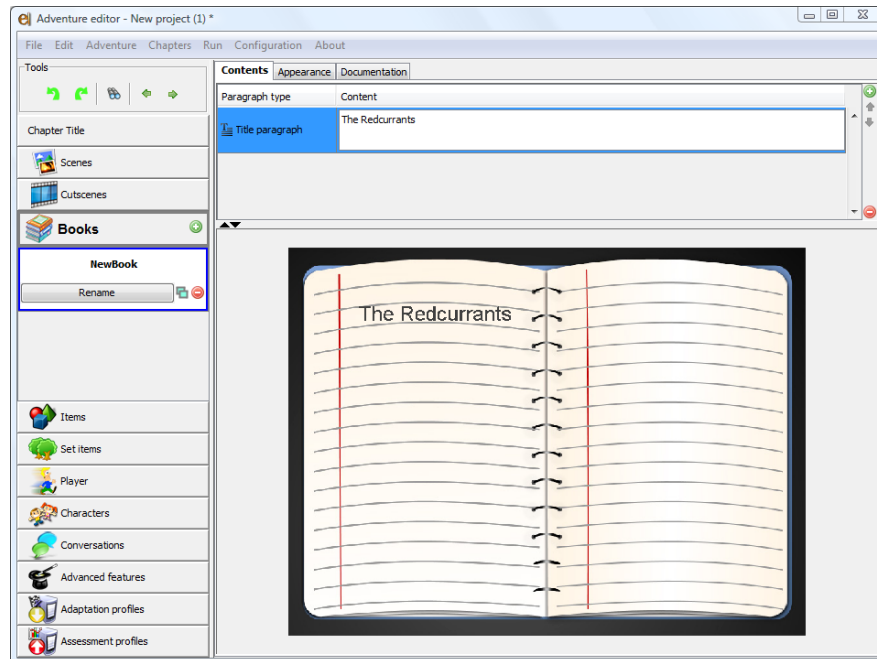


Figure 42. Simple content book edition panel, after adding a title element

After this, we repeat the process for a text paragraph. The contents of this paragraph can, for instance, be copied from Wikipedia. Finally, we add a new image paragraph and choose an image of some raspberries (Figure 43).

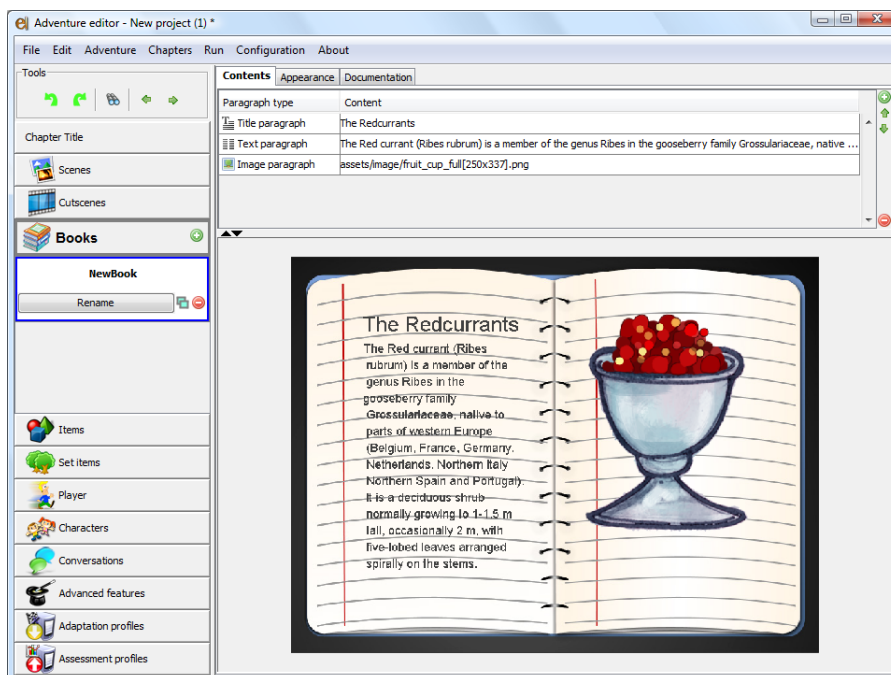




Figure 43. Simple content book edition panel, after adding an image paragraph

The preview allows us to see the result as it is to be shown in the game. We can use the  and  arrows next to the paragraph list to sort them. For instance, we can place the image before the text paragraph (Figure 44).

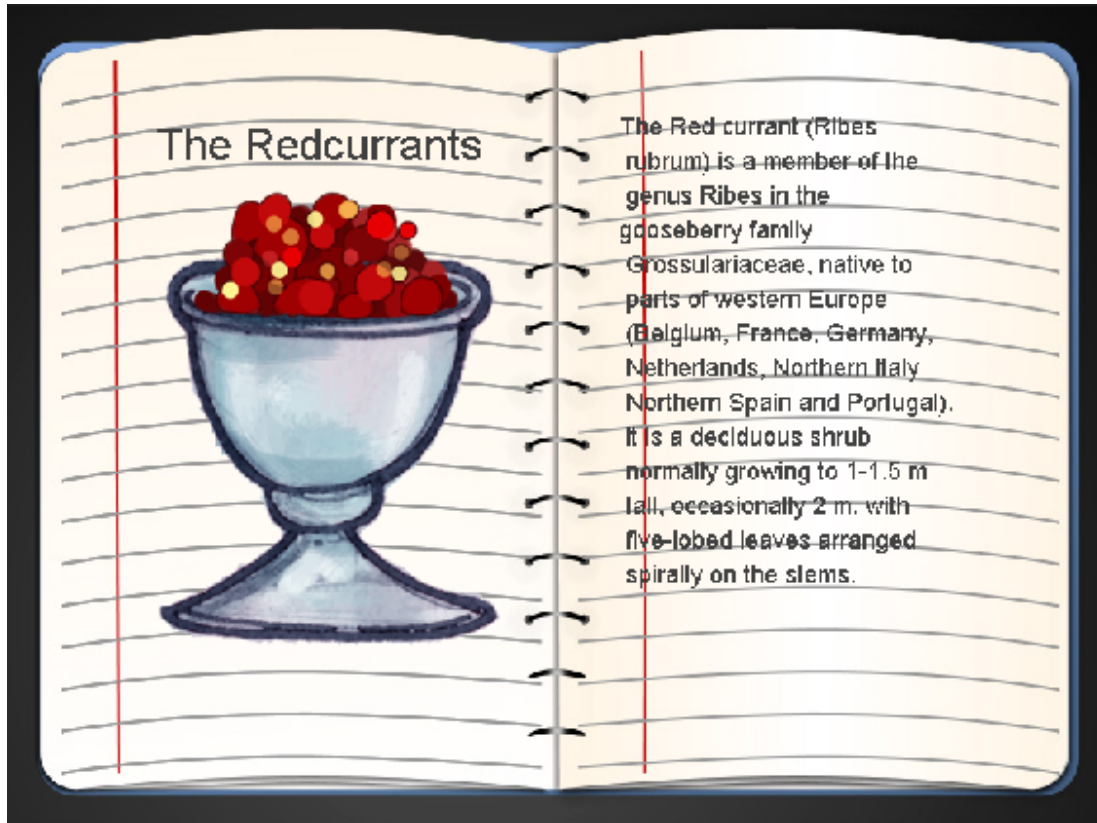


Figure 44. Simple content book edition panel, after resorting the elements in the book

### 2.6.2. EXAMPLE: Create and edit a formatted text book

A book similar to the one in the previous example can be created using the formatted text option. This book will use an HTML file.

It should be noted that HTML file support in Java is limited and some problems can arise if files with meta-data are used. Most of these problems, however, should be fixed automatically by the <e-Adventure> platform. Besides, exporting files as “Filtered webpage” from Microsoft Word might help achieve better results.

Another important aspect of HTML files is that all required files (for instance, images) should be included in a folder with the same name as the file but ending in “\_files”. For instance, the images for “raspberries.html” should be placed in a folder named “raspberreis\_files”.

HTML files created within <e-Adventure> take all these features into account and there is no need to consider them.

In this example, we use the same information that in the previous one, but in this case we write the information into one HTML file (Figure 45). This sample HTML files uses different font sizes and colors, and it includes an image.

**The Red Currants**

The **Red currant** (*Ribes rubrum*) is a member of the genus *Ribes* in the *gooseberry* family *Grossulariaceae*, native to parts of western *Europe* (*Belgium*, *France*, *Germany*, *Netherlands*, Northern *Italy* Northern *Spain* and *Portugal*). It is a *deciduous shrub* normally growing to 1-1.5 m tall, occasionally 2 m, with five-lobed *leaves* arranged spirally on the stems. The *flowers* are inconspicuous yellow-green, in pendulous 4–8 cm *racemes*, maturing into bright red translucent edible *berries* about 8–12 mm diameter, with 3-10 berries on each *raceme*. An established bush can produce 3-4 kilos of berries from mid to late summer.

**Figure 45. A web page about raspberries**

Books with formatted text are created in the same way as simple content book, but in this case we choose the “Formatted text” option just after clicking the add button. The resulting panel will look like the one in Figure 40. In this case, the top section of the panel is used to manage the pages of the book. We’ll start off by adding a new page, creating a new series of editable field in the table.

There are three different options as to how to provide the content of the page:

- **Local resource:** This option allows us to select a file from our system. The selected file will be copied automatically to the project folder.
- **URL:** This provides access to documents hosted in a web server. This allows us to select a web page that might need to be updated at a further time, however, this option would require the user to have an active Internet connection. When this kinds of pages are used, the reference will need to be validated and the icon to the right will turn green if OK.
- **Image:** This allows us to use any image as a page in a book. For instance, this can be used to attach slides of a PowerPoint presentation (if exported as image files) as pages of a book.

Besides, the margins for each page in the book can be edited independently from the table. This can be useful for big images that do not fit correctly on the screen, or if the background image needs a certain margin (such as the one used by default). All four margins can be edited independently in the dialog that appears for that purpose.

The changes to the book are reflected automatically in the preview section of the panel. For a full preview, there is a “Preview” button included. After adding the HTML file to the book, the preview button results in a window as the one in Figure 46.

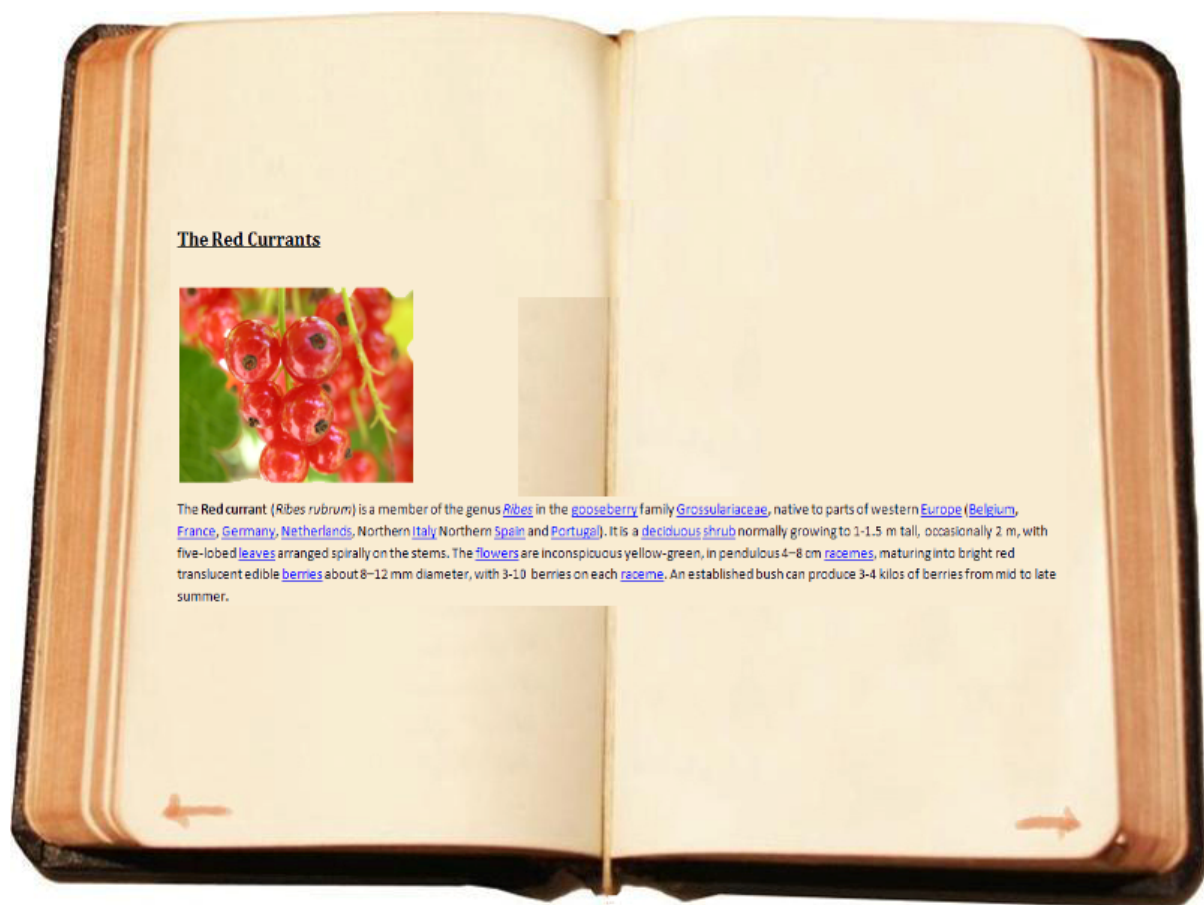


Figure 46. Preview of a formatted text book

The other button in the lower part of the panel, “Export HTML book” creates a version of the book as image files for use in the engine (instead of directly using the HTML files). This is provided to optimize the engine, especially in regards to future platforms where HTML support might be limited. We encourage you to use this feature when no further editing in the book is expected.

## 2.7. Cutscenes

<e-Adventure> cutscenes are used to display multimedia content that are shown in full-screen mode, have limited user intervention and after finishing are followed by new scenes or cutscenes. There are two basic kinds: Slidescenes and Videoscenes. The ones of the first kind are made up of a series of images. The ones of the second kind use videos in MOV, AVI or MPG (with some important format and codec restrictions, due to limited support in Java).

### 2.7.1. Slidescenes

To add a new slidescene, we click the add button next to the “Cutscene” element when it is selected in the structure panel. In the dialog, we choose the “Slidescene” option and this will result in a new panel with three tabs (Figure 47).



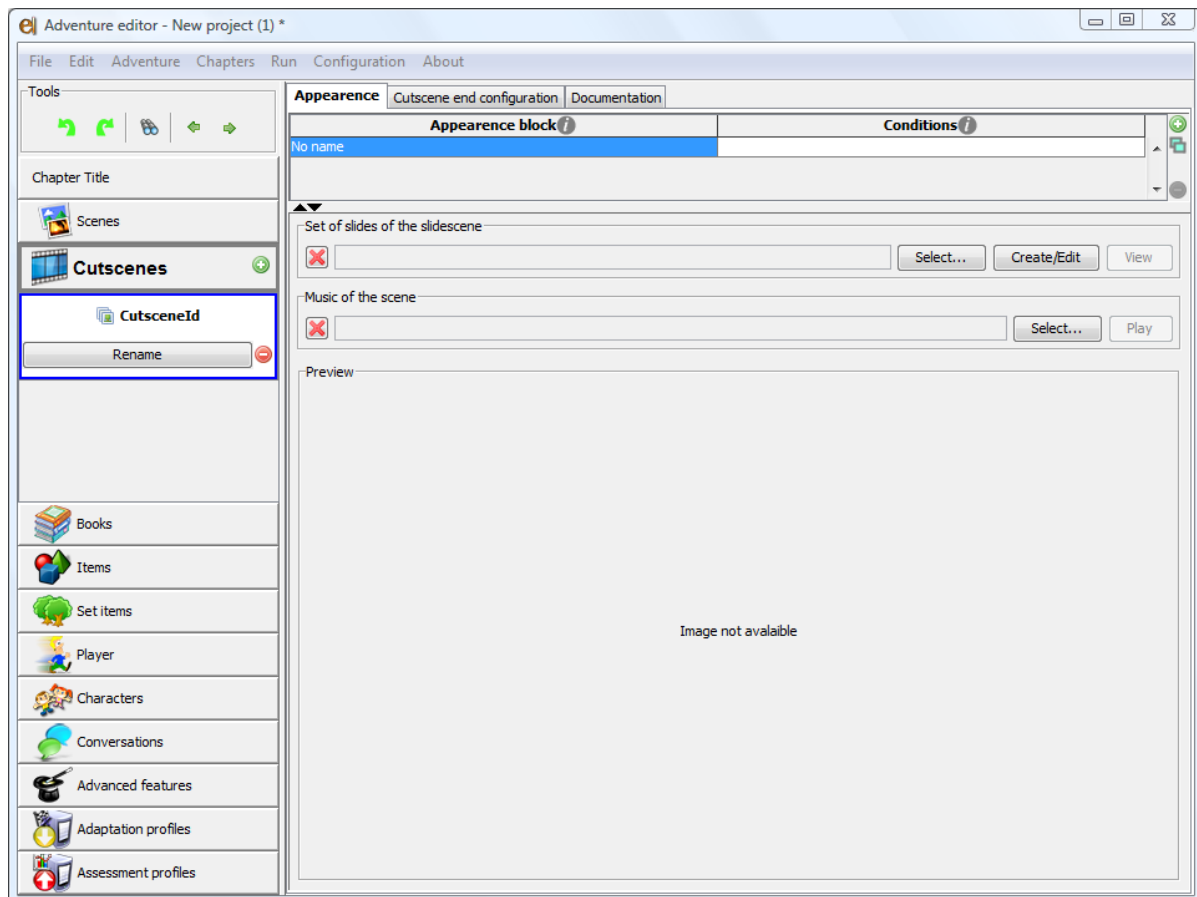


Figure 47. Slidescene edition panel

The three tabs in the slidescene edition panel are:

- **“Appearance”**: this is used to set up the images that make up the slides of the cutscene. Slidescenes use animations, that can be configured either by providing files with consecutive numbers (e.g. “slide\_01.jpg”, “slide\_02.jpg”, etc.) or by clicking the “create/edit” button and using the animation editor (see 3.11.1)
- **“Documentation”**: as in other <e-Adventure> elements, the documentation is not used by the game and just provides additional information during development.
- **“Cutscene end configuration”**: this tab is used to establish what happens when the cutscene ends, see 2.7.3 for details.

Slidescenes are sometimes used as title sequences, at the start or ending of a chapter. For instance, Figure 48 shows the preview of a slidescene that will be used as an opening title in a game.



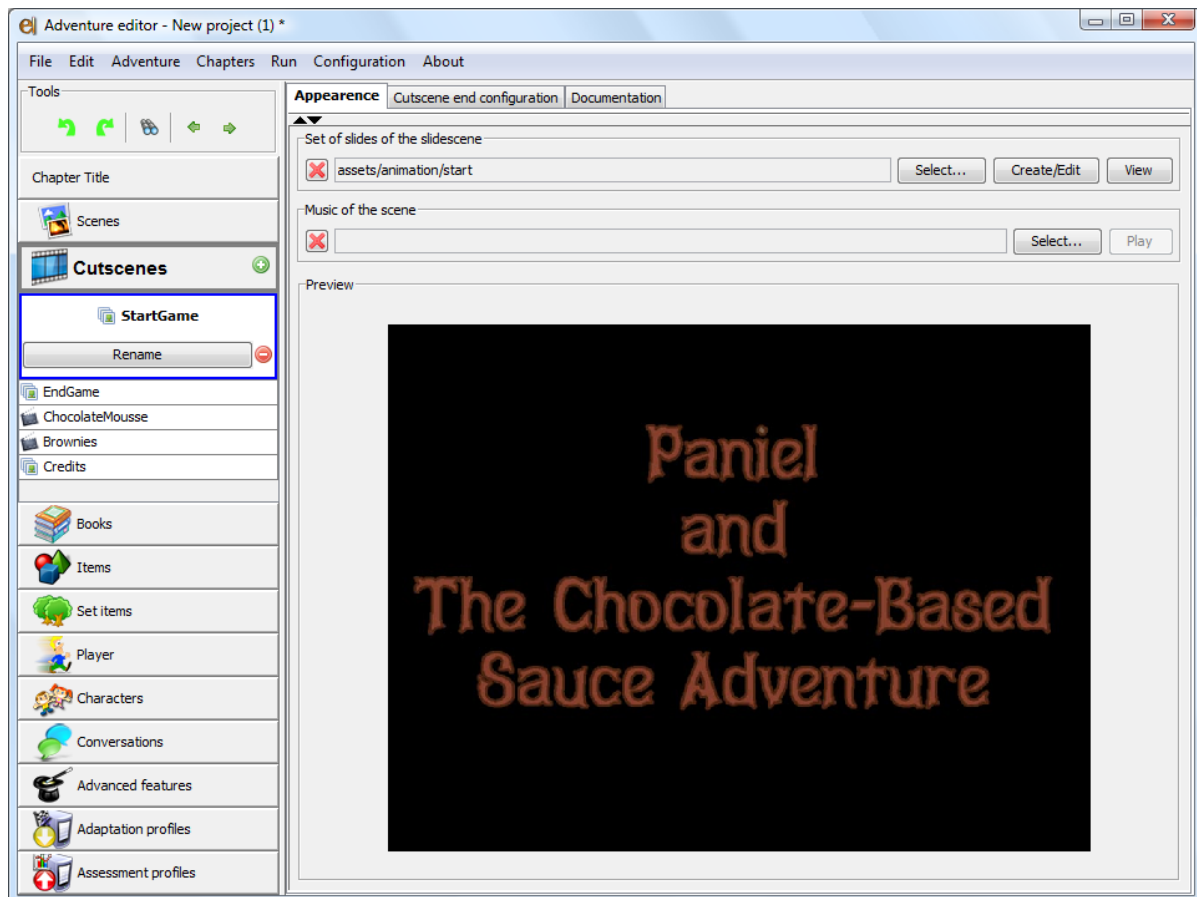


Figure 48. Cutscreen edition panel after adding a slide. The slide is the opening title for a game

### 2.7.2. Videoscenes

Videoscenes are added in a similar fashion to slidescenes. The tabs in the videoscene edition panel are the same that are found in the slidescene panel, with the only difference that the asset required in this case is a video file. After selecting a file, a “Play” button allows us to test the video and its compatibility with the platform.

<e-Adventure> supports some kinds of MOV, MPG and AVI files. In particular, MPG files must use the MPEG1 video codec. MOV and AVI files can use DIVX4, XVID, DIVX3low, S-Mpeg 4 v2 and DIVX5, although further restrictions apply.

Next formats and codecs have been tested in <e-Adventure>:

TIPO DE FICHERO	TIPO DE CODEC	FAMILIA DE CODECS	SOPORTADO
.AVI	DIVX 4	MPEG-4V	YES
.AVI	XVID	MPEG-4V	YES
.AVI	FFDS (VIDEO FLASH)		NO
.AVI	AVC	AVC	NO
.AVI	AVC	H264	NO
.AVI	MJPEG	MJPEG	NO

.AVI	mpg1		NO
.AVI	mpg2		NO
.AVI	WMV1	Win Media Video 7	NO
.AVI	WMV2	Win Media Video 8	NO
.AVI	DIVX3low	MPEG-4	YES
.AVI	S-Mpeg 4 V2	Microsoft MPEG-4 V2	YES
.AVI	MS-mpeg4 v1	MPEG-4V	NO
.AVI	DIVX5	MPEG-4V	YES
.AVI	MPEG-4 visual	MPEG-4V (fmp4)	NO
.AVI	MPEG-4 video	MPEG-4V (mpv4)	NO
.MPG	MPEG1 video	MPEG-V	YES
.MPG	MPEG2 video	MPEG-V	NO
.MOV	MPEG 4 visual	MPEG 4V	NO
.MOV	H 263	H 263	NO
.MOV	AVC	AVC	NO

**Figure 49. Audio and video formats tested in <e-Adventure>**

In this table there are more results for .AVI than the rest of formats. But the results about codifications here detailed can be extrapolated to the rest of formats. For example, a .MPG or .MOV file with DIVX5 codec, will be supported, and files with WMV1 won't.

### 2.7.3. Connecting cutscenes with other scenes

Just as regular scenes, cutscenes can be connected to other scenes in the game. This will affect the game in meaningful ways. The process is very similar to the one used for regular scenes, just that there is only one exit as cutscenes can not be interacted with. This is configured in the "Cutscene end configuration" by choosing one of the available options (Figure 50).

- Returns to the previous scene: if selected, when the cutscene ends the game will go back to the previous scene (either regular or cutscene). If the cutscene is the first in the game, an error will arise and the user informed.
- Goes to a new scene: This allows us to choose a new scene to be triggered once the cutscene ends. There are some details to be configured:
  - "Next scene": The id of the next scene in the chapter, to be selected from a combobox.
  - "Edit effects": Allows for the configuration of effects that will be triggered after changing the scene to the new one.
  - "Use a destination position for the player" and "Edit the destination position": this option activates the selection of a destination position for the player in the next scene, for games in third person, and the button is used to choose the position.
  - "Transition" and "Transition time": A transition (and its duration) can be selected for the change to a next scene in the game.
- Chapter ends: When this option is selected, the chapter will end, taking the game to the following chapter or finishing if it does not exist.

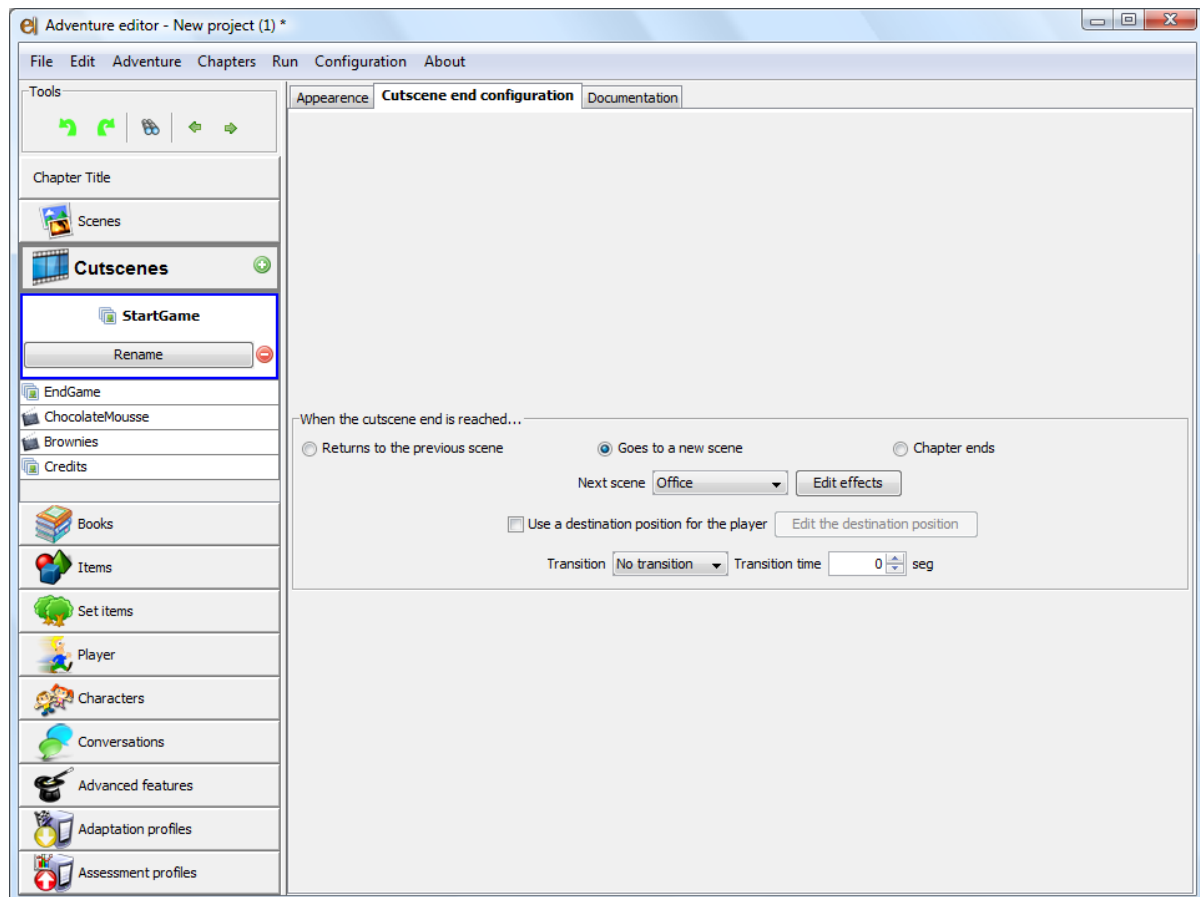


Figure 50. Cutscene edition panel in the “Cutscene end configuration” tab

## 3. Extending basic games: Advanced features

---

The previous section presented all the necessary tools to create our first basic <e-Adventure> game. However, the platform has some advanced features that increase the potential of games created with it. These features are studied in this section.

### 3.1. Conditions and effects

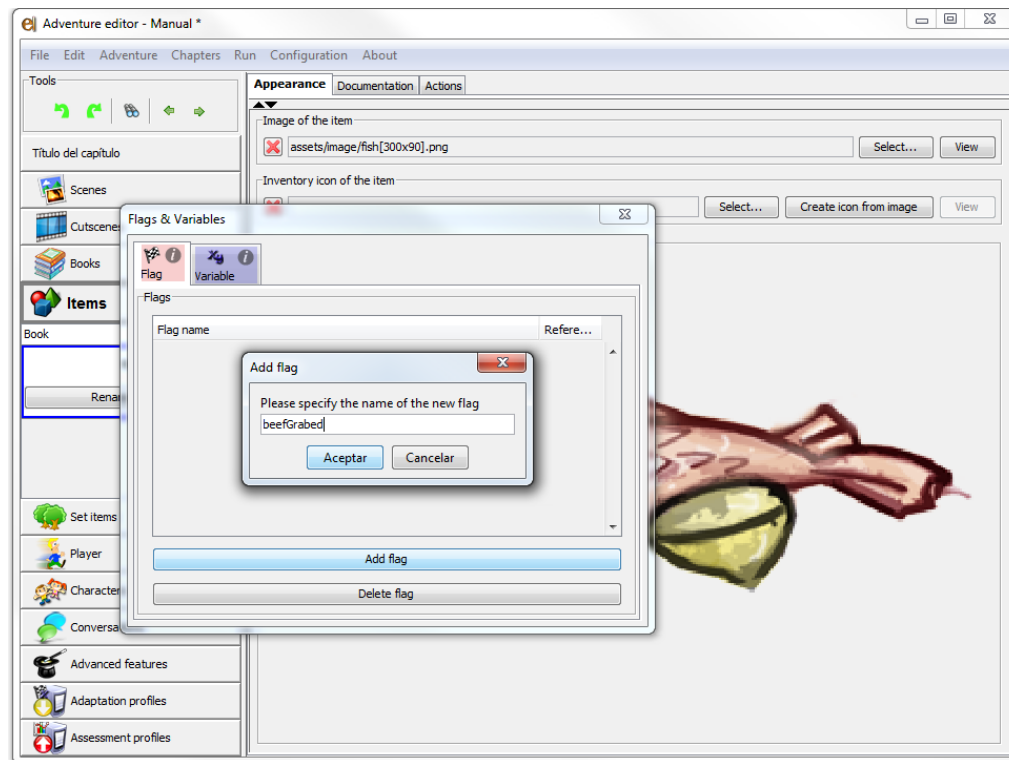
The basic elements in <e-Adventure> are items, characters and scenes. However, they are not enough by themselves to create an interesting game. Games require a narrative, a story to be told. In <e-Adventure> the story flow is managed by *flags* and *variables*. Flags work as switches, at any point in the game can be either active or inactive. New flags can be defined and used to create conditions that can be valid or not at a given point, depending on the status of flags. On the other hand, variables can have integer values and can also be used as part of conditions, depending on whether they are greater, smaller or equal to a given value.

#### 3.1.1. EXAMPLE: A simple condition

Supposing we have a player with 2 items: a beef and a fish. The idea is for the player to only be able to grab the fish after having grabbed the beef. In our adventure this condition must be expressed as conditions and flags. To do this, we create a new “beefGrabbed” flag, which will be inactive until the beef is grabbed. Once the player grabs the beef the flag will become active and the player should be able to grab the fish. The full process is:

1. Add the new flag. This is done in the “Chapter” > “Edit flags and variables” menu (or using the Ctrl+F shortcut), which will show a new dialog. In this new dialog, we use the “Add flag” button and create the “beefGrabbed” flag (Figure 51). After clicking OK we should have a flag with 0 references.

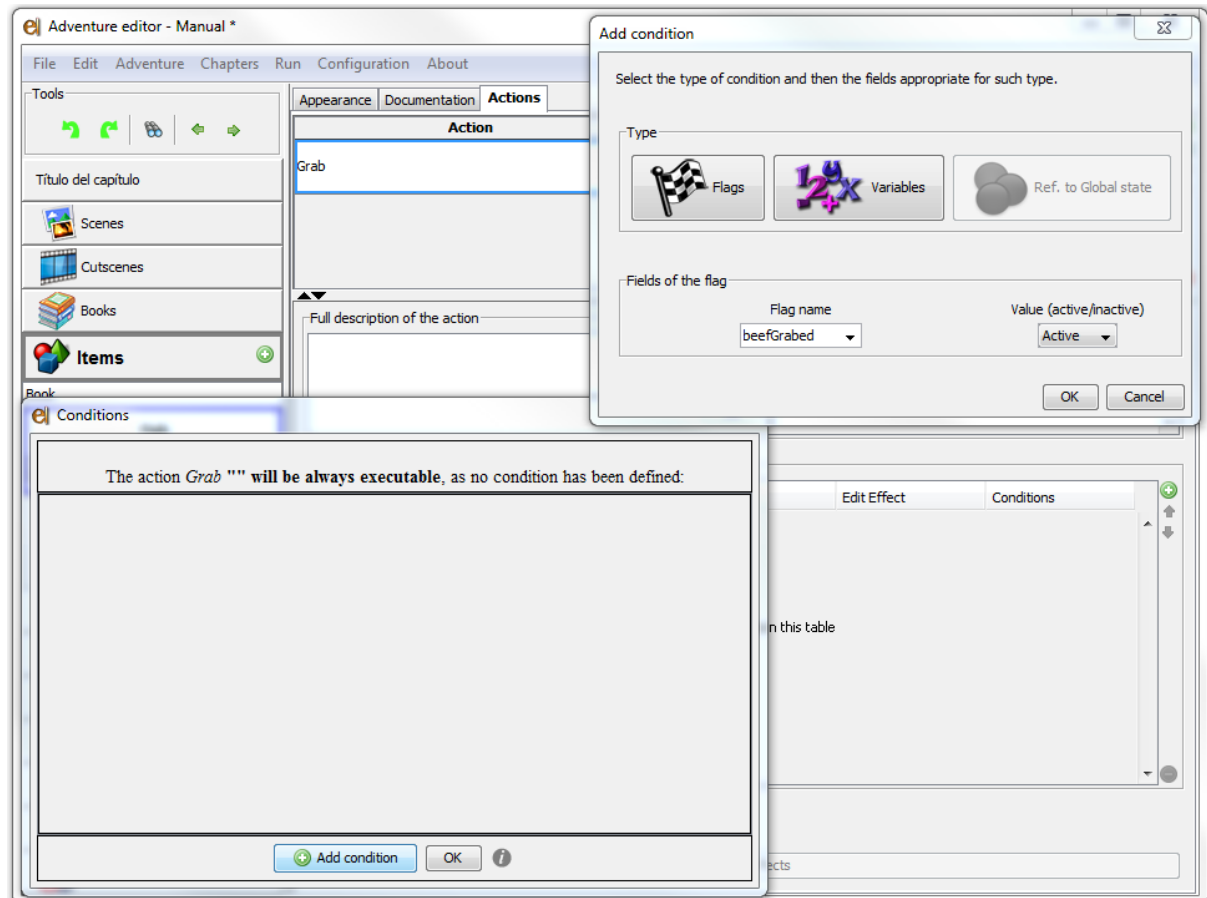
This same dialog can be used to add and delete flags (or variables) for the chapter. However, to delete a flag or variable it must first have 0 references. Besides, flag and variable names should be unique, have no spaces and start with a letter.



**Figure 51. Adding a new flag**

2. Reference the flag in the conditions. We should select the fish element and add a new “Grab” action to it. To do this, first click “Items”, select the “Fish” item from the list, go to the “Actions” tab and click the add button (+) next to the action list. We should then click the conditions button (🔒) of the new action. In the new dialog, we click the “Add condition” button. A new dialog will ask for a flag or variable (Figure 52). We should choose Flag and the “beefGrabbed” id from the list. In the other combo box we select “Active”, as we want the condition to be valid only once the flag has been activated.

It should be noted that the combobox where we choose the flag is editable, meaning that a new flag could also be added as needed, without the need of the “Edit flags and variables” menu.



**Figure 52. Adding a new condition using an existing flag**

3. An effect should be added to the “Grab” action of the beef. This will be studied in detail later in this guide.

### 3.1.2. Variables

As flags can be limited for some situations (such as when something needs to be counted), variables are introduced. Variables are similar to flags, but instead of being either active or inactive, they can have a value of 0 or more. By default, variables start the game being 0.

Adding conditions that use variables instead of flags is done following the same process. However, when adding a variable condition we need to set the id of the variable, the comparison function (<, >, =, etc.) and the value (0, 1, 2, etc.) with which to compare (Figure 53).

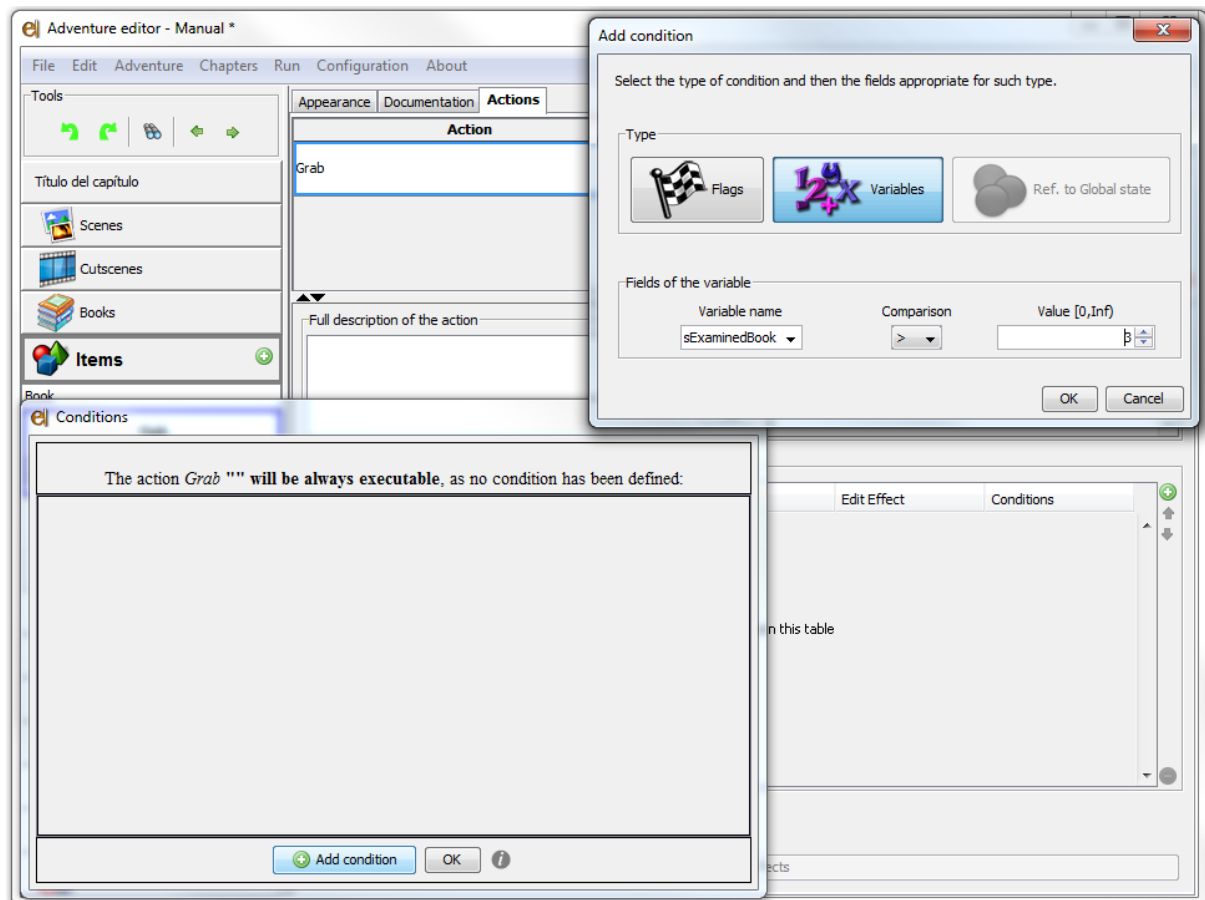


Figure 53. Adding a condition to an action based on the value of a variable

### 3.1.3. Global states

Another, even more advanced, feature in <e-Adventure> is the possibility to define conditions based on “global states” of the games. “Global states” are sets of conditions that are defined as a whole, allowing them to be reused in different parts of the game. These are especially useful in complex games where a point in the game might depend on the combination of different values for several flags and variables.

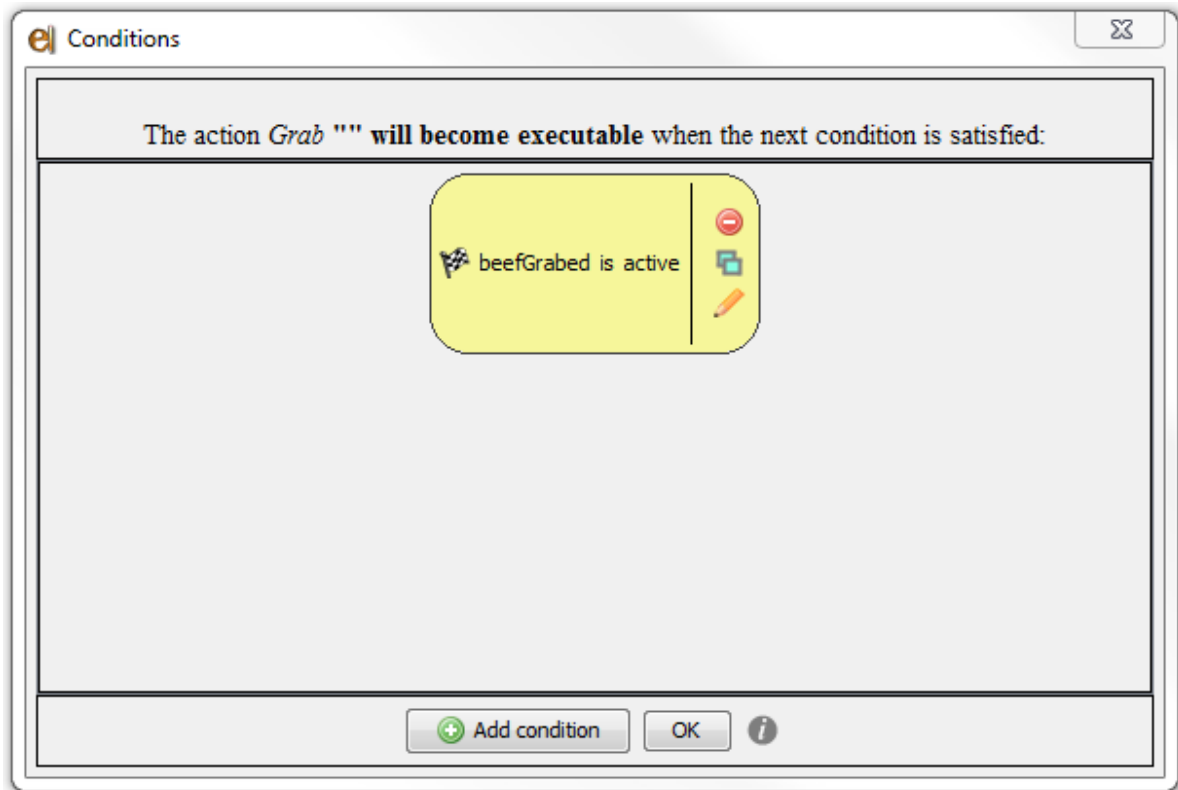
“Global states” are created within their own tab in the “Advanced features” element in the structure panel to the left of the editor. “Global states” are displayed and edited just like the conditions of an element, but they are limited to reference themselves.

When editing the conditions of any element, a reference to a “Global state” is added just like the references to variables or flags were added in the previous sections, but using the relevant button.

### 3.1.4. Condition edition dialog

The condition edition dialog has many features. In the first place, placing the mouse cursor over a condition shows a contextual menu as seen in Figure 54. This menu has buttons to delete, replicate or edit (respectably) the condition.





**Figure 54. Condition edition dialog (showing the contextual menu of a condition)**

Besides, complex conditions with several flags and variables can be created from within this dialog. To do this, more conditions can be added using the “Add condition” button. By default, when a new condition is added it will be attached to the rest of the expression at the end and with the “AND” operator (that means that all the conditions must be true for the whole condition to be true). However, the operator can be changed to an “OR” (in a case by case basis). This is done by clicking the “AND” operator and selecting the desired one from the list.

For example, let’s expand the previous example situation. In this case, the “Beef” object can only be grabbed if the “Fish” was grabbed, or either the “Milk” or “Eggs” (or both) were grabbed. Assuming the “beefGrabbed”, “fishGrabbed”, “milkGrabbed” and “eggsGrabbed” flags are activated when grabbing each of those elements respectively, the condition should be represented as shown in Figure 55.

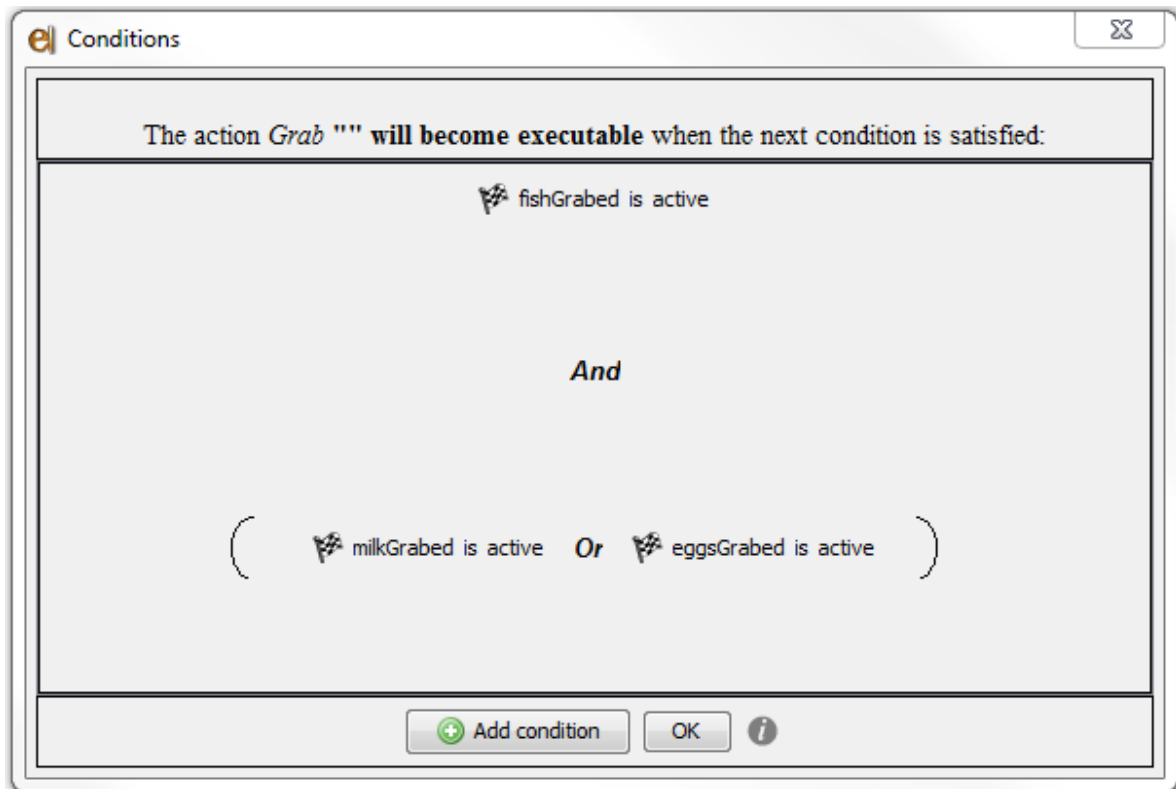


Figure 55. Creating a complex expression with the condition edition dialog

### 3.1.5. What can be done with conditions?

Conditions can be used in almost every part of <e-Adventure> games, for example:

- In actions (for objects and active areas)
- In conversations (for each line)
- In barriers (to activate or deactivate them)
- To change the behaviour of an exit (for example, activating one exit or another in the same place)
- In references to elements, allowing the elements to appear and disappear as needed
- To change the appearance of an element, by activating or deactivating specific appearances (Figure 56).

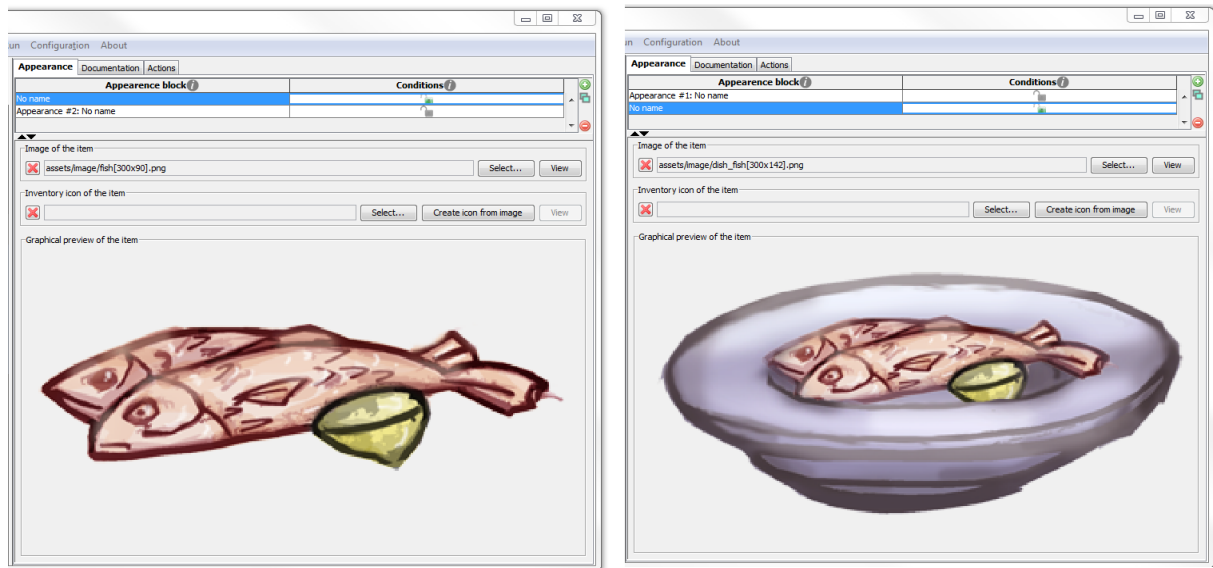


Figure 56. An element with several appearances with different conditions

### 3.1.6. Activating and deactivating flags: Effects

To change the state of a flag, <e-Adventure> uses the effect system. Effects are the results of different situations and are defined for specific elements in the game:

- For actions (in items and active areas): when an action is performed by the user, the effects in its block will be triggered one by one.
- In conversations (for each node): every node in the conversation can have an effect block. After all the lines in the node are read, the effects will be triggered one by one, before the next node is started.
- In transitions (for cutscenes and exits): effects can be added to be triggered just after (post-effects) and, in some cases, just before (pre-effects) the next scene is shown.

Following the same example of the previous sections, if we want to activate the “beefGrabbed” flag when grabbing the beef we have to edit the actions of the item. In the “Actions” tab, we should add the “Grab” action, and in the lower part of the panel we should add a new effect (using the add button (+) to the right of the list). The dialog will allow as selecting an effect (in this case, an “Activate” effect (Figure 57). When clicking OK, a new dialog will ask us to select the id of the flag we wish to activate.

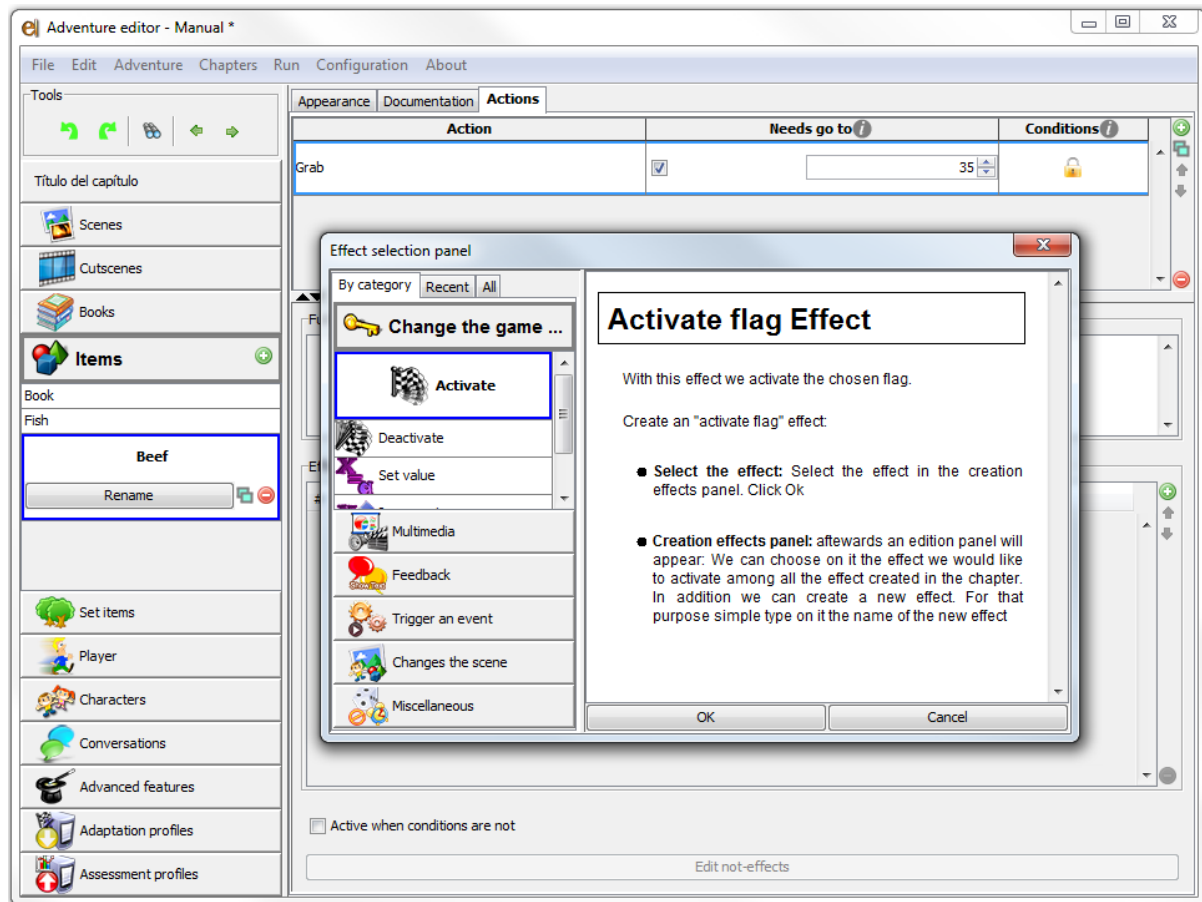


Figure 57. Editing the effects of an action, selecting the “Activate” effect

### 3.1.7. Setting the value of a variable

Just as flags can be activated and deactivated through the use of effects, variables can be given a value using three different effects (Figure 58):

- “Set value”: give the variable a pre-defined value.
- “Increment var”: increase the value of the variable by a given amount.
- “Decrement var”: decrease the value of the variable by a given amount.

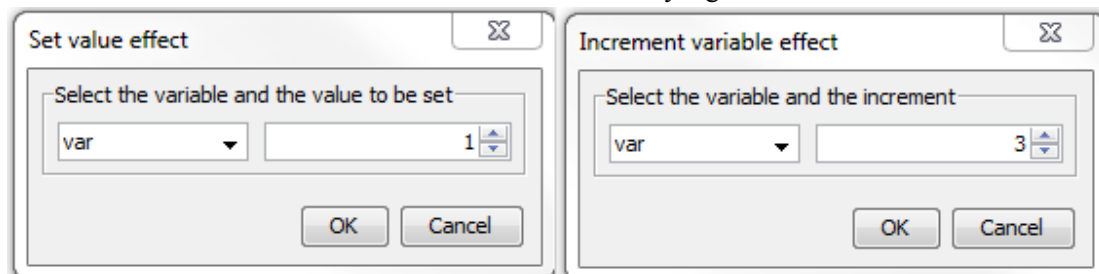


Figure 58. Effects to set the value of a variable or increase its value in 3 units

### 3.1.8. Other effects

There are many more effects in <e-Adventure> besides the ones used to modify the values of flags and variables (Figure 59). However, the effect edition panel provides documentation for each of them that is directly accessible to the user before adding them to the game.

Effect	Description	Usage
Play a sound	Plays a given sound file (only once)	Enhance attractiveness of games by adding sound to actions (e.g. steps echo when a character moves).
Play an animation	Plays a given animation (set of frames) in the position (x,y) of the scene selected	Enhance visual appearance of games. Can be used, for example, to include elements with special effects such as fades, which are not supported by the game editor.
Player speaks	The main character (i.e. player) will say the given dialog line	Useful to provide short feedback messages with little effort.
Character speaks	The character selected will say the given dialog line	
Show text	Displays the given phrase in a specific point of the scene for the specified time gap.	Similar to player/character speaks. Main difference is that it allows to select the place where the text will be printed in the scene, and the amount of time to be rendered (these parameters are automatically set in other effects).
Trigger a conversation	Starts the selected conversation	Allows to trigger a conversation that is not attached to any character as a “Talk to” action
Trigger a scene	Changes the current scene	Allows to make scene transitions without using exits
Trigger previous scene	Goes back to the previous scene displayed in the game (cutscenes are not taken into account). If current scene is the first, an error is prompted.	Useful for implementing menus (using scenes): this effect allows you to define elements that can behave as a “back” button.
Trigger cutscene	Plays the selected cutscene.	Allows to trigger cutscenes without using exits.
Trigger book	Opens the specified book	This is the only way to make a book appear on the screen. Typical usage: define an object to launch the book when it is examined (e.g. an object like a notebook or similar).
Consume an object	Removes an object from the inventory	Allows to make objects disappear from the inventory.

Generate an object	Puts an object in the inventory and removes it from the scene.	Allows to put an object in the player's inventory without defining a "Grab" action
Highlight an object	Highlights the specified object in red/green or blue. It also allows to animate the object. The object will remain highlighted until the scene is changed.	Very useful to guide the player's attention to interest places at some points of the game.
Move player	Makes the player walk from the current position to a specified destiny on the scene (x,y)	Adds dynamism to the games
Move a character	Makes the character walk from its current position to a specified destiny on the scene (x,y).	
Move an object (Interpolation)	Interpolates the object from its current position to a specified destiny on the scene (x,y). Object can also be scaled and animated. Interpolation speed is configurable. The object will remain at the destiny position until the scene changes.	
Launch a macro	Launches the selected macro	Allows for the creation of "set of effects" to be triggered together.
Cancel action	Prevents default effects to be executed in actions	For example, adding a "Cancel action" effect to a "Grab" action will avoid consuming the object from the scene.
Effect with probability	The next effect to be triggered will be chosen randomly between a list of two. The probability of each effect can be customized.	Allows for introducing simple random behaviours in the games.
Wait	Blocks the game for a given time gap	

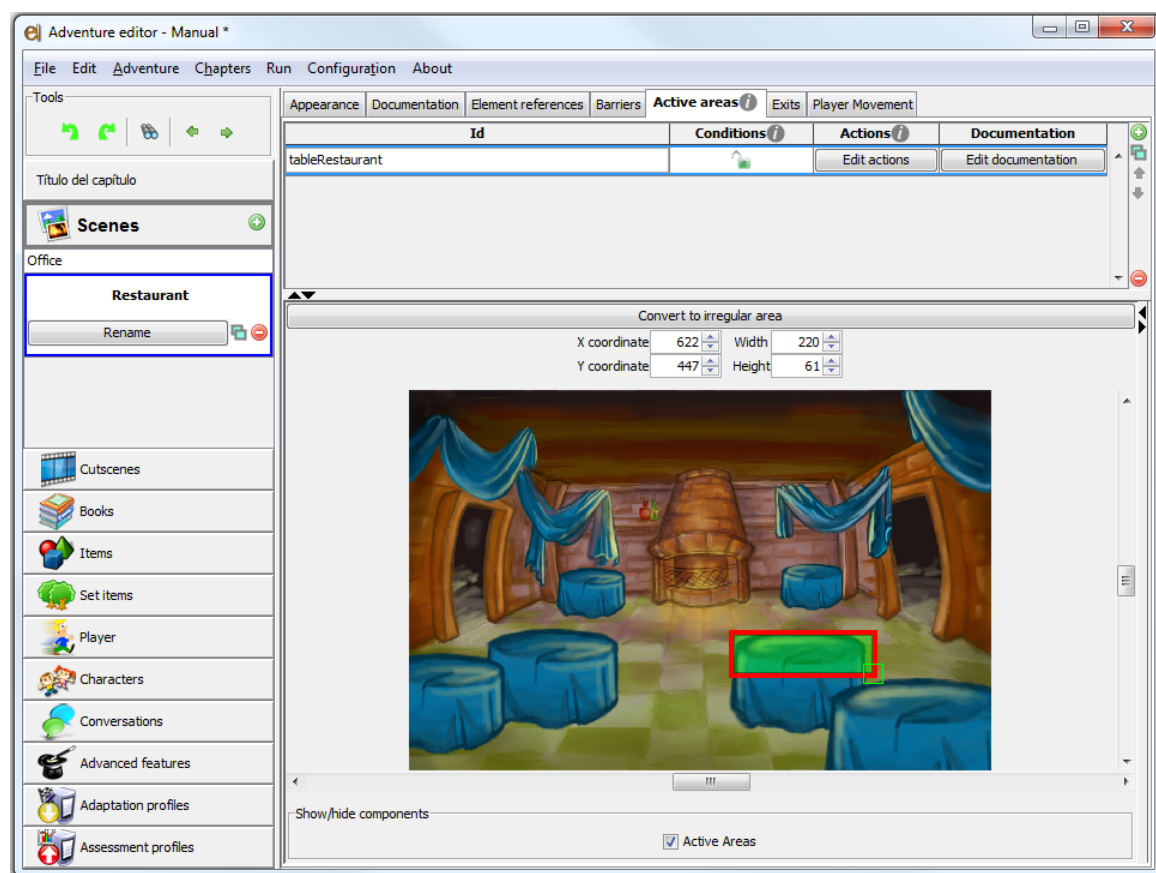
Figure 59. Table with all the effects available in &lt;e-Adventure&gt;

### 3.1.9. EXAMPLE: Clarification about effects "consume object" and "generate object". Grab objects and place them in different scenes.

In simulations and adventure games is interesting that the player can grab objects and store them in the inventory to use them by themselves or in combination with other interactive elements in different parts of the game. You can define these behaviors in <e-Adventure> with effects "generate" and "consume"

object”. When an object is in a scene and a “generate object” is applied over it, the object disappears in all the scenes that is referenced in the current chapter and appears in the inventory allowing to simulate that the object has been picked up. The “Grab” action performs this behavior by default. The effect “consume object” extracts the object from the inventory preventing to perform more actions over it.

It may happen that you want to show an object that has been picked up in other scenes of chapter. This behavior can’t be implemented using a single object due to once the object is grabbed won’t be showed again in the chapter. Let see an example. The main objective is to collect the book created in section 2.3.1. in the “Office” scene and place it on the table that is in the scene “Restaurant”. First you have to create an active area named “tableRestaurant” in the scene “Restaurant” (active areas are parts of the scene that can be interactive, more information in section 3.6) boxed one of the tables that already appears (Figure 1).



**Figure 60. EXAMPLE: Pick up and place objects in different scenes. Adding to active areas to the scene "Restaurant."**

You have to add a “Grab” and “Use with...” action to “Book” object. The action “Use with...” have to include a “consume object” for “Book” and an “activate flag” for “bookOnTable” (Figure 1).



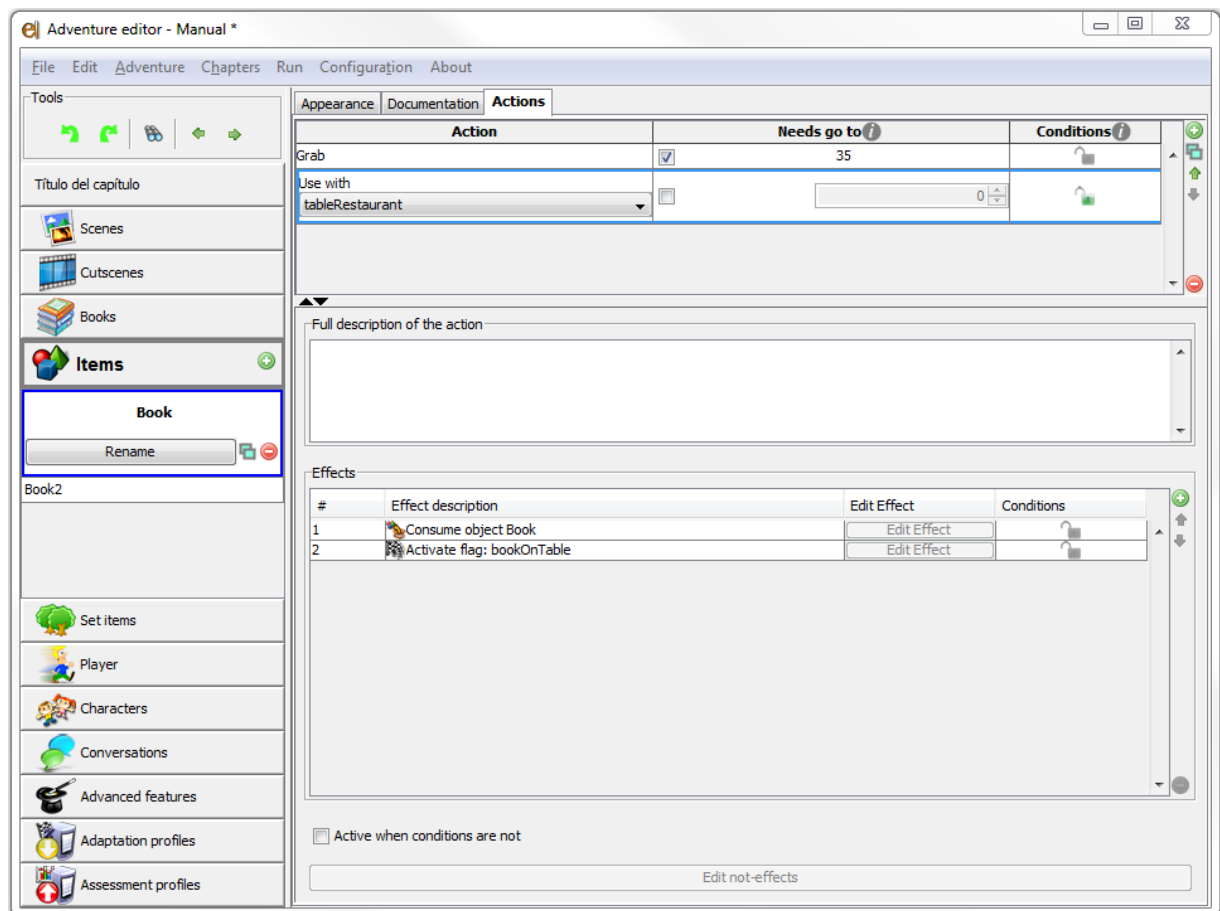


Figure 61. EXAMPLE: Pick up and place objects in different scenes. Adding effects to “Book” object.

Finally, you have to create other object (with name “Book2”) with the same image as “Book” object to represent the book that will be placed once it is in the inventory. Add the “Book2” in the “Restaurant” scene as in the section 2.3.4 but adding a condition in to the reference in order to hide it until the “Book” will be used with the “tableRestaurant” (more information about how define conditions and modify the game state in section 3.1). It means, you create a condition that will be met when the flag “bookOnTable” will be activated.

### 3.1.10. Macros

A *macro* is just a group of effects that is given a unique identifier. This allows launching effect blocks in different parts of the game without duplicating the block.

To add a macro, click on the element “**Advanced features**” in the structure panel, and then select tab “**Macros**”. Click on the add button (+) on the right. After introducing the Macro ID (must be unique) press the “OK” button. Effects can be added, deleted or ordered using control buttons on the panel titled “List of effects of the macro”. Only restriction is that a macro cannot contain an effect “Launch macro” to itself.

To launch all the effects in a macro, just add a “Launch Macro” effect at any point of the game.



## 3.2. Organization of the element references in the scene

Section 3.2 covers different features provided in the “Element references” tab of the scenes. This tab is divided in two vertical sections: a reference list (top) and a preview panel (bottom). Sections 3.2.1, 3.2.2 and 3.2.3 cover settings of the reference list. Section 3.2.4 addresses the preview panel.

### 3.2.1. Layers

The order in which elements referenced in a scene (characters, items and set items) are drawn can be configured. This feature is important to represent depth (z coordinate) in 2D scenes like those present in <e-Adventure> games.

A layer is the order in which an element will be drawn in the scene. Element in layer 0 is the first to be drawn, and therefore will be perceived as the farthest from the observer (player).

To change the layer of each element reference, go to the scene and click on the tab “**Element references**”. The layer of each element is shown in the first column of the list of references. You can alter the layer of an element by clicking on buttons up (  ) and down (  ) on the right (see Figure 62).

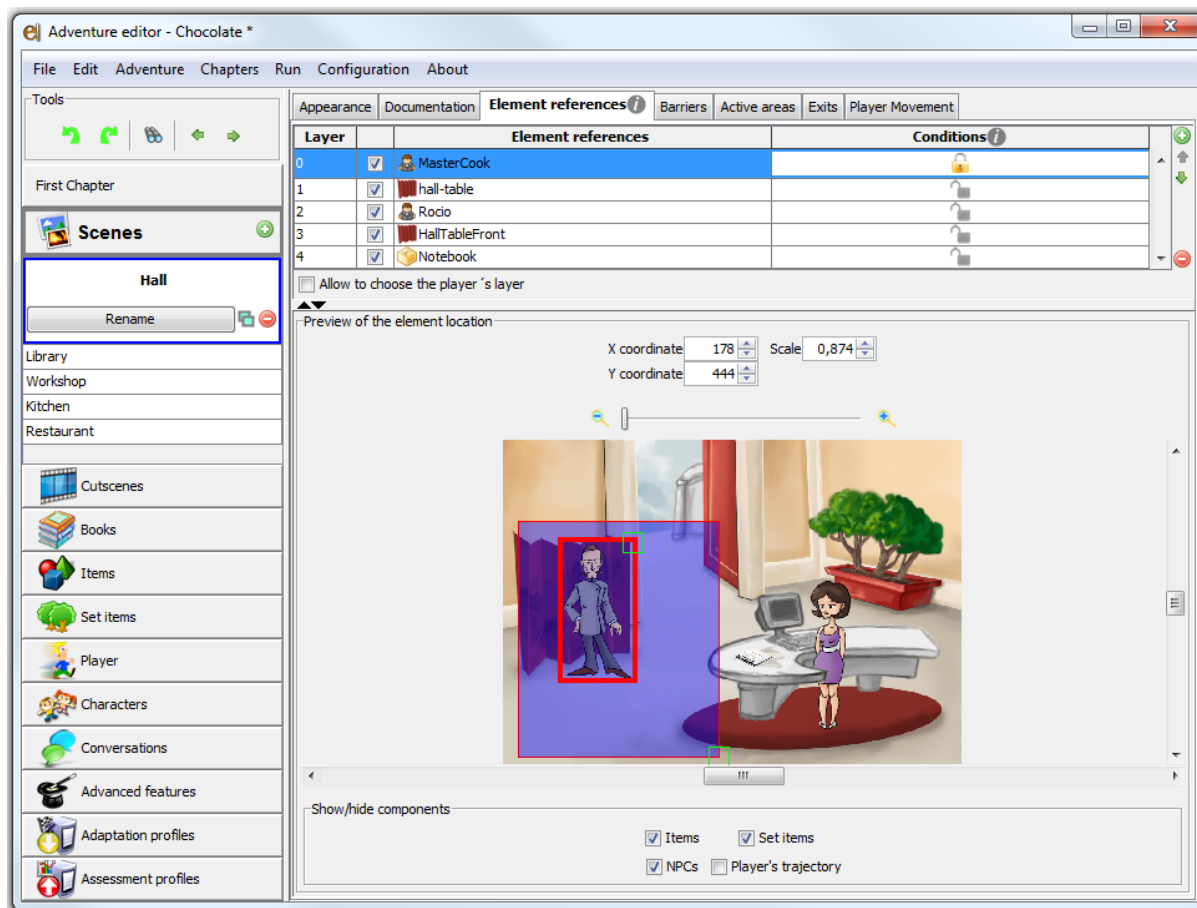


Figure 62. Edition of the layer of element references

The layer of the player is treated in a singular manner. Since the player can move in the scene, the <e-Adventure> engine will assign its layer automatically depending on how far is estimated to be from the observer according to its current position and trajectory. To determine whether the player or an element

must to be drawn closer to the observer, coordinates y of both are compared (the lower y is considered to be in the front).

For example, this allows for getting the player drawn before an element when it is supposed to be farther. In Figure 63 we can see that when the player enters the scene it is behind the table, but after moving towards the bottom of the scene the player is drawn in front.

However, the player's layer can also be set manually just as any other element in the scene by activating the option **"Allow to choose the player's layer"**.

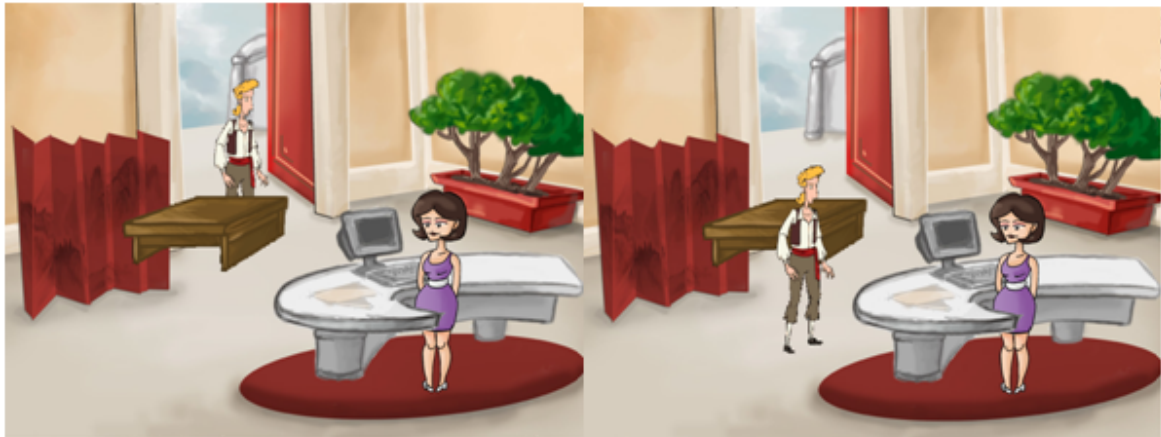


Figure 63. Example of using layers to adjust depth in the scene

### 3.2.2. Conditions

When an element reference is selected in the "Element references" tab (either using the preview panel on the bottom or the reference list on the top) there is a button that allows the edition of conditions. This allows for setting restrictions on the visibility of each element reference by using flags, variables or game states (as explained in section 3.1.1). Elements can appear or disappear in the scene depending on the conditions established.

### 3.2.3. References list

In this panel all the element references and the player are displayed. They are ordered according to their layer. References can be added or deleted using add (+) and delete (-) buttons on the right side of this panel. Although this process is straightforward, for more information on how to add each type of element see sections 2.3.4 (items), 2.4.3 (set items) and 2.5.3 (characters).

In addition up (↑) and down (↓) buttons allow to modify the layer of any element reference.

### 3.2.4. Elements preview

This sub-panel provides a preview of how the elements will look in the scene. In addition, this panel allows modifying the position and scale of each element reference. To modify an element reference, just click on the element and then drag the green square on the upper right corner to vary its size (scale), or drag the element itself to place the element in a different position.

On the bottom of this panel controls to hide/show elements are provided (“**Items**”, “**Characters**”, etc.). We can hide groups of elements by just deactivating the appropriate category (see Figure 62). Hiding a group of elements will only affect the preview on the game editor (all element references will always be drawn in the game).

### 3.3. Export as a Learning Object (LO)

One of the most relevant features of the <e-Adventure> platform in educational contexts is its ability to export games as Learning Objects (LO). Learning Objects are self content packages that can be deployed in Virtual Learning Environments (VLE). This allows the games to be easily added to educational courses just as HTML documents and be played directly in a web-browser. VLE must support one or more packaging and communications standards to be able to include <e-Adventure> LO. In order to increase compatibility several different standards are supported in the platform, in the way of different export profiles:

- IMS-CP: IMS Content Packaging is used to describe and package educational contents to be added in content repositories or VLE.
- WebCT 4: this content packaging format is used in the platform of the same name. This is not a standard but rather a specific export profile, and games exported for this platform are not supported in others.
- SCORM: SCORM is a set of standards and specifications created by US Defence Department, that includes packaging and communication features. This is one of the most popular specifications nowadays. <e-Adventure> supports two versions: SCORM v1.2 and SCORM 2004
- AGREGA: this is a Spanish content repository, financed by the Science, Education and Industry Ministries. Games exported following this profile are compatible with this specific repository, so this profile is neither a standard nor specification.
- LAMS (Learning Activity Management System): LAMS is a educational management system that allows for the creation of on-line learning sequences (<http://www.lamsinternational.com/>). Support for this system is expected by December 2010.

Besides exporting games as LO, a series of meta-data can be attached to the games that will be later used by the web server to index and retrieve information. The meta-data need depends on the exportation profile, so the format must be chosen accordingly: LTSC-LOM (IMS-CP, SCORM 2004 and LAMS), IMS-LOM (SCORM 1.2) and LOM-ES (AGREGA). No meta-data is needed for WebCT.

To edit the meta-data, a specific editor is available at the “*File*” > “*LOM Metadata editor*” submenu. For instance, if we choose to edit the “Learning Object metadata” we will edit the LTSC-LOM data in the editor shown in Figure 64.

The screenshot shows a window titled "Learning Object Properties" with a close button (X) in the top right corner. The window has three tabs: "General", "Life cycle & Technical", and "Educational". The "General" tab is selected. It contains four input fields: "Title" with the text "Title of the Learning Object", "Language" with a dropdown menu showing "English", "Description" with a text area containing "Learning Object's description.", and "Keyword" with a text field containing "<e-Adventure>, tutorial". Below these fields is a "Set defaults" button. At the bottom of the window is an "OK" button.

Figure 64. LTSC-LOM meta-data editor for Learning Objects

This editor allows the following data to be changed:

- “General” tab
  - *Title*: the title of the game
  - *Language*: the language of the game
  - *Description*: description of the game content
  - *Keywords*: keywords used for indexing purposes
- “Lifecycle and Technical” tab
  - *Version*: the version of the game
  - *Minimum version*: non-editable value, indicating the minimum compatible version of the <e-Adventure> engine
  - *Maximum version*: non-editable value, indicating the maximum version needed to run the game.
- “Educational” tab
  - *Intended user rol*: This is the rol of the target user of the game. It can take different values (Teacher, Author, Learner, Manger).

- *Semantic density*: This is a measure of the density of information in the game content (Very low, low, medium, high, very high).
- *Learning resource type*: The type of resource represented by the game (Exercise, Simulation, Exam, Self assessment, Lecture).
- *Context*: The context where the game is applicable (School, Higher education, Training, Other).
- *Difficulty*: The difficulty of the game (Very easy, easy, medium, difficult, very difficult).
- *Interactivity level*: The level of interactivity of the game, ranging from very low to very high.
- *Interactivity type*: The type of the interactivity present in the game (Active, expositive, mixed).
- *Description*: Description of the educational aims of the game.
- *Typical age range*: The age range of the target users (e.g. 16-25).
- *Typical learning time*: The median time needed to complete the game.

It must be noted that not all values are required to export the game.

The other editors (“Learning Object properties for SCORM 1.2, IMS-LOM” and “LOM-ES properties”) are very similar to this one, but include meta-data of the IMS-LOM standards (a specification on with IEEE based LSTC-LOM) and LOM-ES (a particularization to the Spanish educational system). This editors change some words and some fields. In the case of LOM-ES, all fields have default values given that they are all required.

After editing the meta-data of the game, the game can be exported as a LO. The LO will be a ZIP file that will be saved in our folder of choice (Exports folder, by default). To do this we use the “File” > “Export as Learning Object” menu. A dialog will allow us to select the packaging we want for our game. For a SCORM export, the adaptation and evaluation profiles should fit those specifications or the game will not be exported. The same is true for AGREGA, requiring SCORM 2004 profiles. More information regarding this is to be found in section 3.5.

Other fields in the export dialog are the name of the Learning Object, of the author and of the organization, and a checkbox that defines whether the game will run within the web-browser or in a separate window. The author and organization name will be displayed in a security policy panel when running the game and should be 6 characters or longer.

After clicking OK we will have to select the path of the LO. After the export process is finished, the game can be uploaded to a VLE. For instance, Moodle and Ilias3 are compatible with IMS-CP, SCORM 1.2 and SCORM 2004 objects. AGREGA packages can be attached to the Spanish repository, but are also compatible with any SCORM 2004 VLE.

### 3.4. Export games as JAR (runnable Java programs)

Exporting games as JAR files allows the games to be played directly on any computer, without having to install <e-Adventure> or opening a web-browser. To run games in this mode requires the JRE (Java Runtime Environment) in version 1.6.0\_18 at least. This way games can be easily distributed.

To export games this way, we use the “File” > “Export” > “Export as runnable JAR” menu. This will ask us for a folder and file name. The export will take some time to complete.

### 3.5. Adaptation and evaluation profiles

Following current educational trends, <e-Adventure> support both the assessment (evaluation) of student performance and the adaptation of game contents to fit the student needs. These features are configured through Adaptation and Evaluation profiles. Each profile is made up of a set of rules (conditions of flags and variables). When the conditions are met, rules are applied (to assess student behavior or modify the game, respectably).

The use of profiles allows several different ways to adapt or assess the games to be chosen when exporting or running the game. This increases the reusability of games, making them usable in different context or settings.

An adaptation and evaluation profile can be selected for each chapter. To edit the adaptation profiles or evaluation profiles, “Assessment profiles” or “Adaptation profiles” must be selected in the structure panel.

#### 3.5.1. Assessment profiles

An assessment profile is created just like any other game element, by using the + button next to the “Assessment profiles” selection in the panel to the left.

The edition panel for an assessment profile (Figure 65) has a section at the top that allows the selection of the type of profile (Normal profile, SCORM 2004 or SCORM 1.2). Besides this, the way the profile is used can be configured by selecting the “Show assessment report when chapter is over” option. If the report will be shown to the player, it can also be sent by e-mail to the student after he/she gives consent.



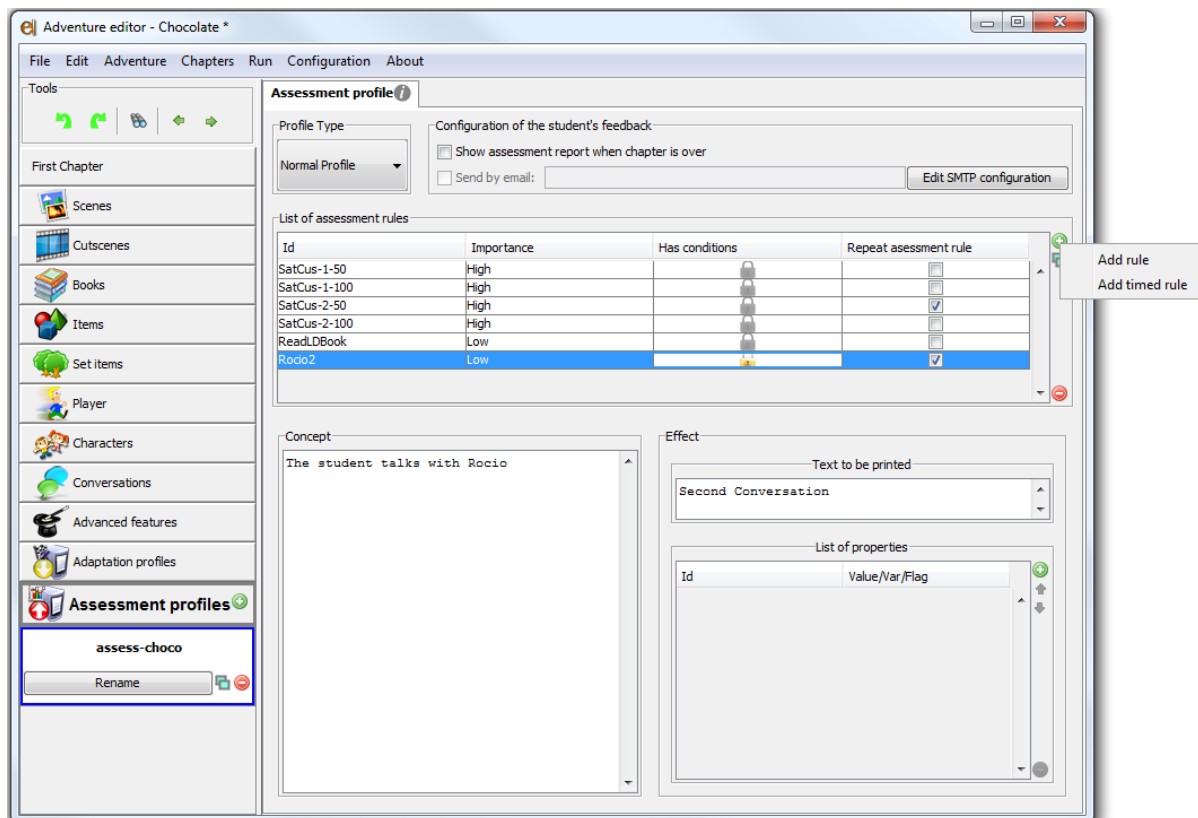


Figure 65. Assessment profile edition panel

For the email system to work, the game creator must configure the SMTP service for a valid email account. This can be done for Gmail, Hotmail, Yahoo or any private provider. The necessary data is either publicly available or must be asked to the system administrator.

Changing the profile type will change the data model used, changing the variables that can be modified as a consequence of the rules. New rules can be added with the add button (+) next to the list of rules. There are two different kinds of rules.

### Regular rules

A regular assessment rule (Figure 66) has the following fields:

- **Id**: the id of the rule. This must be a unique (within the profile) and valid.
- **Importance**: “Normal” by default, the importance can range from “Very low” upto “Very high”.
- **Conditions**: This are the set of conditions that must be valid for the rule to be triggered. This will have effect as soon as the conditions are changed and is used to trigger a rule, for instance, after the student correctly answers a particular question.
- **Repeat assessment rule**: This option allows for the assessment rule to be repeated (the consequences triggered once again) every time the conditions for the rule become valid once again (it should be made invalid first). This should be used carefully, as it might produce to much information in the report or the VLE to be useful.
- **Effect**: the effects to be triggered with the rule. This is made up of the concept, text and properties fields.
  - *Concept*: A brief description of the rule’s meaning.

- *Text*: The text to be shown in the report about what triggered the rule.
- *Properties*: these are variable-value pairs. When the rule is triggered, the variables will be given the value selected in the corresponding property. This will not be reflected in the report and is used to send information directly to the server (or VLE). The ids of the variables will be conditioned by the selected type of the profile.

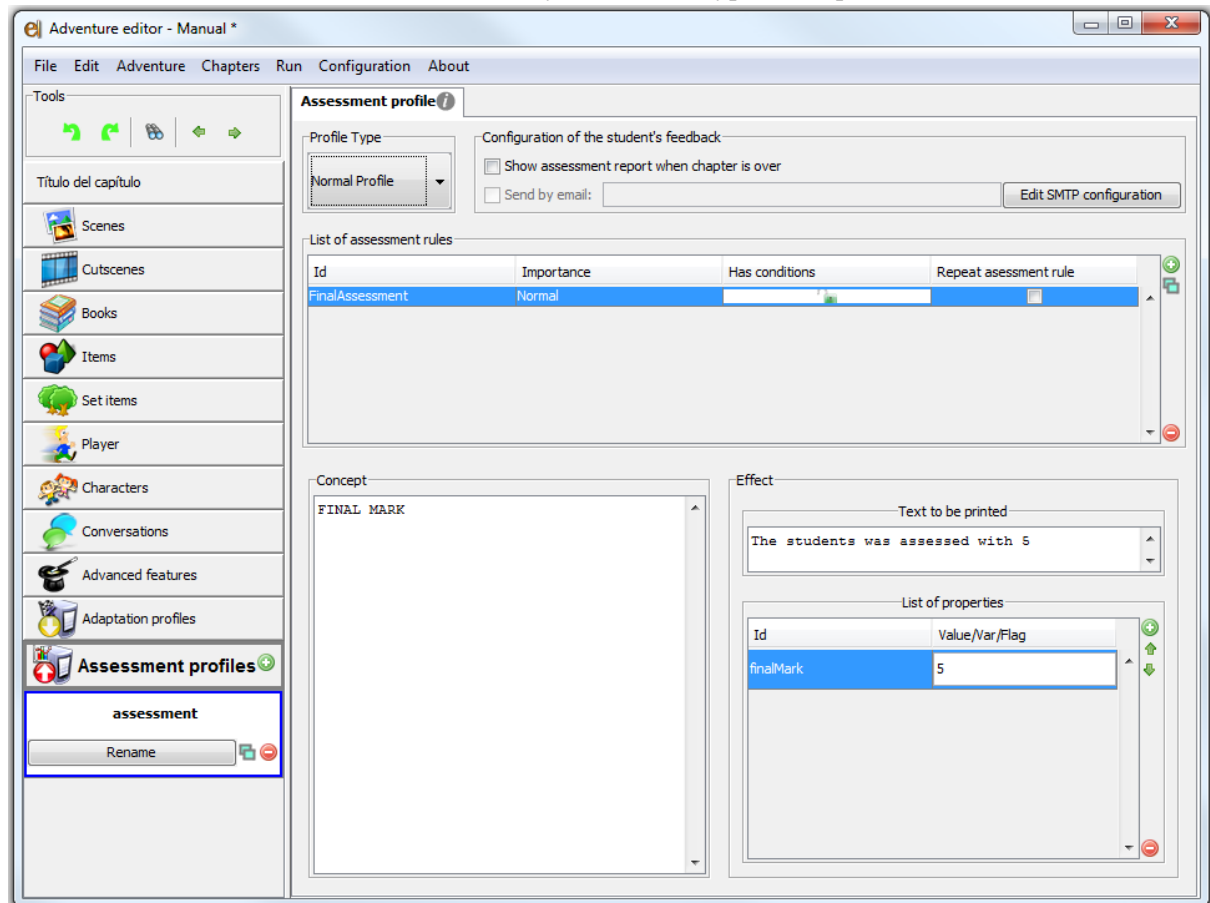


Figure 66. Edition of a regular assessment rule

## Timed rules

This sort of rules triggers a timer to start when the conditions are met. The timer will stop when the finishing conditions are reached. The time between these events is measured and used to produce different assessments of the students depending on the range it falls into.

Multiple rule effects (as defined for regular rules) can be defined, one for each range (Figure 67). When the time falls within a range, its effects are triggered producing information to be written in the report and information being sent to the VLE. If the time falls outside all ranged, the rule produces no effects. If it falls within two different ranges, only the first one will be valid.

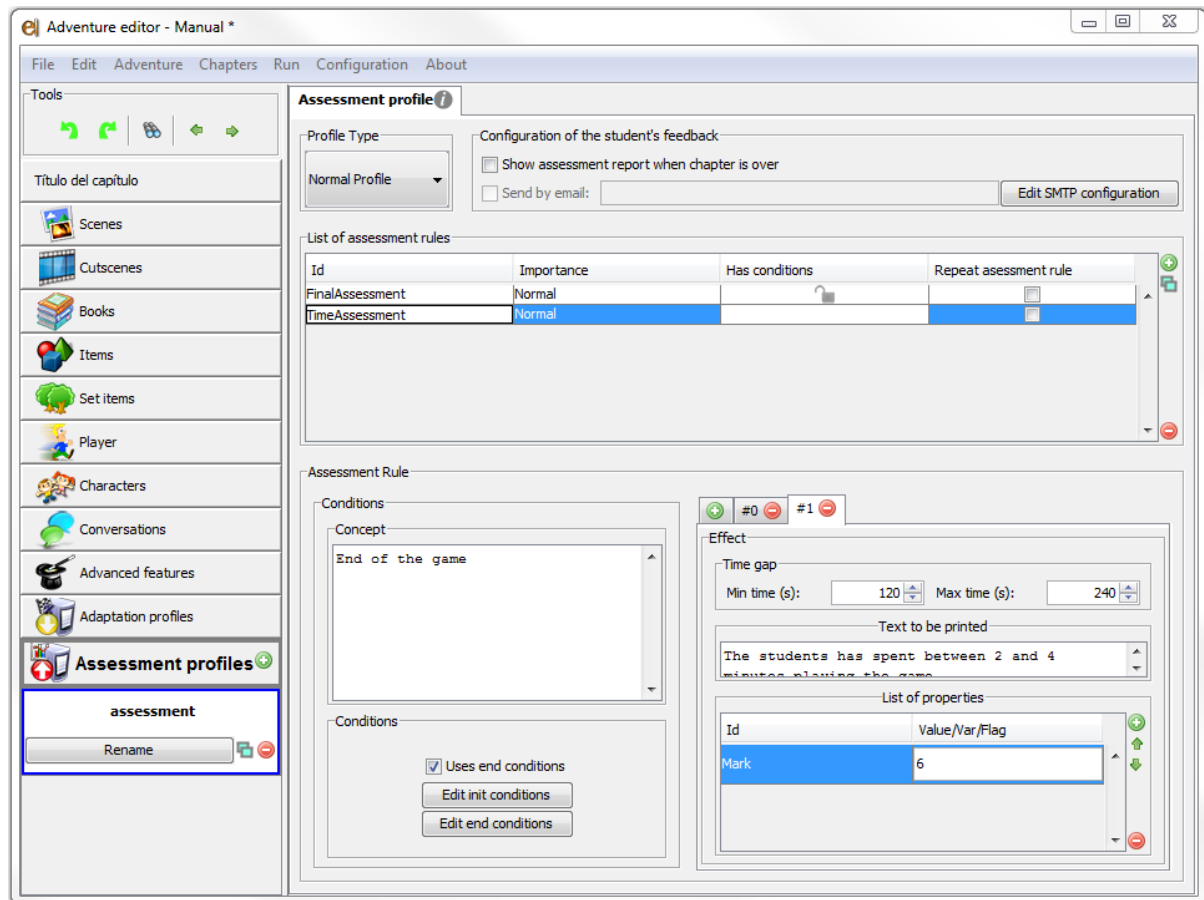


Figure 67. Edition of a timed assessment rule

## Report

The rules will produce an XML (for a VLE server) or HTML report (for the student or teacher). For example, an HTML might look like the one in Figure 68.

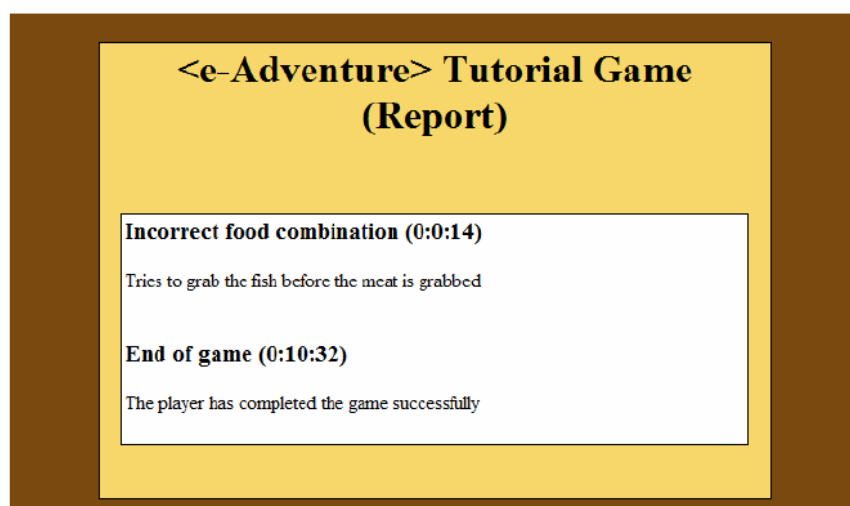


Figure 68. A sample assessment report.

### 3.5.2. Adaptation profiles

Adaptation profiles are created in a way analogous to how assessment profiles are generated. These profiles are used to adapt the behavior or content of the game based on a series of “external properties” provided by a Virtual Learning Environment (VLE). When a set of properties has the value specified by the rules at the beginning of the chapter, flags, variables or even the initial scene of the chapter are modified. This allows for the number of elements in the game to be changed, the difficulty adjusted, etc. Just like with assessment profiles, the type of the adaptation profile can be SCORM 1.2 or SCORM 2004. The ids of the variables will be fixed according to the data model of the selected type.

The definition of adaptation rules includes a description and the initial scene of the chapter for that rule (Figure 69). This information is edited directly in the table of adaptation rules. In the lower part of the panel there are two tabs. The “LMS state” tab allows for the edition of the conditions that must be met by the variables in the LMS for the rule to be triggered (Figure 70). The “Game state” tab (Figure 71) is used to modify the values of flags and variables in the game when the rule is triggered. These modifications are done at the beginning of the chapter.

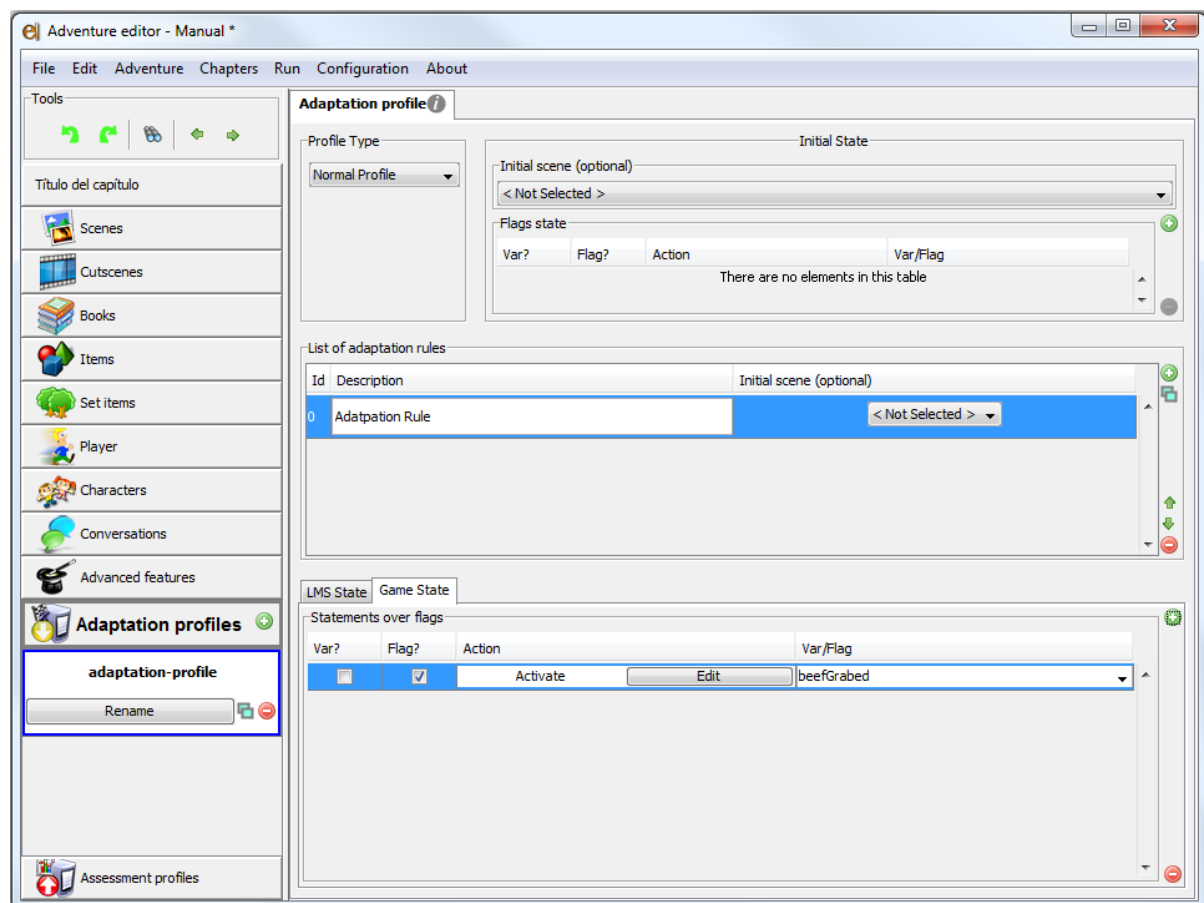


Figure 69. Adaptation profile edition panel after adding a new rule

Id	Op	Value
Student	=	Beginner

Figure 70. Adaptation rule edition panel, “LMS State” tab.

Var?	Flag?	Action	Var/Flag
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Activate	beefGraded

Figure 71. Adaptation rule edition panel, “Game state” tab

Besides, the adaptation profile provides a mechanism to modify the initial scene and variables whenever it is selected for the chapter (Figure 72). This can be used to change the initial scene and values of flags and variables without a connection with an LMS. This can be used to provide off-line adaptation of games, by choosing a different profile for each different version of the game.

Adaptation profile *i*

Profile Type: Normal Profile

Initial State

Initial scene (optional): < Not Selected >

Flags state

Var?	Flag?	Action	Var/Flag
There are no elements in this table			

Figure 72. Static initial state edition panel in an adaptation profile

### 3.5.3. Choosing the active profiles of the chapter

Once the profiles are created, it is important for them to be selected for the chapter. This is modified in the main component of the structure panel, the “Chapter” edition panel (Figure 73).

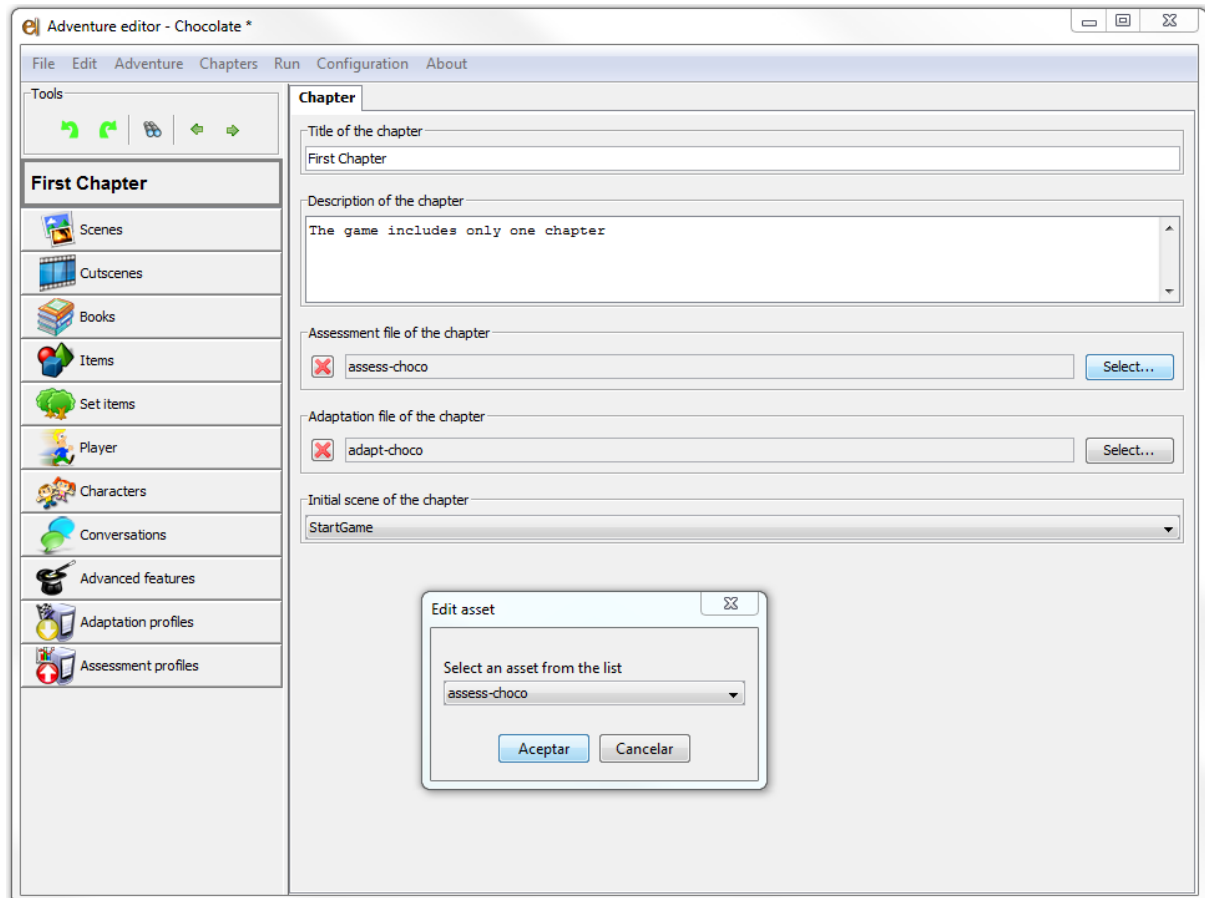


Figure 73. Choosing the adaptation and assessment profiles in the chapter edition panel

### 3.6. Active areas

Active areas are rectangles or polygons defined as part of the scene that define a part of the background that can be interacted with. These are used to provide interactivity with background rich in elements and are especially relevant in photo-realistic games.

An active area behaves much like an item, but it belongs to the scene and is not created independently (Figure 74). They are defined in the same way that exits where, and can be of different sizes and in different positions.

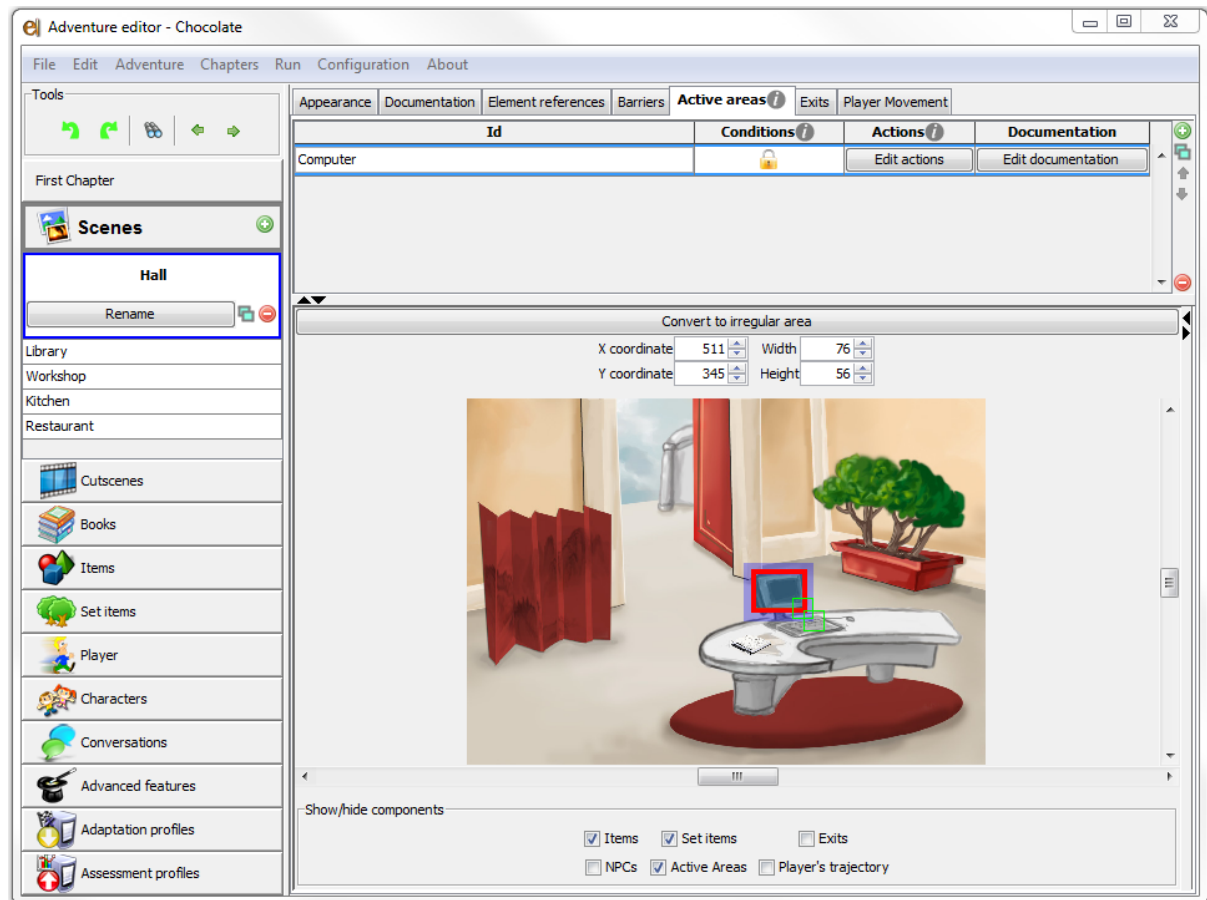


Figure 74. “Active areas” tab in the scene edition panel

### 3.7. Barriers

Barriers are only available in third person games.

Barriers, just like exits or active areas, are rectangular areas or polygons defined on a scene. Barriers provided the ability to limit the players movement (other characters in the game are not constrained by barriers). Conditions can be defined as to when the barrier is active and when it is not. When a barrier is active, the player will stop just before it until the conditions are not met anymore.

These elements are edited just like any other, in the scene edition panel, in the “Barriers” tab (Figure 75).

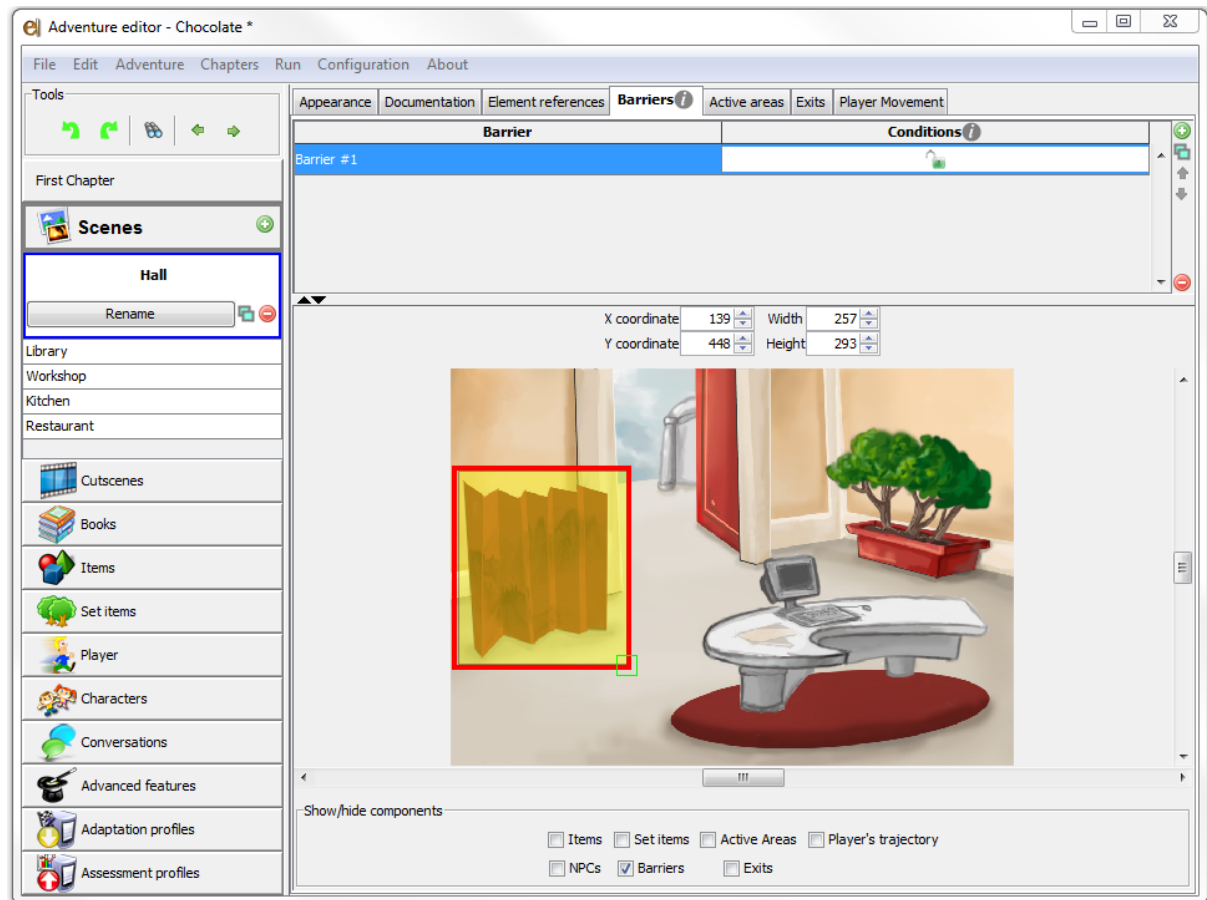


Figure 75. “Barrier” edition tab, in the scene edition panel

### 3.8. Player movement

The player movement tab and trajectories are only available in third person games.

The way the player moves in a scene can be configured in two different ways: setting an initial position or using trajectories. When an initial position is used, which is the easier way to do it, the player will only be able to move along the horizontal line that goes through that point. Besides, the player will start (by default) in that position. This needs no further configuration and all objects can be reached when near their X coordinate.

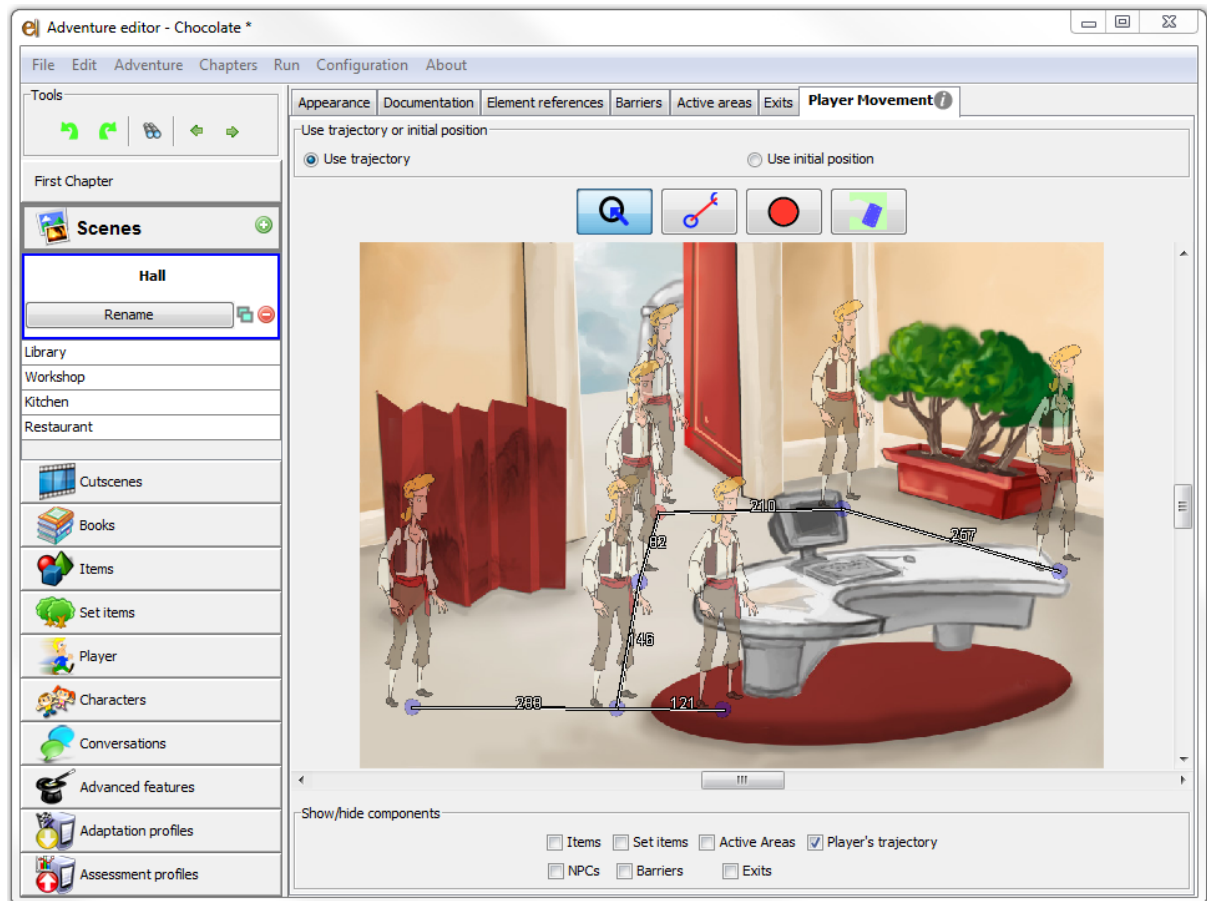
The other option, using trajectories, requires the creator to explicitly draw all the paths for the player. This is done by first defining the nodes or points of interest along these lines and the lines from one to another. There is no limit in the number of nodes, lines or loops, but it is strongly advised that they are kept to a minimum for efficiency sake.

When editing the trajectory of a scene, there are several tools at our disposal (Figure 76). These tools are over the preview of the scene. They are (in order):

- Edit nodes: this tool is used to select, move and scale nodes, as well as creating new ones.
- Edit sides: this tool is used to add new sides (paths from one node to another) to the trajectory. Sides are created clicking one node and then another.



- Select initial node: this tool allows for the selection of the initial node in the trajectory. This is the node where the player will appear in the scene.
- Delete: this tool is used to delete any node or side by clicking on it.



**Figure 76. “Player movement” tab in the scene edition panel**

The scale of the nodes is used to create a sense of depth in the scene, making the player seem bigger or smaller depending on the current node. If there is a path connecting two nodes with different scales, the player will be drawn with a scale equaled to the pondered mean at every point. The speed with which the player travels along these sides can also be modified by changing the length of the side (the number that appears over the line).

Barriers only affect the trajectory if they directly cut a path. Objects can only be interacted with if their influence area intersects the path. The influence area is edited just like other scene elements. Influence areas are also applied to exits and active areas. Influence areas can only be edited while the element is selected in the “Element references” tab (Figure 77). Active areas appear in blue.

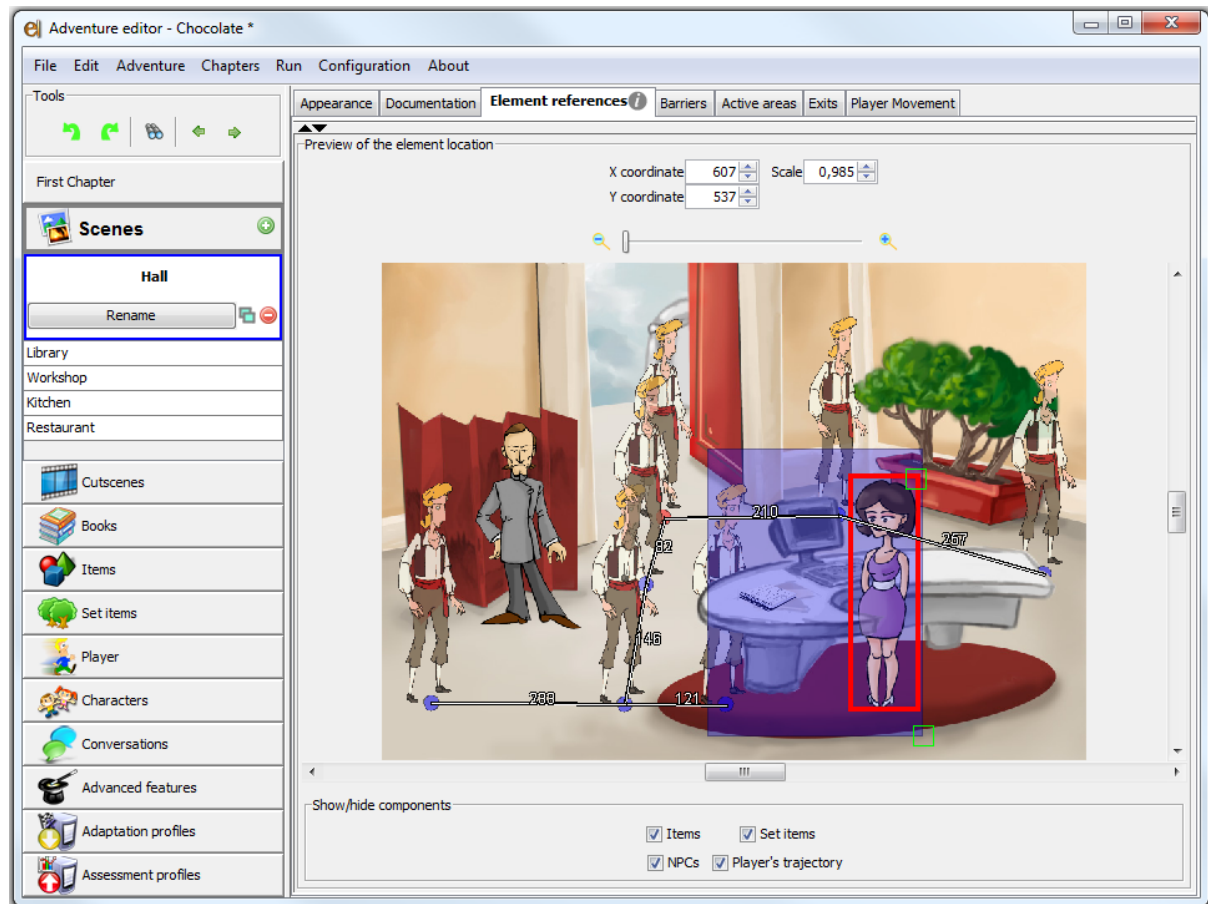


Figure 77. Editing the influence area of an element reference in the “Element references” tab

### 3.9. Timers

Timers allow triggering blocks of effects periodically or after a specific amount of time has elapsed.

To configure timers, click on “**Advanced features**” on the structure panel. The first tab displayed on the right panel will be the timer list.

Timers can be added or deleted using add (+) and delete (-) buttons on the right side of the timer list (which by default is empty).

The basic idea of how a timer works is simple. When the defined set of conditions are met, then the timer will start counting until a time limit is reached. Then, a block of effects will be triggered.

Nonetheless, timers can be configured to tweak this behavior in several ways. First, timers can be forced to stop by defining a set of “End conditions”. When these conditions are met then the timer will abort the count (if the time limit has not been reached yet). In addition, another set of effects can be triggered when the timer is forced to stop.

Second, it can be configured if the timer should remain active during the whole game or just once in a couple of ways:

- It can be defined if the timer should start multiple times or not. If this option is selected, then the timer will start counting again if the initial conditions are met at any time after first finalization.

As a consequence, if a timer with this option activated expires because the time limit defined has been reached, the timer will start again if the initial conditions are satisfied. This behaviour allows reusing the timer in different parts of the game.

- The timer can be defined to run in loops. When a timer run in loops, it will restart if the time limit is reached, until end conditions are met.

The full list of parameters that can be configured for a timer is as follows:

- **Time** (second column of timer list): this is the time limit. When reached, the timer will trigger the effects block and restart or end depending of the configuration.
- **Display in game** (third column of timer list): if active, the elapsed time will be rendered on the upper right corner of the screen. Other parameters will be available then on the panel below the list:
  - Display name: Text that will be displayed along with the elapsed time.
  - Count-down: If active, the timer will count backwards and trigger the effects when zero is reached.
  - Show when stopped: If active, the text and time elapsed will be displayed on the screen even if it has not started counting or if it has been stopped.
- **Documentation**: Text introduced here will not be used in the games (only valid for internal documentation).
- **Loop control**
  - Multiple starts: defines if the timer to start once or multiple times.
  - Runs in loops: defines if the timer has to start counting again after reaching the time limit or not.
- **Conditions to start the timer**: the timer will start counting when these conditions are met.
- **Conditions to stop the timer**: the timer will stop (even if the time limit has not been reached) when these conditions are met.

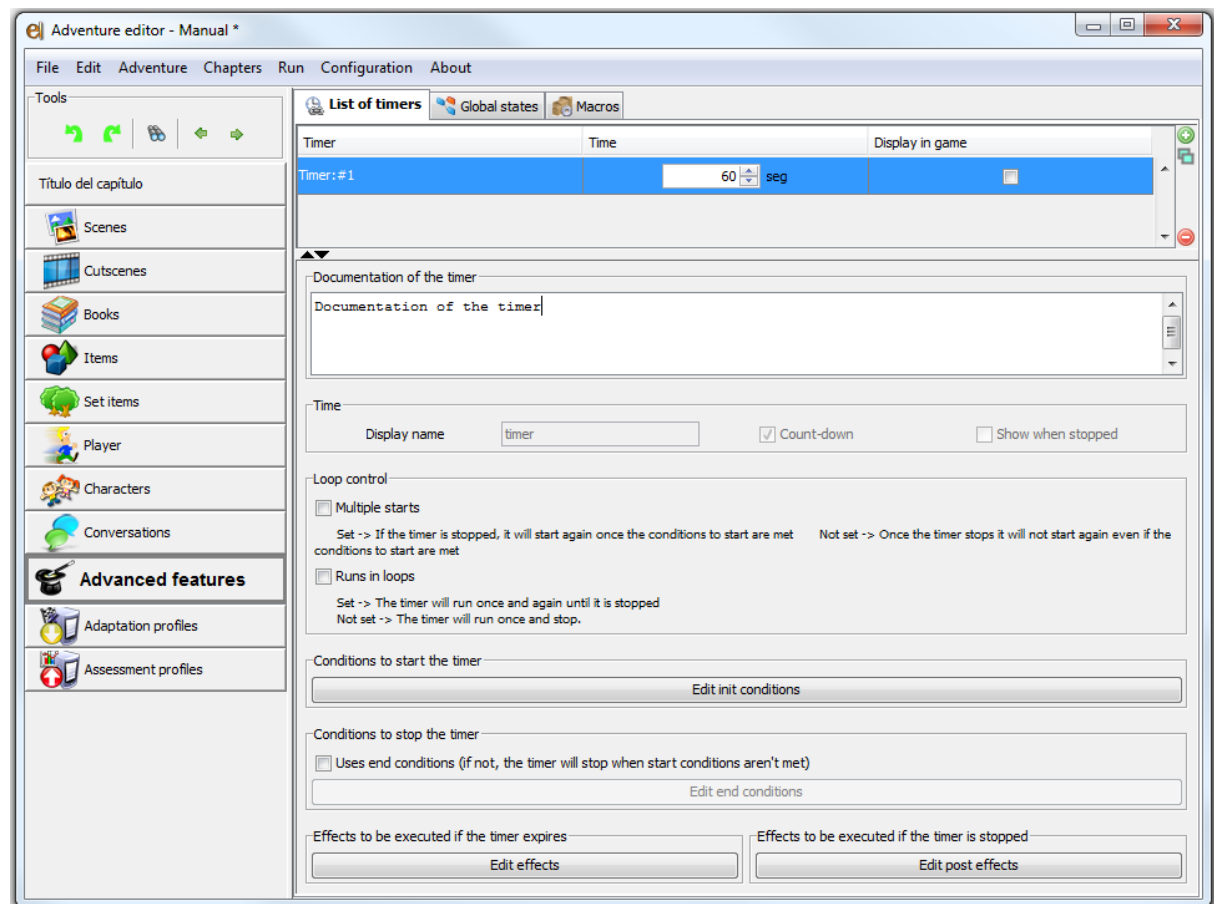


Figure 78. Edition of advanced features, “Timers” tab

### 3.10. Custom actions

In section 2.3.2 actions were introduced. Those were predefined actions, which have specific default behavior. For example, a “Grab” action is always going to remove an object from the scene and add it to the inventory.

Custom actions allow defining interactions in the game with no specific default behavior, extending the expressiveness of the <e-Adventure> platform to suit any specific needs.

There are two types of custom actions depending on the number of game elements involved. If the action is unary (the action is executed over the own element), like “eating”, “throwing” or “turning on” it will be called simply an **action**. If the action is binary (the action uses the first element but is executed on a second target element), like “combining element1 with element2” or “giving element1 to character2”, it will be considered an **interaction**.

The drawback is that custom actions require setting the images for the buttons that will use the player to trigger the action. For default actions <e-Adventure> sets predefined buttons (e.g. a “hand button” for use, grab or use-with actions), although these can also be customized (see section 4.3.3.2). However, as the platform cannot foresee the meaning of custom actions the author of the game will compulsorily have to define them.

In addition, it may also be required to define a custom animation of the player executing the action. For default actions <e-Adventure> uses animations defined for the player, but these may be inaccurate for custom actions. However, defining the player animation is optional: if not set, player's "use animation" will be applied. In first person games, this step is not required.

Two images are required for an action button. The first will be used to represent the button in normal state. The second will be used to represent the button when the mouse is over (Figure 79).



Figure 79. Example of normal and over images for a custom action button

To add a custom action click on the add button (+) on the right side of the action list (actions tab). Within the contextual menu that will appear, option "Custom action" must be selected.

Then the editor will ask for a name. Use the dialog to introduce a representative name for the action (take into account that this will be the text used in the game for this action). After clicking on the "OK" button the system will ask if the custom action is unary (action) or binary (interaction).

If interaction is selected, the system will also ask which is the target element involved in the action.

A new custom action will be added to the list. The name and target element (in case it is an interaction) can be edited, although it is not possible to change the type of the action. Below the action list an edition panel will appear. This panel contains two different tabs: "**Personalization**" and "**Configuration**" (Figure 80). Use the "Personalization" tab to select the images for the action button and animations of the player (optional). Use the "Configuration" tab to edit the effects of the action (see section 2.3.2 for more details).

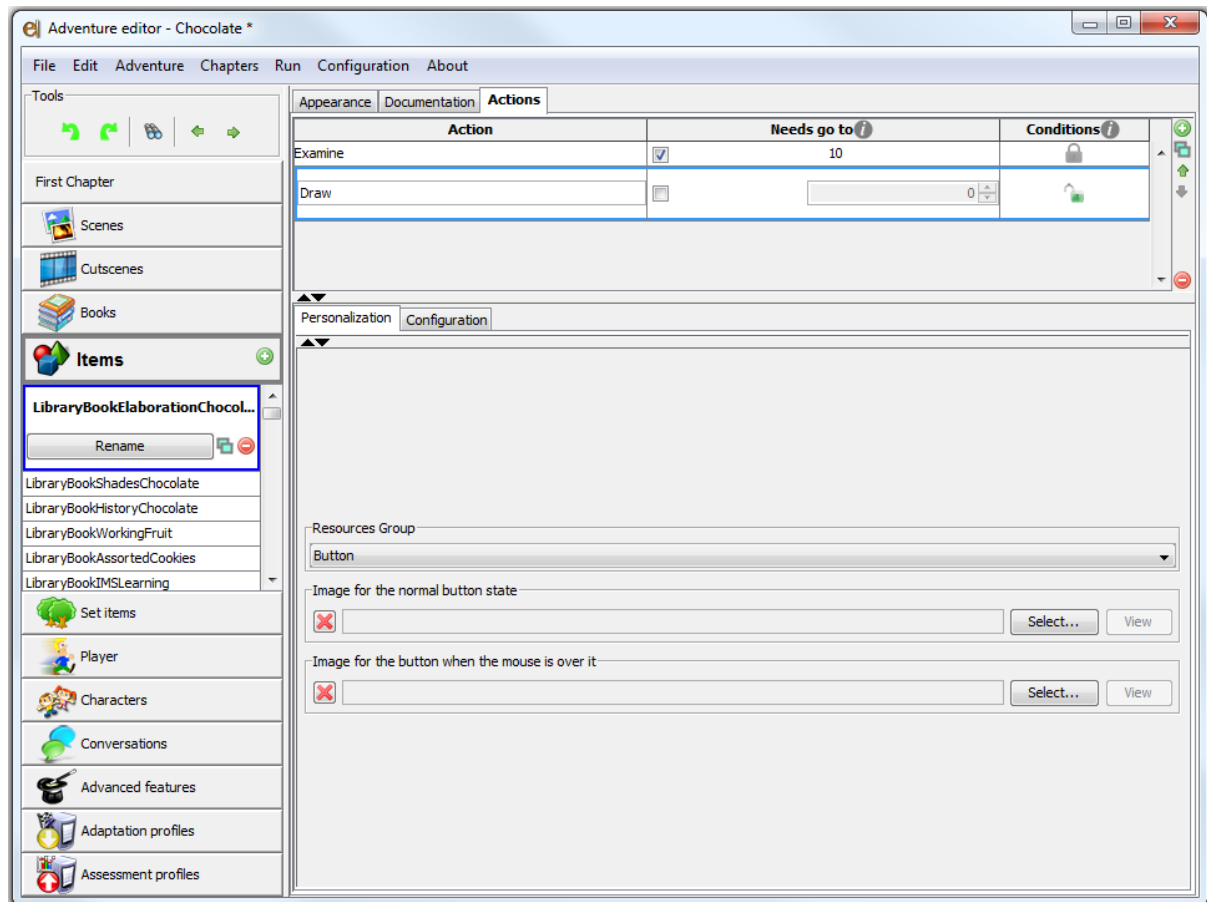


Figure 80. Adding a new custom action

**NOTE:** Please take into account that the optimum resolution for action button's images is 40x40 pixels. Recommended format is PNG. Buttons defined with images that do not comply with these guidelines may behave unexpectedly.

## 3.11. Built-in art resources edition tools

Although art resources have to be produced with specialized tools (e.g. Adobe Photoshop(TM)), <e-Adventure> provides support to make some slight adjustments or produce simple art resources. Some of these features have already been described. For example, in section 2.3.1 how to remove the white background of an image is described. In this section two tools are explained: the animations editor and the HTML editor.

### 3.11.1. Animations editor

This editor allows customizing animations for the <e-Adventure> game engine. Animations produced with this editor will be stored in format ".eaa", which is a proprietary format for <e-Adventure>. These animations will be used either for cut-scenes and characters. The animation editor also converts old <e-Adventure> format (groups of frame images with a common file name \*\_NN.png / \*\_NN.jpg where NN is the index of the gram) to the .eaa format automatically.

The animations editor is accessible through the “Create/Edit” buttons that any animation resource presents (Figure 81).

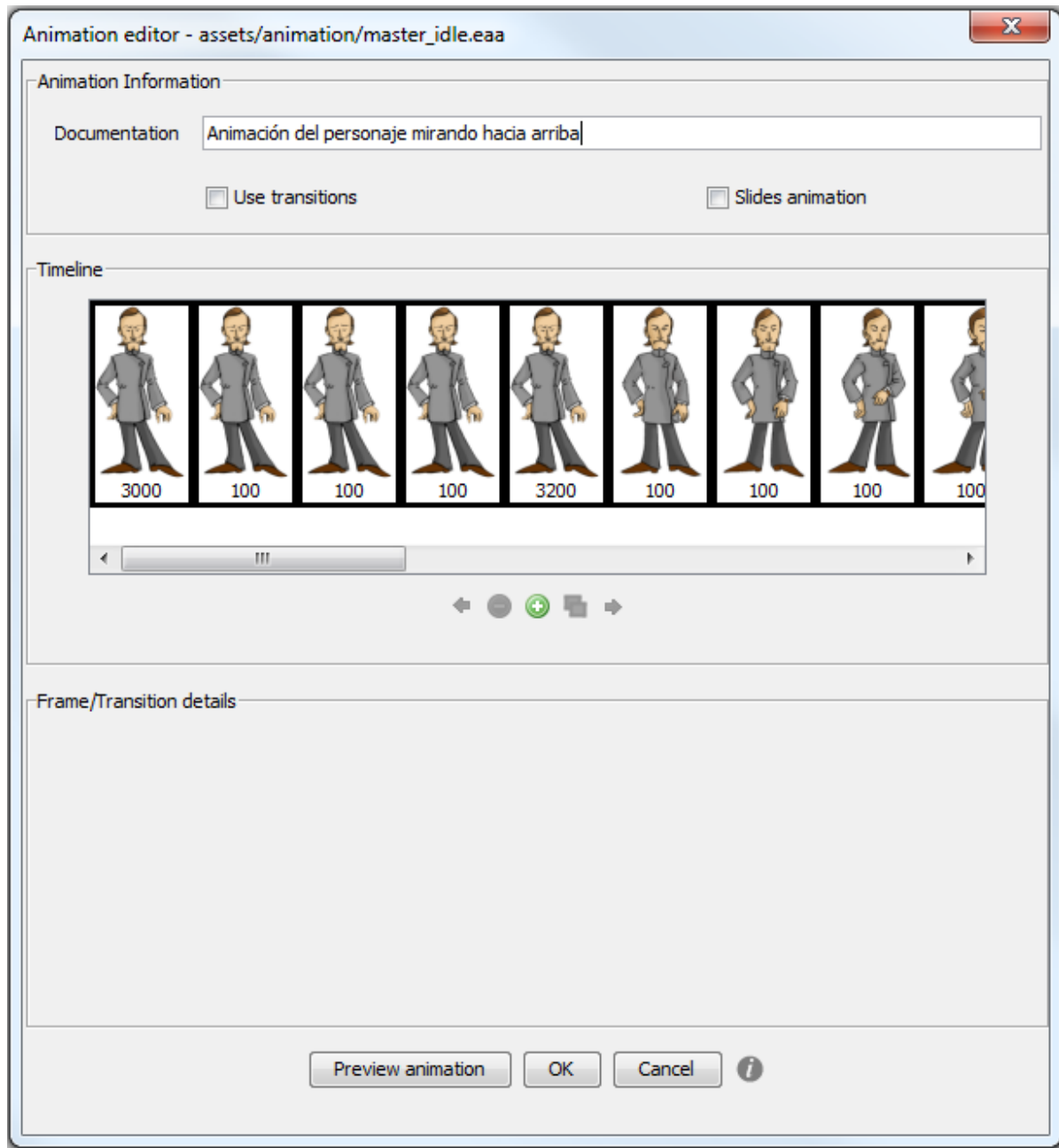


Figure 81. Animation editor

The documentation field is used only to provide an internal description of the animation, but is not used in the game. Under this field the next fields are presented:

- *“Use transitions”*: activates/deactivates the possibility to include transitions between frames.
- *“Slides animation”*: activates a special behaviour for slides, adding the possibility to block the next frame until the player clicks on the screen. It allows players to read slides at their own pace.

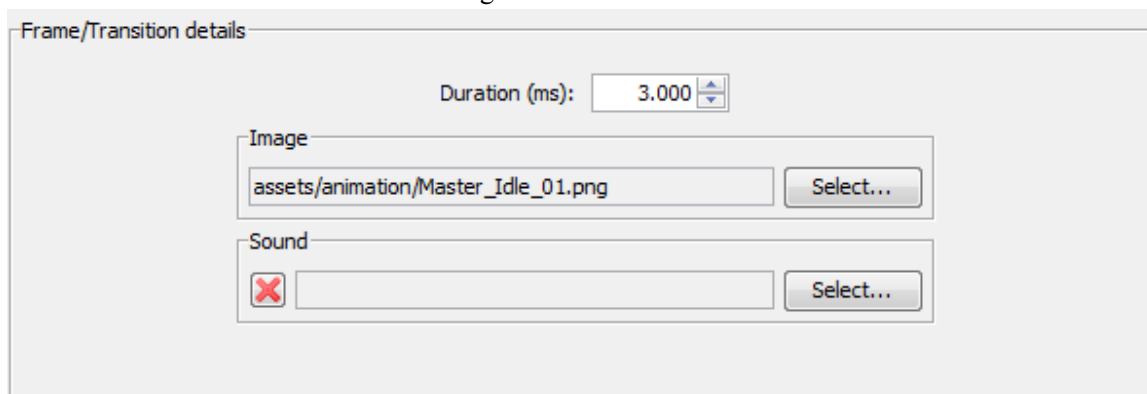


Under these options the timeline is presented. In this timeline frames and transitions are displayed in order. Frames can be added, removed or duplicated using the controls below (+, -, ). In addition, below each frame or transition its duration (in milliseconds) will be displayed.

Frames and transitions can be selected by clicking on them in the timeline. When a frame is selected its properties can be edited using the panel that appears just below (titled “Frame/transition” details).

When a frame is selected the next properties can be edited (Figure 82):

- Duration (ms): Time that the frame will be displayed on the screen (in milliseconds). By default this value is set to 40.
- Image: The image of the frame. Any png, jpeg, bmp or gif (static) image is accepted.
- Sound: A mp3 or midi sound that will be played synchronized with the frame. In this manner we can add sound effects to animations (e.g. step sounds when a character walks). Please take into account that these sounds will not stop when the next frame is loaded, so it is recommended that the frame’s duration is set to the length of the sound file selected.



**Figure 82. Properties of a frame in the animations editor**

When a transition is selected, the next properties can be edited (Figure 83):

- Duration (ms): Time the transition will be displayed (ms). By default this is set to zero.
- Type of transition: three types of transitions are currently supported:
  - *Fade in*: fades the previous and next frames: the previous frame disappears and the next one appears progressively.
  - *Horizontal*: the previous frame moves towards the bottom of the screen and the next frame appears on the top.
  - *Vertical*: similar to horizontal, but in this case movement is from left to right.



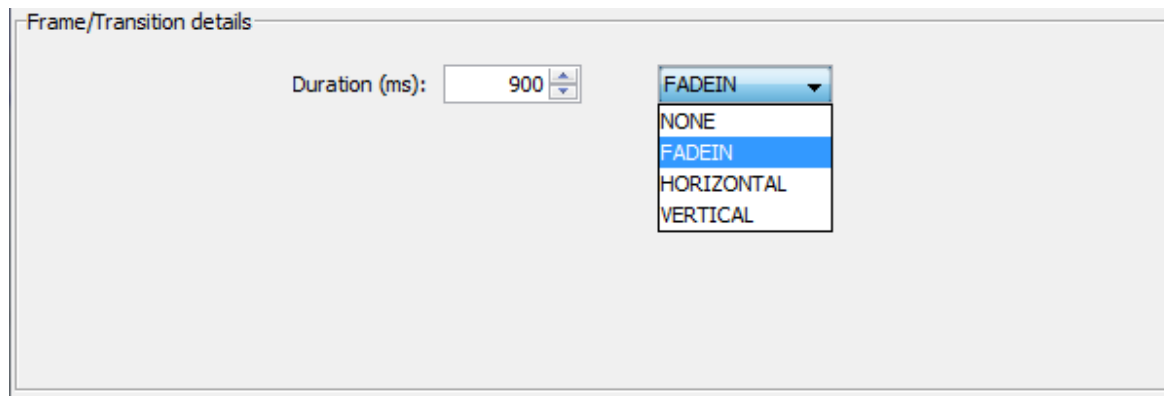


Figure 83. Properties of a transition in the animations editor

At any point the “Preview” button can be clicked and the animation will be played from start to end. OK and Cancel buttons are used to save or discard changes respectively.

### 3.11.2. HTML editor

This editor can be accessed when a “Formatted text” book is created. When a local resource is used to create a book page, this can be created directly in the editor by clicking on “Create”. We can also edit these files (even if they have been imported from an external source) by clicking on “Edit”. It must be noted that when HTML documents are imported these must not contain Meta data, since rendering problems may occur.

The editor follows a WYSIWYG philosophy (what you see is what you get). The editor allows adding resources as images in the documents, modifying text fonts, etc (Figure 84).

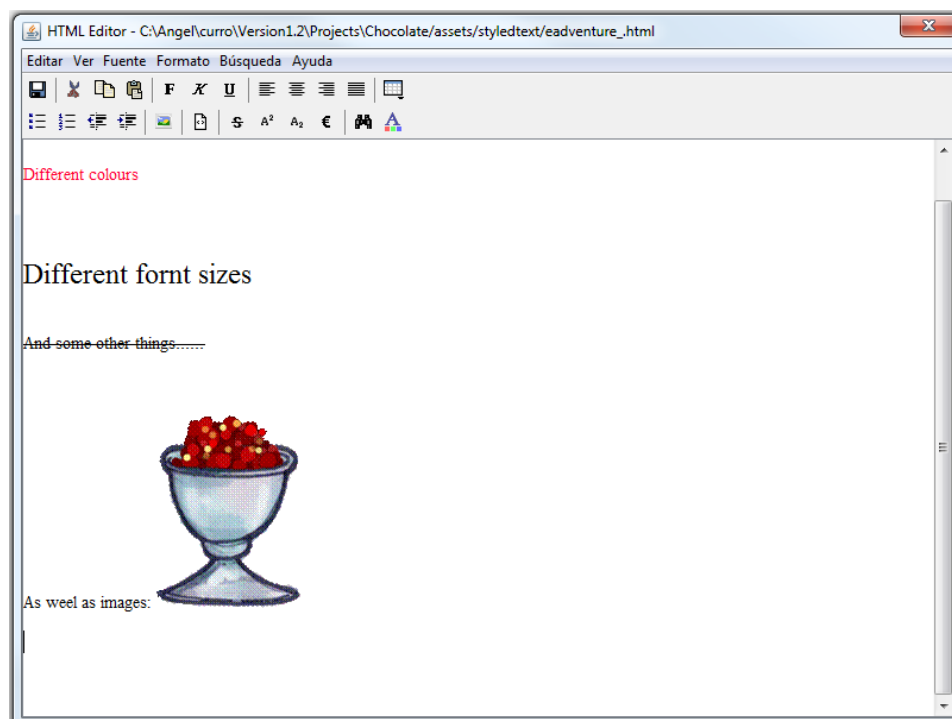




Figure 84. HTML editor

Once changes in the document have been saved, the results can be previewed in the “Preview” button that is provided at the bottom of the “**Contents**” page. You can also adjust the margins of the book by clicking on the “**Edit margins**” button that is provided for each page in the book list.

To improve rendering performance in the games remember to use the “**Export HTML book**” feature once the book is ready.

**NOTE:** The HTML editor has not been developed by the <e-Adventure> team and therefore we do not provide explicit support for this tool; it is provided “as is”, in the aim that it will be useful, but we are not responsible for any bug found. For more information please visit the web site of the author of this HTML editor: <http://editor.kafenio.org/>. We thank the authors for making this software available.

### 3.12. Polygonal exits and active areas

Exits and active areas can be defined using complex polygons instead of rectangles since version 0.10. This facilitates using exits and active areas when the background image is complex. The use is exactly the same in both types of elements. To activate this feature, select an exit or active area and press on “**Convert to irregular exit/area**”. Two new controls should appear just behind the button. The first button () can be used to add a new vertex to the polygon, while the second button () can be used to remove a vertex. Besides, in case the scene had been configured to use trajectories (see section 3.8) the influence area of the active area can be edited.

Each vertex can also be edited by clicking and dragging them on the preview panel below.

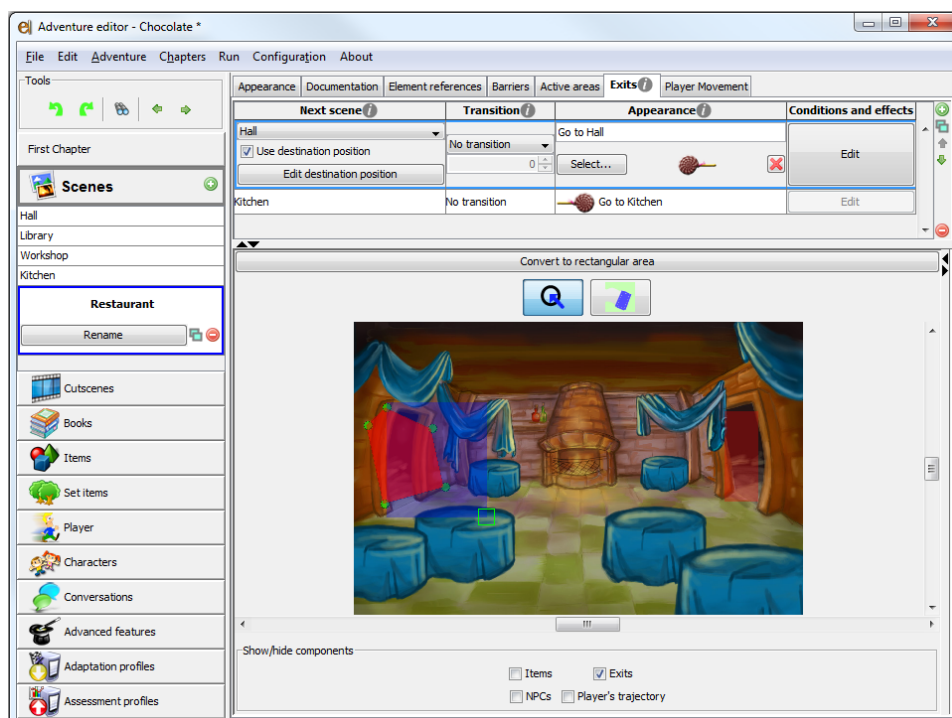


Figure 85. Creation of polygonal exits in a scene

### 3.13. Other features: error dialog and zoom panels

#### Error dialogs

Although <e-Adventure> is now a stable product and has been tested, it cannot be guaranteed that it is free of errors and bugs. If the editor or game engine crashes unexpectedly, an error dialog will be provided (Figure 86).

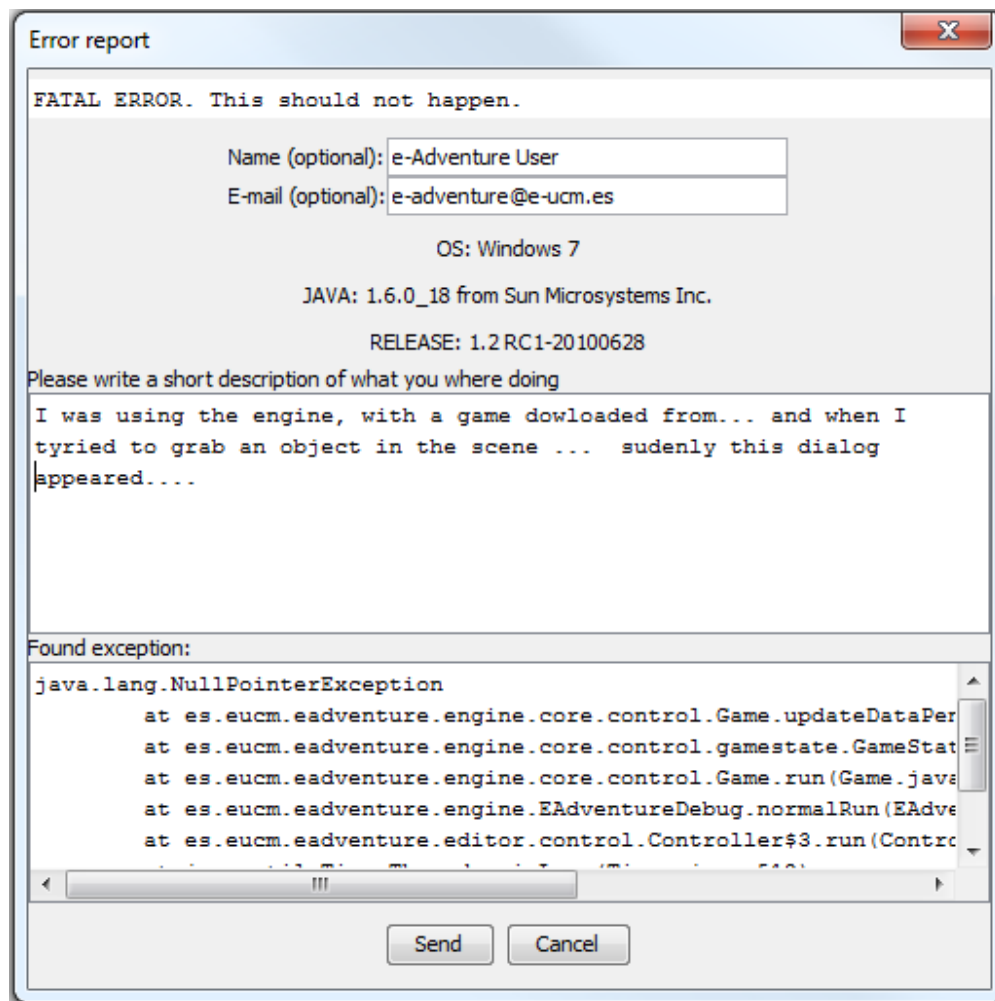


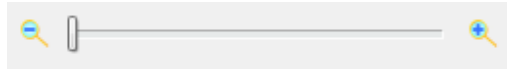
Figure 86. Error dialog

In case this may happen, users are kindly asked to fill in the form provided as much information as possible and press the “Send” button. The information presented in the dialog, and nothing else, will be automatically sent to the <e-Adventure> development team. Thanks for your cooperation!

#### Zoom in panels

In some panels it is possible to zoom in/out its content by using a slider and the zoom in (🔍) and out (🔍) buttons. This feature allows inspecting parts of the game in more detail and placing elements in the

scene more accurately. Therefore this feature does not affect the game itself, but just the previews in the game editor.



**Figure 87. Zoom controls**

## 4. Menu options

---

In this section the menus on the window bar will be explained. These include common features such as management of games and projects and other characteristics that allow further customization of the games.

### 4.1. File menu

This menu supports basic options to manage files, create new game projects, export games, exit, etc. Most of these features have been described in section 1.

### 4.2. Edit menu

This menu includes different edition tools:

- Undo: Allows undoing the last change made in the game project.
- Redo: Applies again the last discarded change (using the undo feature).
- Search: allows to search a string in the game project. A report dialog showing all the occurrences of the text will be returned. By clicking on each occurrence that element will open.

### 4.3. Adventure menu

This menu allows editing different aspects and provides general information of the game.

#### 4.3.1. Check adventure consistency

This option will return a dialog informing of all characteristics in the game that have not been defined correctly. If these errors are not solved the game will not be executed in the engine. For example, one of the most common errors is to leave the background image of a scene undefined.

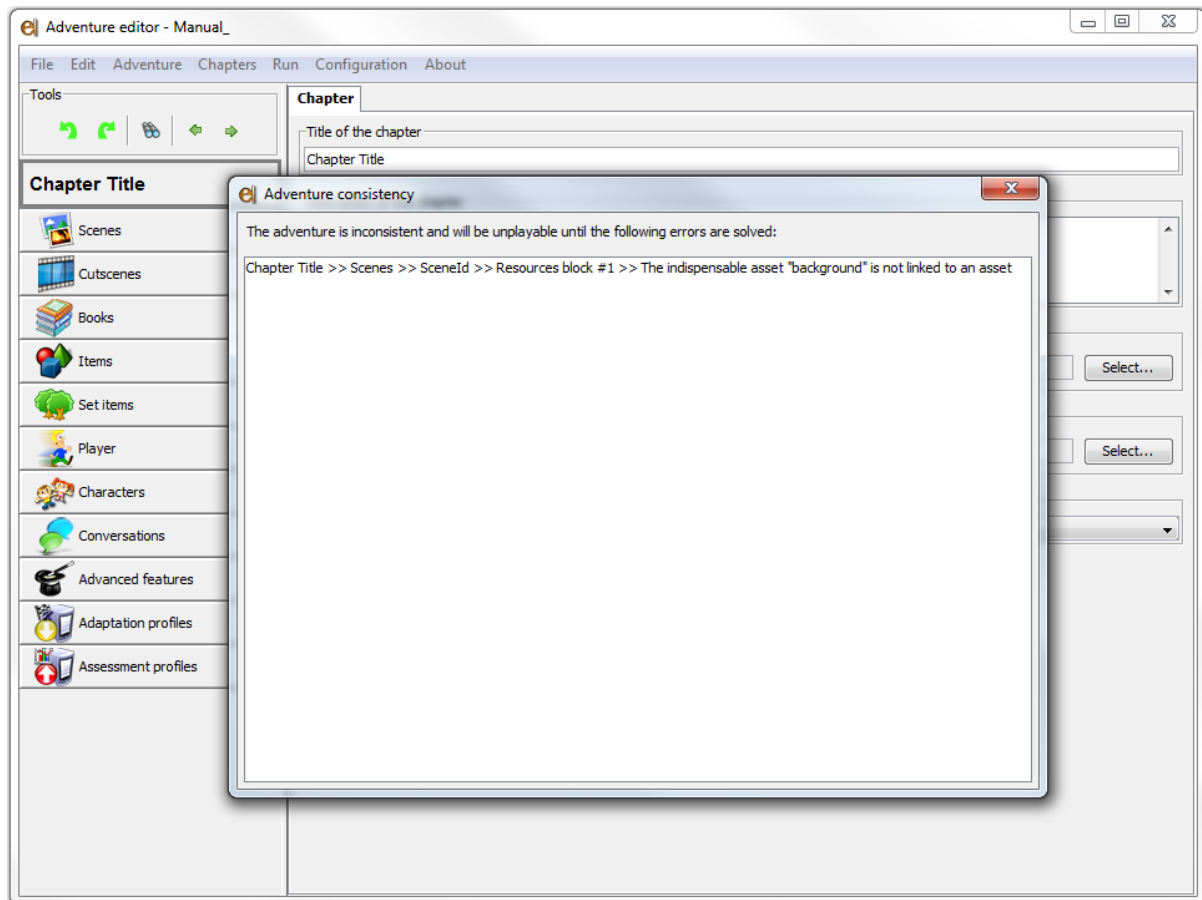


Figure 88. Check adventure consistency dialog

### 4.3.2. Edit adventure data

Launches a pop-up dialog for the edition of general adventure data, like the title and description.

In addition it indicates the current adventure mode (first person/third person) and provides a check box that can be used for activating automatic comments in the game. When this option is active, if the player tries to trigger an interaction with an element that is not available yet (because its conditions are not met at that point) or that has not been defined then an automatic error message will be spoken by the player's avatar.

In addition users can configure how dialog lines will be advanced in conversations. If "keep text until user interacts" is active then dialogs will no go on until the player presses a button. Otherwise dialogs will be automatically advanced.

When **Default action for right-click** is set the normal behavior when an *Item* or *Character* is right-clicked will apply. By selecting **Show details**, a text with the element's full description will be shown. Selecting **Show Actions**, available actions for the element will be displayed.

**Perspective applied...** sets the perspective to be used in character moving. It's recommended using **Isometric** option, because it's more realistic.

**Adventure data**

Adventure data

Title of the adventure  
Adventure title

Description of the adventure

Player mode (3rd or 1st person)  
Current mode  
3rd person

When this mode is selected, the player is normally painted in the game, as another character of the adventure. This mode is suitable for the production of common point-and-click adventures, such as the Monkey Island<sup>TM</sup> saga.

Automatic commentaries  
☐ Check to use automatic commentaries

For each conversation line  
☐ Keep text until user interacts

Default action for right-click  
Show details

Perspective applied to character and player movement  
Regular

Close

Figure 89. Adventure data dialog

### 4.3.3. Visualization sub-menu: customize GUI elements

<e-Adventure> allows customizing some of the elements of the user interface in the games: cursors, action buttons and the inventory.

#### 4.3.3.1. Cursors

Any of the default cursors used in the game can be customized by clicking on the “Select” button. Only restriction to be considered is that cursors must have a size of 32x32 pixels. Otherwise cursors could be rendered unexpectedly in the game engine.

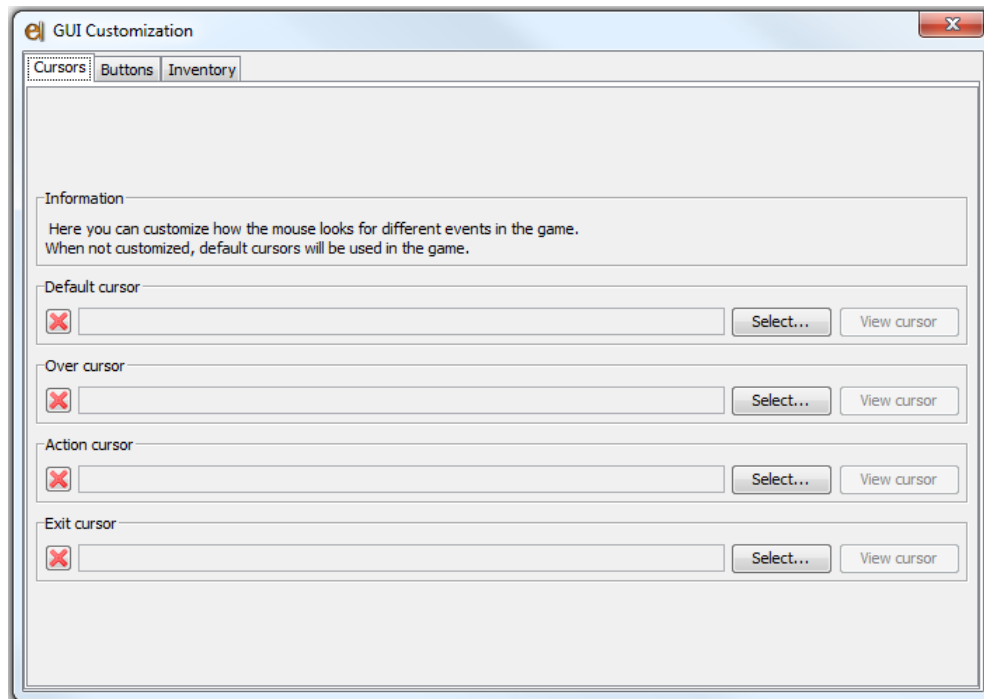


Figure 90. User interface customization dialog, “cursors” tab

#### 4.3.3.2. Action buttons

This tab allows configuring any of the default action buttons in the game (e.g. hand button for use, grab or use-with actions). Buttons should be at least 40x40 pixels dimensions. Each button has two images, one is used when the button is in normal state, and the other is used when the mouse is over the button.

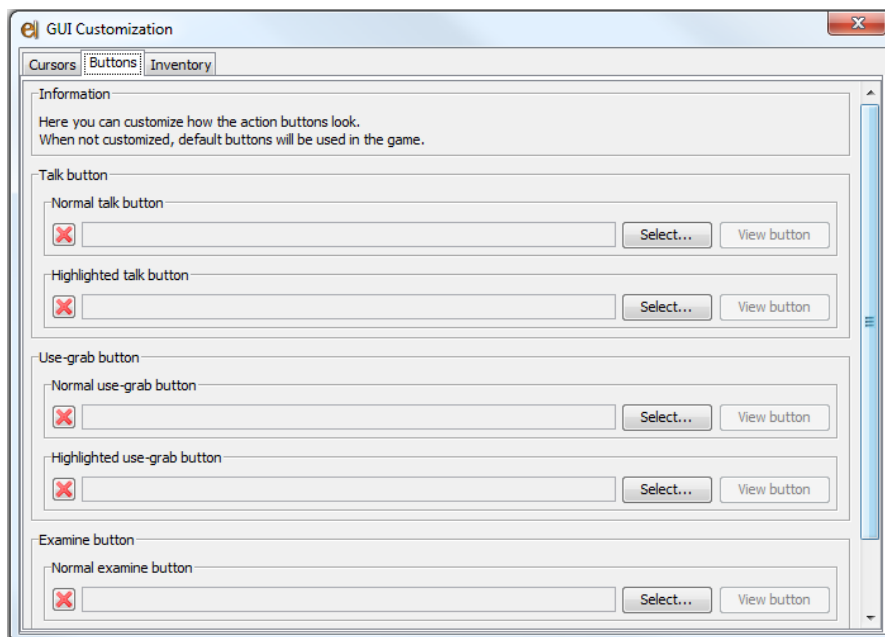


Figure 91. User interface customization dialog, “buttons” tab



#### 4.3.3.3. Inventory

This tab allows configuring the left and right arrows that appears in the inventory when it contains more items than could fit in the screen. Arrows must be at least 80x48 pixels.

In addition, areas on the screen where the inventory will appear can be configured (Top and bottom, only top, only bottom or none).

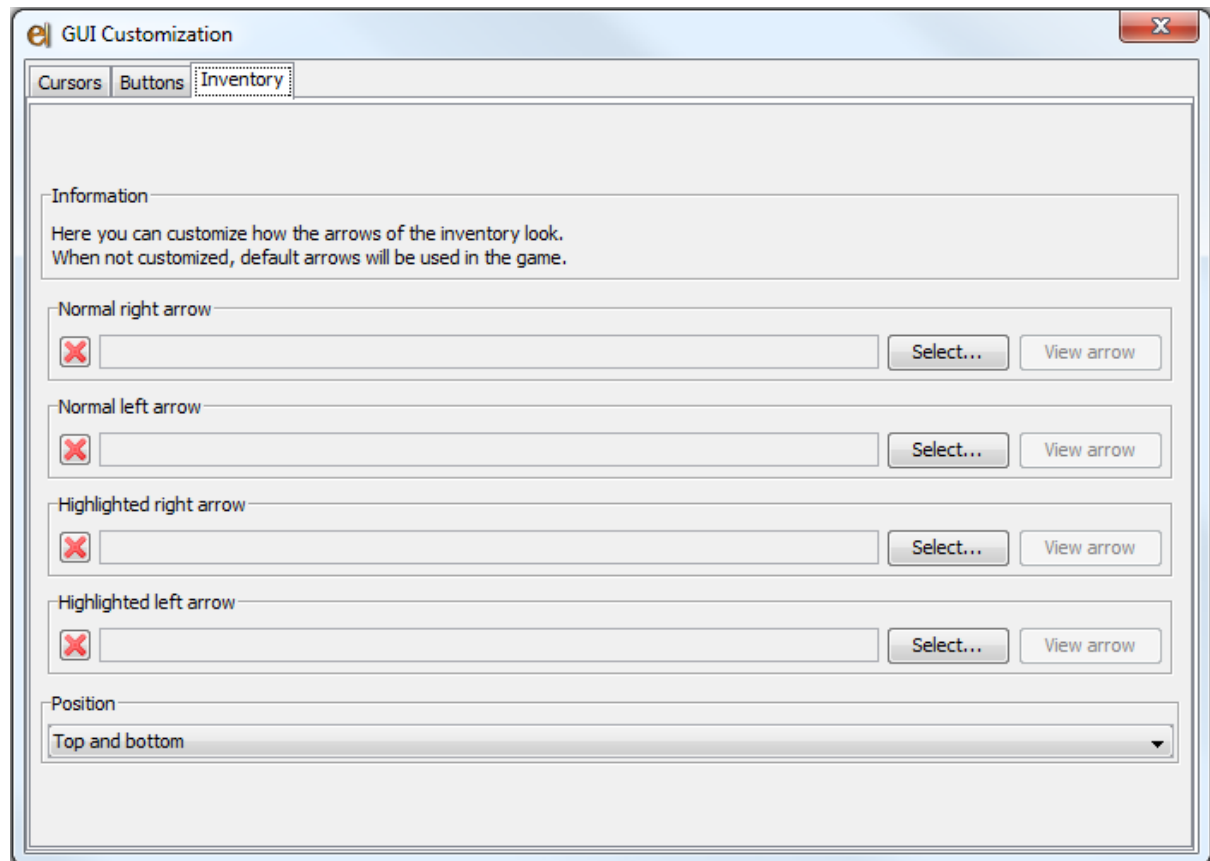


Figure 92. User interface customization dialog, “inventory” tab

#### 4.3.4. Visualization sub-menu: select graphic configuration

This dialog allows users to choose the graphic mode of the game window. Three options are provided: “Run in a window”, “Run with a black background” and “Run in full screen mode” (Figure 93).

- Run in a window is the default option. The game will run in a normal window of 800x600 pixels dimension. The game window will be centered in the screen.
- Run with a black background will add a black background behind the game window to hide the screen of the user.
- Run in fullscreen mode only works in Windows systems. It will change the monitor resolution to 800x600 so the game window will cover the whole screen.

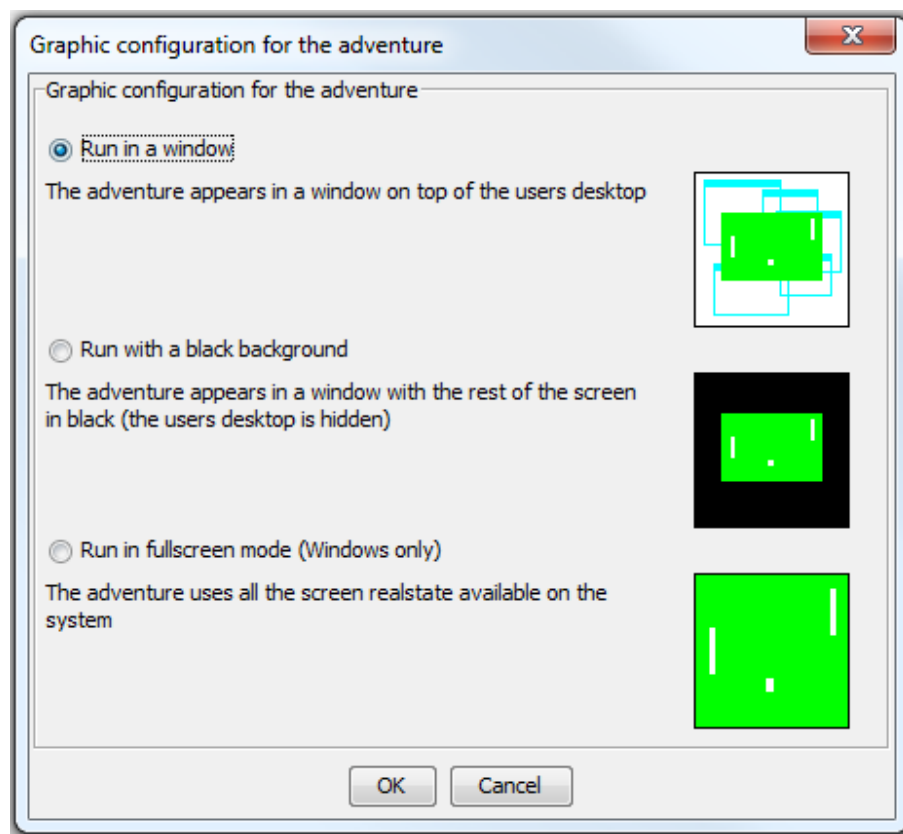


Figure 93. Graphic configuration dialog

#### 4.3.5. Change adventure mode (Convert to 1st/3rd person)

<e-Adventure> allows the creation of two types of games: first person and third person. In the first type the player perceives the game as if he/she would be part of the virtual world. In the second type the player will control an avatar that represents him/her in the virtual world.

This option allows changing the adventure mode. The process is completely reversible since no data is lost.

#### 4.3.6. Delete unused assets

This feature, added in version 0.10, allows removing art assets that have been added to the game project but that have not been referenced. This option is very useful when the game is about to be released/distributed, since it will remove some unnecessary weight. However, it must be noted that this process cannot be undone since files will be removed from the hard disk.

### 4.4. Chapters menu

This menu can be used to add, delete, import and order chapters. In addition flags and variables of the chapter can be edited using the option included in this menu.

## 4.5. Run menu

This menu allows running the games from the <e-Adventure> editor, which is very convenient to test while the game is being edited. To do this, press on the “Run>Normal” option (Ctrl+R).

There is also a debug mode, which allows running the game with additional information. When the game is launched in this mode the current value of flags and variables will be displayed on the left panel (Figure 94). In addition the event log located at the bottom of the screen will display events produced in the game.

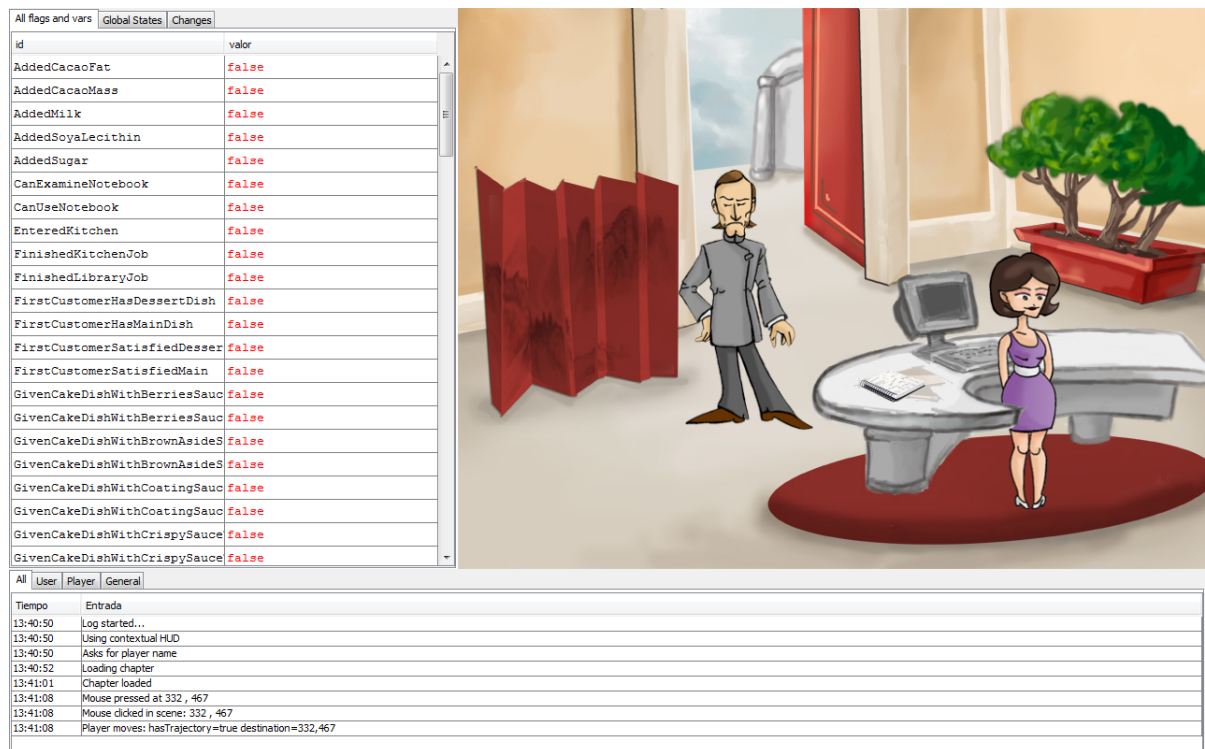


Figure 94. Running a game in debug mode

## 4.6. Configuration menu

This menu allows configuring the language of the editor. Currently <e-Adventure> is available in Spanish, English, Romanian, German and Portuguese. Please visit <http://e-adventure.e-ucm.es/contributors/> to see a full list of people that kindly produced each translation. Without them <e-Adventure> would never be multi-language. Thanks!

## 4.7. About menu

This menu provides information about contributors and authors. It also allows sending comments and suggestions to the development team.

Send comments and suggestions

Name (optional): e-Adventure User

E-mail (optional): e-adventure@e-ucm.es

OS: Windows 7

JAVA: 1.6.0\_18 from Sun Microsystems Inc.

RELEASE: 1.2 RC1-20100628

Please use this space for comments or suggestions

I have used e-Adventure for...

I think is very useful the fact of...

But I have found some problems when I tried to....

Send Cancel

Figure 95. Suggestions and comments dialog