

Quadrax X

Editor for Fans

User guide

Version 1.0

Translation to English: [Benko](#)

Contents

1. Information about the symbols and texts in the guidebook	6
2. Getting started	7
2.1. Introduction.....	7
2.2. Read carefully!.....	7
2.3. System requirements of QX EFF.....	7
2.4. Start slowly	8
2.5. Before you start for the first time	8
2.6. Distinctions between Quadrax X and QX EFF.....	8
2.7. Important warning	9
3. The main levels editor	10
3.1. Launching.....	10
3.2. Controls	10
4. The level editor.....	12
4.1. Editor modes, working with menu, windows and objects	13
4.1.1. Basic mode	13
4.1.2. The mode of object selection.....	13
4.1.3. The mode of object insertion	13
4.1.4. The mode of active object modification	14
4.1.5. The mode of object moving	14
4.1.6. The mode of creating object connections	14
4.1.7. The mode of testing level playing	14
4.2. Play (Enter)	15
4.3. Links (F1).....	15
4.3.1. Creating an object connection	15
4.3.2. Canceling connection	16
4.3.3. Additional information	16
4.4. Basic (F2).....	17
4.4.1. Wall.....	17
4.4.2. Trap-cellar	18
4.4.3. Ladder.....	19
4.4.4. Rope.....	19
4.4.5. Lift block	19
4.4.6. Figures	20
4.4.7. Levers	21
4.4.8. Buttons	22
4.4.9. Moving walls.....	23
4.4.10. Exit	25
4.4.11. Crystals	26
4.4.12. Clue scroll	26
4.4.13. Global lighting	26

4.4.14. Sign of open edges	27
4.5. Stones (F3)	28
4.5.1. Regular stones	28
4.5.2. Metallic stones	29
4.5.3. Explosive stone	29
4.5.4. Fragile stone	29
4.5.5. Wooden stones	30
4.5.6. Anti-gravity stone	30
4.5.7. Slippery volcanic stone	31
4.5.8. Slippery metallic stone	31
4.5.9. Magnetic stone	31
4.5.10. Metallic stone for magnets	31
4.5.11. Levitating stone	32
4.6. Movers/Ports (F4)	32
4.6.1. Figure passages	32
4.6.2. Figure teleports	33
4.6.3. Entrance stone teleport	33
4.6.4. Outlet stone teleport	34
4.6.5. Stone shifter	35
4.6.6. Force fields	38
4.6.7. Lifts	38
4.6.8. Entrance stone mover	39
4.6.9. Outlet stone mover	39
4.6.10. Crane	40
4.6.11. Magnet	40
4.6.12. Transporting belt	40
4.7. Mortal (F5)	41
4.7.1. Arrow shooter	41
4.7.2. Laser	42
4.7.3. Electric fence	42
4.7.4. Mortal flame	43
4.7.5. Target	43
4.8. Animations (F6)	44
4.8.1. Basic controls	44
4.8.2. Shift of animations in the axes x & y	44
4.8.3. Animation speed setting	45
4.8.4. Animations without animations (sounds)	45
4.8.5. Paper wall	46
4.8.6. Bulky generator	46
4.8.7. Light bulbs	46
4.8.8. Moving chains	47
4.8.9. Clock	47
4.8.10. Water of fountains	47

4.8.11. Luminous beacons.....	47
4.9. Create bitmap.....	47
4.10. Lock level	47
4.10.1. Level playability check.....	48
4.11. Clear level	49
4.12. Special editor functions.....	49
4.12.1. Moving an entire level.....	49
4.12.2. Level mirroring	49
4.12.3. “Object copying”	50
4.12.4. Visual display of the ancillary raster grid	50
4.13. Careful with the saveslots	50
4.14. Level statistics.....	51
4.15. Finishing work with the level editor	51
4.16. Important information about object collisions during level creating	52
5. Level textures editor.....	53
5.1. Introduction to the level texture editor	53
5.2. The difference between a level in the level editor and its real graphics look in the game	53
5.3. Editor and level texturing controls	53
5.3.1. Layers of the image	54
5.3.2. Showing and hiding of individual elements	54
5.3.3. Drawing the foreground (layer 2)	55
5.3.4. Drawing/erasing of the mid-ground (layer 1)	57
5.3.5. Ladders and ropes	58
5.3.6. Where are the shadows and lighting from the sun, flames and torches? Where are the statues and other decorations?	59
5.3.7. Finishing work in the texture editor.....	60
5.4. Advanced options of editing the level layers in external graphics programs	60
5.4.1. Placing files of the level layers	60
5.4.2. The principle of image arrangement in the game	61
5.4.3. Data format of PNG files and their dimensions	61
5.4.4. An example of a graphics project for the level one in QX.....	62
5.4.5. Tips and tricks for advanced layer editing	62
5.4.5.1. Using the wall masks for creating custom wall textures.....	62
5.4.5.2. Shading.....	63
5.4.5.3. Ancillary layers in the graphics project	63
5.4.5.4. Lighting from the sun, torches, flames etc.	64
5.4.5.5. Ladders and ropes for experts	64
5.4.5.6. Trees, statues, greenery and other graphics objects.....	64
5.4.6. Mysterious layer 3.....	64
5.4.7. The caveat for the end	65
6. Advanced options of game modding	66
6.1. Custom episode background.....	66
6.2. Custom level statistics background	66

6.3. Custom episode music.....	67
6.4. Custom texts of clue scrolls and diary.....	67
6.4.1. Texts of clue scrolls	67
6.4.2. The text of the diary	68
6.5. Custom graphics of active elements	69
6.6. Lastly about modding of QX EFF.....	70
7. Project manager	71
7.1. What a “project” is	71
7.2. After launching	71
7.2.1. Active project (ID).....	72
7.2.2. Project name	72
7.2.3. Project description	72
7.2.4. Allowed “Quadrax Neverending” moves	73
7.2.5. Allowed full editing mode	74
7.2.6. Number of playable levels in project	75
7.2.7. Date and time of creating and the last modification of the project	75
7.3. Projects.....	75
7.3.1. Create new project.....	75
7.3.2. Select active project	76
7.3.3. Import project	76
7.3.4. Projects maintenance.....	76
7.3.5. Exit.....	77
7.4. Active project	77
7.4.1. Setting	77
7.4.1.1. Basic	77
7.4.1.2. Advanced.....	77
7.4.2. Create copy.....	78
7.4.3. Export project.....	79
7.4.3.1. Lock exported project	79
7.4.3.2. Export saveslots	80
7.4.3.3. Export levels statistic	80
7.4.3.4. Further information about file exporting.....	81
7.4.3.5. Creating an export file.....	81
7.4.4. Check project.....	81
7.4.5. Delete project.....	82
7.5. Language.....	82
7.6. Info.....	83
8. Project publications from QX EFF in Quadrax Neverending	84
9. Conclusion	85

1. Information about the symbols and texts in the guidebook

In the text of this guidebook you can encounter three types of highlighted texts. Their meanings are respectively:

TIP



This is how useful tips will be highlighted which might help you with the editor or Project manager.

INFORMATION



This is how importation information will be highlighted related to the currently explained topic.

WARNING



This is how warnings during activities will be highlighted which might have great importance or during which critical operations with data are carried out or if they can cause other danger.

Moreover, there are images in the text which illustrate the needed keys, mouse buttons (or both) for changes and alterations of the individual level objects and other activities. For instance:



Pressing the B key.



Pressing the SHIFT key, etc.

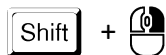


Pressing the left mouse button.



Pressing the right mouse button.

If a combination of keys or keys and mouse buttons is required, these are shown like this:



This means pressing and holding the SHIFT key and afterwards clicking the right mouse button.

And now you can get started with reading this manual! ;-)

2. Getting started

2.1. Introduction

It's vital to realize that the entire **Quadrax X – Editor for Fans** is solely a freeware product which is available absolutely for free. Therefore, I don't provide any tech support to it, etc.

This editor is based on the editor I've developed for creating Quadrax X and whose controls I've adjusted to my own manners from the editor for previous Quadrax games. So, if controlling of some elements will seem illogical or confusing to you, don't write to me e-mails where and how and what is bad in your opinion and how it ought to be. Controls work the way they do, and if you do want to use the editor, you'll have to get accustomed to them, including sometimes slightly lengthy setting of object parameters.

Despite this all, this editor is far more comfortable to use than the editor QV EFF. It's just important to get used to its specificities and then you can start creating levels (if you, of course, have the aptitude to always come with new and new points of levels). ;-)

2.2. Read carefully!

In this document you'll find all the information how to work with the editor and Project manager. In case you can't get the hang of anything, try studying the respective chapter more thoroughly – you're sure to find everything you need. This guidebook is interactive – this means you can click on the content items and underlined links using the combination of pressing the CTRL key and the left mouse button on the link. You'll get instantaneously where that link leads. Nonetheless, I strongly recommend to print the entire guidebook so that it will always come in handy whilst working with the editor.



I don't describe in this guidebook the rules of figure moving (apart from the difference of figure moving in comparison with Quadrax Neverending) and object behavior – I assume everyone who intends to work with the editor knows my Quadrax games in detail!

2.3. System requirements of QX EFF

Basically, QX EFF connects in itself both – the game and the editor. Therefore, it has a lot higher memory requirements for the graphics system than Quadrax X! The minimal capacity of graphics card is 512 MB, optimal is 1024 MB and more. Requirements for processor and operating memory aren't noticeably increased in comparison with QX.

What is very important, though, is the monitor. Some information and icons have been designed on a pixel level and on resolutions lower than 1920x1080 they needn't be visible properly (or not visible at

all)! So the monitor requirement is rather important here. On the monitors with lower resolutions working with the editor can be literally frustrating.

Minimal requirements essential for launching QX EFF:

- Processor Pentium 4 or higher (alternatively an AMD equivalent)
- OS Windows XP/Vista/7/8/10
- DirectX 9.0c
- RAM at least 1 GB (for Windows XP), for Vista and higher at least 2 GB
- At least 1.2 GB of free space on your hard drive
- Graphics card with at least 512 MB memory (ideally 1024 MB and more)
- LCD monitor (I strongly recommend only ones with the native resolution of 1920x1080 or 1920x1200)
- Sound card compatible with DirectX

2.4. Start slowly

Don't try to create very complex levels at first. Instead focus on making simpler levels and once you get accustomed to working in the level editor, you can gradually add more and more elements. The complexity of levels is not about cramming a shed-load of levers into it, but about the idea – a great and difficult level can also be made using stones only. So, do start slowly by building simpler levels, and then you can gradually make more and more complex levels.

2.5. Before you start for the first time

Before you start constructing your own levels, I strongly recommend to create a new project for it. Certainly, you can also work in the default project 0000, but it's better to leave it "void" and create a new project for your first attempts. The instructions regarding creating new project can be found in the controls description of Project manager in the chapter [Create new project](#).

2.6. Distinctions between Quadrax X and QX EFF

What will prompt you immediately once you launch QX EFF is that there is no user account system known since the times of Quadrax VI. That's because it is not needed. On its place, in the main menu, there is an entry menu for entering the level editor. And if you click the item best results, you'll be shown the best results for your currently active project.

You won't find too many visual distinctions in it. However, under the hood of QX EFF there's a significantly improved engine of the original Quadrax X beating which I have altered and innovated according to my best knowledges. It takes the best from QX and also adds a heap of new features.

Before you start working with QX EFF, though, I recommend reading the entire guidebook in detail. I understand that those who have been restlessly waiting for QX EFF would mostly like to immediately

start creating astonishing and smart levels, but without the knowledges from this guidebook your levels will be neither astonishing nor smart.

2.7. Important warning



Despite the fact this editor originates from Quadrax X basics, its data structures for levels and objects are vastly distinctive! One of the reasons is that it supports project administration (and each object has an encrypted trait of the unique project identifier) and also the objects themselves have a different data structure and a different encrypting method is used for key files.

I'm warning you in advance that no one should try opening source files of Quadrax X via it. It doesn't work!


In a better case you'll only damage data in Quadrax X, in a worse one the configuring and data files of the game or editor will be damaged!

3. The main levels editor

3.1. Launching

You can get to the main editor easily from the main menu of the game by clicking the item surprisingly called “Levels editor”. ;-) Then you’ll be shown a window where there’s an insight of all 100 levels which are present in the project. Levels are sorted to the 10x10 raster and each row of ten levels represents one episode (the level group which share the same background, music, environment settings etc.). Each insight schematically shows the content of each level what makes the orientation in more complex projects easier.


3.2. Controls

The controls of the main level editor are truly easy – using mouse you can open, move or copy individual levels. When mouse is moved above the levels, the current level number is shown nearby the mouse cursor. If you click on a respective level with the left mouse button , the editor of that level will pop up. If you hold the left button, you can move the level insight on the screen (the cursor arrow disappears and behaves like the entire level miniature which has its level number in the left-hand corner). As long as the level is “on its original place”, EDIT will pop up next to the level number – this indicates that releasing the mouse button will launch the level editor. However, if you move the level whilst holding the button, an indicator <-> (a bi-directional arrow) will show up next to the level number and behind it the level number that is currently under the moved level will appear. After releasing the button, the level movement will be carried out – the level that you were moving with the mouse will place on the level space above which you’ve released the left button and the original level, which had been lying under the moved one, will get on its original place.

This way, you can discretionarily move and arrange the levels in the raster. Therefore, it doesn’t matter on which place you’ve created and edited your level – you can always move it elsewhere in the full scope of all hundred levels.



Together with the level all regarded data is moved as well, this means the entire level statistics, textures, alternatively saveslots, etc.

During the level moving, if you click elsewhere holding the key , the indicator in the insight will change to -> (right arrow) what indicates that releasing the left mouse button will **copy** the level to the new place. This way, the level will stay on its original placement and its entire content will be copied to the level above which you’ve released the left mouse button.



Copying and rewriting isn't anyhow protected with an additional question "Do you really wish to...?" and releasing the mouse button will carry out immediate and IRREVERSIBLE data copying! So, beware of copying levels in order not to rewrite a level you have been working very long before! It's appropriate to lock the prepared levels what protects them before unwanted rewrite or deletion, see the chapter [Lock level](#).

4. The level editor

The level editor opens, once you click on any level in it (see the chapter above). This editor is the main core of the entire QX EFF and using it you can completely edit and test an entire level including all its elements and objects. This is why the largest part of this manual will be dedicated to this “central” editor.

Visual user interface of the editor is maximally simple and it basically consists only of the main menu pane on the top of the screen. The items of the menu can be easily chosen by hovering the respective part of the menu with mouse or pressing the respective functional key, if the menu item contains it. Respective elements of the menu will be described more in detail in the next part of the guidebook, here is a concise list of menu functions:

- Play (Enter) – after clicking (or pressing ENTER), it initiates the level playing in the special editor mode
- Links (F1) – after clicking, it shows the editing mode of connections of functional objects
- Basic (F2) – after hovering with a mouse, it shows a window with the main objects
- Stones (F3) – after hovering with a mouse, it shows the window of stones selection
- Movers/Ports (F4) – after hovering with a mouse, it shows a window of the active objects
- Mortal (F5) – after hovering with a mouse, it shows a window of the “mortal” objects
- Animations (F6) – after hovering with a mouse, it shows a window of the animations
- Create bitmaps – after clicking, it opens the editor of level textures
- Lock level – after clicking, it locks away level and protects it from being edited (confirmation is asked)
- Clear level – after clicking, it completely erases the entire level (confirmation is asked)



If the editor is in the basic mode (see the next chapter), the menu pane can be hidden at any time by pressing the key **[M]**. Next pressing of the key **[M]** will show the menu again.



If any of the editors (of levels, level, textures) is active, the function of immediate game quitting using the key Pause (BHK) is temporarily turned off because of security reasons!



If the top menu pane doesn't show in the basic mode, you're likely to have hidden it by pressing the key **[M]**. The next pressing of the key **[M]** will turn the menu back on. **Keep in mind that in the modes of inserting or removing of object or in the mode of target selection during creating connection the top menu pane is always automatically hidden!**



The huge amount of editor functions is controlled by using keys on the keyboard or their combinations, alternatively by the combination of keys and mouse buttons. In the editor itself there are no drop-down menus that could give you these options. This is why all the controlling keys of all the editor functions are thoroughly described in this guidebook.

4.1. Editor modes, working with menu, windows and objects

Menu items which have a function key (F1 – F6) in brackets behind their names can be activated by pressing of these respective keys if the editor is in the basic mode.

The level editor has seven main modes:

4.1.1. Basic mode

This mode is always set by default once you enter the editor from the main level editor. A standard mouse cursor is shown and there's an option of clicking with a mouse on respective menu items and all active object in the level (the distinction between active and static objects is described in the chapter [Basic \(F2\)](#)).



Regardless of what editor mode you're in, you can get back to the basic one by clicking the right mouse button once or more times (if the cursor is not placed above an active object).

4.1.2. The mode of object selection

If you hover with a mouse at some of the menu items Basic to Animations or press a respective function key (F2 – F6) a window with the objects regarding the respective menu item will pop up. By clicking with the left mouse button, you can choose a discretionary object. By clicking with the right mouse button or pressing the key Escape you'll close the active window (if, however, the cursor doesn't point at the menu item) and the editor will revert back to the basic mode.


4.1.3. The mode of object insertion

If you choose from the windows of the menu an active object by pressing a left mouse button, the mode of inserting a specific object to the level will activate. The mouse cursor will afterwards transform to the chosen object and will start blinking – the picture of the object will alternate with the sign PUT OBJ (put object). At this moment you can place, even repeatedly, the object to the level by clicking with the left mouse button. This means, as far as you're clicking with the left mouse button, more and more objects of the same type will be inserted to the level. If you press the right mouse button or the key Escape, the mode of object inserting will terminate, and the editor will revert back to the basic mode. **In the mode of object inserting, the top menu will always automatically disappear!**


4.1.4. The mode of active object modification

If the editor is in the basic mode and the mouse cursor is placed above an active object, the cursor will get transparent, and the active object below the cursor will start pulsing. At this moment you can change the parameters of the respective object below the cursor and also additional information about the state of the object will appear above this object. The possibilities of parameter modification of the respective objects are always strictly set by the values of the respective object. The descriptions of the possibilities of parameter editing are always listed at the description of the respective object in this guidebook.

4.1.5. The mode of object moving

If you press in the mode of active object modification (in other words, if the mouse cursor is placed above an object, it's translucent and the object below the cursor pulses) the left key button, the mode of object moving will activate. The mouse cursor will change to the chosen object and you can move it anywhere else in the level. The object will be blinking and alternating with the square with the sign MOVE/DEL OBJ (move/delete object). After the move, if you press the left mouse button once again, the object will place to the new spot in the level. If you, during the move, press the right mouse button or key Escape, the object will return back to its original place and the editor will revert back to the basic mode. If you press in the moment of the move the key Delete () the object will be erased, and the editor will revert back to the basic mode. **In the mode of object moving, the top menu is always automatically hidden!**


4.1.6. The mode of creating object connections

If you click the item Links (F1) in the basic mode or press the key , the editor will switch to the mode of creating object connections between the active elements of the level. The detailed description of this mode is listed in the chapter [Links \(F1\)](#).

4.1.7. The mode of testing level playing


After clicking this menu item (after pressing the key [Enter] in the editor basic mode) the level playing in the editor mode will be initiated. This is described more in detail in the chapter [Play \(Enter\)](#).




If you want to erase a bigger amount of active objects in the level, then in the basic editor mode press and hold the key Delete () and click with the left mouse button on the respective objects.

In the following part of this manual there are thoroughly described all of the menu functions and all the objects of the respective windows in the order which is listed in the menu.

4.2. Play (Enter)

After clicking this menu item (or after pressing the key  in the editor basic mode) **all the changes made in the level will automatically save**, and the level will run in the special testing mode. It differs from playing levels right in the game (in the regular playing mode) by these facts:

- The background is black screen (instead of the background image of the episode and its animations).
- The music of the episode doesn't play.
- Instead of level textures, schematic editor graphics is used.
- It's possible to continually rewrite all the saveslots.

In order to leave this mode, all it takes is press Escape key () and the editor will revert back to the basic mode. The interruption of level playing and the return to the editor will also happen in case that one of the figures dies, or when all the figures make it to the exit.

In this special playing mode, it's easy to test the entire level or only its parts, and it's the main tool for complex testing of all level functions. So, feel free to use it as often as possible.


4.3. Links (F1)

In this mode the editor allows you to connect all controlling and active elements. Once this mode is chosen, all other menu items will become inaccessible (apart from Play). Connecting of the individual objects is shown using the yellow lines between them. If this option is at some objects non-standard (distinctive from the general principle described in this chapter), it's always expounded at the object description in the chapter below.

4.3.1. Creating an object connection

Connection can be created by simply clicking with the left mouse button on a controlling (or source) object. Controlling and source objects, which allow creating a connection, pulse slightly below the mouse cursor. After clicking the controlling/source object, a small sign SET/DEL TARGET (set/delete target) will appear at the mouse cursor, and also a pulsing line from the source object will begin to appear at the mouse cursor. If the source object is a controlling object (i.e. lever or button), a yellow border will appear around that source object. The links of all other connections of other objects (apart from the source ones) will be shown very translucently so as not to interrupt the selection of the target object for connecting.



If there are many connections in the level, they can interrupt even these translucent links – by pressing and holding the key  all other object connection in the level will be hidden (as long as the key is held), apart from the current object for which you're currently creating the connection.

As soon as you hover with the mouse above the intended object allowing the logical connection with the source object, a yellow border will appear around that object indicating that it is possible to create a connection. Clicking that object creates the connection. If the source object is a lever or button, the mode of target selection remains on even after connection creation. If you want to cancel the mode of target selection, press the right mouse button or the key Escape. Whilst creating connection which can have only one target (between two teleports, tunnels etc.), the target selection will be canceled automatically after creating a new connection or canceling the old one.

The link of the connections between levers, buttons and active objects leads always from the middle of the field, on which a lever/field is placed to the **top left** corner of the intended object. The connection links between the rest of the objects (their beginning and end) also depend upon the type of each object, and they're described in the details of each object below.

4.3.2. Canceling connection

Connection can be canceled the same way you created it. Therefore, by clicking the source and then the target object. The created connection will be erased (this applies even when you're currently creating a connection – first click creates the connection, second click on the same object cancels it). Passage and teleports of the figures are the only exception (which always to have a target) – connection cancellation in that case is carried out by choosing as the source passage/teleport the same one that will afterwards lead to "itself". At objects that can have only one target (teleports, stone movers, passages and teleport of the figures etc.) creating a new connection on other target object automatically erases the original connection.



The editor of connections is intelligent and doesn't allow creating connections between objects that cannot be logically connected (e.g. lever and figure passages).

4.3.3. Additional information

Each lever or button can control up to twenty different objects. On the other hand, each object can be controlled by a discretionary (unlimited) amount of levers or buttons.

By clicking on Links, pressing the key F1 or Escape, or by pressing the right mouse button you'll get to the basic mode of the editor.



Apart from setting object parameters and wall creating, the right mouse button mainly has the same function as the key Escape – i.e. it cancels the current mode or changes and reverts to the basic mode of the editor.

4.4. Basic (F2)

This option shows the window of selection of basic level objects which contains both static and active objects.

In the entire editor there are two types of objects:

- **Static**, creating the level “outline”
- **Active**, creating active level objects

In whole editor are **only seven** static objects which are also placed in this window. They are:

- Solid wall
- Red, yellow or blue trap-cellar
- Ladder
- Rope
- Lift block

Static objects are basically a component of the basic level “construction” and aren’t dependent upon any active object whilst level creating and playing. (They are placed in separate source files of level construction). Their amount in the level is not limited and their canceling follows the specific rules which will be expounded in the following chapters.

All the static objects in the game are basically a “wall” (yes, even ladders and ropes), they just have different values to a regular wall. This means that on one place (position) of the level there can be only one type of a static object, so ladders and ropes or walls cannot be combined in one place.

Whilst placing these objects, ones that are right below them are always erased – keep this in mind; particularly if you create or erase a block of walls!

The descriptions of controls of static objects of the window Basic:

4.4.1. Wall



The basic and real “building stone” of levels. Where there’s a wall, nothing else will get. Since it’s basically the most used element in the editor, you can choose it in the window editor, but also by simply pressing the key **W** (for **W**all) if the editor is in the basic mode.

Object controls:



- creating a wall





- erasing a wall





+



- block wall creating (from the left to right and down)

 +  - block wall erasing (from the left to right and down)

 or  - return back to the editor basic mode



You have to remove all other static object using the function of wall erasing as well – they cannot be removed any other way than being erased using wall erasing!



Block wall creating/erasing is a very handy tool, even though you need to use it carefully. Since there IS NO function UNDO in the editor, it can occur to you very easily that you accidentally erase (or “wall”) a part of the level. Then all you can do is leave the editor using the key shortcut SHIFT + Escape when no changes in the level are saved until the last automatic save. (For more information see the chapter [Finishing work with the level editor.](#))

4.4.2. Trap-cellar



This item reacts according to its type for being passed by on or below, or by a fall of a stone on it. According to the skull's color they have different functions:

- Red ones – react to figures only
- Yellow ones – react to figures or a fall of a stone
- Blue ones – react only to a fall of a stone



If a stone falls on a yellow trap-cellar or blue trap-cellar, it depends upon what the stone fell on with its other half. If it's any stable support, the trap-cellar won't fall apart! Also if it's a magnetic or explosive stone, which at the moment it falls on the trap-cellar reacts with a stone of the same kind (magnetization/explosion), the trap-cellar won't fall apart.




Despite the fact the amount of trap-cellar in level is unlimited, the engine is capable of managing the actions (disintegrations) of up to 40 trap-cellars at one moments. This means that if you created a level, where 40 stones fall on 80 trap-cellars, an “error” will occur and only the first 40 trap-cellars will shatter!

Object controls:



- creating a trap-cellar



or  - return to the basic editor mode

4.4.3. Ladder




This item allows you to insert vertical ladders into a level.

Object controls:



- creating a ladder



or  - return to the basic editor mode



Figures CANNOT climb above the level of the last ladder rung! So, if a ladder is supposed to lead on a higher level, it's needed that the ladder is long enough! On the other hand, if a figure climbs from an unfinished ladder in the way down, it will fall from it.



If you have already rendered ladders and ropes to some of the layers of the graphics level look (see the chapter [Ladders and ropes](#)), **creating, erasing or rewriting ladders in the level editor is blocked!**

4.4.4. Rope




This item allows you to insert horizontal ropes into a level.

Object controls:



- creating a rope



or  - return to the basic editor mode



If you have already rendered ladders and ropes to some of the layers of the graphics level look (see the chapter [Ladders and ropes](#)), **creating, erasing or rewriting ropes in the level editor is blocked!**

4.4.5. Lift block




This special object is used only for preventing a lift moving anywhere beyond that object (it can be blocked from the top or bottom edge). It doesn't react with any other objects and is not visible whilst playing. This object can be used particularly there where a wall cannot be placed (typically by hindering a way of the lift above the level of a top storey, etc.), therefore, where you need a lift to be incapable of passing, but where figures and stones could move without any restrictions.

Object controls:



- creating a lift block



or  - return to the basic editor mode



All the static objects mentioned above are truly “static”. **Hence, they can’t be moved with a mouse after being created unlike active objects!** So, if you want to place any static object elsewhere, you have to erase that object using the function of wall erasing (see the chapter [Wall](#)) and then place a new one somewhere else!

Now we’ve come through all static objects, and now the description of active object will follow. So, please, pay attention to the following information!



There can be only 255 active objects in a level! If you attempt to insert an object 256, a warning tone will beep and the object won’t be able to be placed to the level. Consider this fact when you create more complex levels and insert, for instance, additional animations (dripping water, torches etc.) once you’ve created and tested the entire level!

The descriptions of controlling active objects of the window Basic:

4.4.6. Figures



Well, these are our heroes. All three figures are active in each level by default and are placed in the left-hand bottom corner of the level, so choosing them from the window is necessary only in that case if you happen to accidentally remove the respective figure. ;-)

Object controls:




- choosing a figure for moving or removing



- (Delete) removing a figure if the moving mode is active from the step above



or  - return to the basic editor mode, if the mode of object moving is active



There always must remain at least one figure in the level. This one cannot be removed!

4.4.7. Levers



This is the basic object for controlling the rest of active objects of a level. One lever can control up to twenty objects.

Object controls:



- choosing an object for moving or deleting



- switching the position of a lever (left/right)



- switching the kind of a lever between red/blue/green/gold(timed)



- (Delete) removing an object, if the mode of object moving is active






- return to the basic editor mode, if the mode of object moving is active

At the “golden” (timed) type of levers a 4-digit counter will appear above it defining how long a lever will be switched before reverting back to its original place. The value is set in the cycles of the game, and one unit corresponds to approximately 16.6 milliseconds (1/60 sec). By default, the value is set to zero – in this case a lever would work just like a blue one – after switching it remains blocked permanently (it also works this way if it’s switched to the value of 1). If you want a lever to be switched for a specific amount of time, the value you have to assign to a lever can be counted easily using the equation **amount of seconds × 60 = value**. This means that if you want a lever to be switched for **three** seconds, assign it a value 0180 (**3 × 60 = 180**). For ten seconds you would assign it a value 0600 etc.

Switching values of a timed lever:



- switching the units


Right  +  or  - switching tens

Left  +  or  - switching hundreds



Don't assign to timed levers a value smaller than 20! (I.e. approximately one third of a second.) Despite the fact the game engine allows this, in this case the figure animation switching a lever is not synchronized with lever move. (A lever will revert back faster than the figure animation for switching a lever happens and it looks odd!) If you set up an even smaller interval, the move of a lever may not be visible by a naked eye at all ... Also, avoid setting too long times. If you go beyond the value of 9999 (about 166 seconds), the counter of a lever will show the symbol xxxx, but the inner counter will still work. However, such long times are not needed for a good level!

Special lever functions in the mode of creating object connections:

If the mode of object connecting is on, and if a lever is active as a source (i.e. if there's a line coming from it to the mouse cursor), you can, using the key  create/delete special level function – **the switch of direction or values**. This function is usable only for objects such as [transporting belt](#), [force fields](#), and [stone shifter](#) (see the descriptions of these objects), and the lever will afterwards change the direction of belt sliding or force field, alternatively it cyclically changes the stone type in a stone shifter. (In the standard mode a lever turns off/on these objects). For all other objects this function has no use or effect and lever with the direction switch mode turned on behaves for these objects the same as a standard lever.

The switch mode is indicated with a yellow square around the middle of a lever (a small square) from where a link connecting a lever with other objects comes.

4.4.8. Buttons



On the web pages and in older texts buttons are called “platforms”, however it is still the same element!



Buttons have the very same setting options of all parameters just like [levers](#), so everything that was mentioned in the lever description in the chapter above also relates to buttons.

Certainly, switching the lever position doesn't work (it would have no sense at buttons), but, on the contrary, you can change an orientation of a button ground/wall/ceiling – see below.

Object controls that are different to the lever ones:



+



- rotation of button positioning in a clockwise direction



+



- rotation of button positioning in a counterclockwise direction

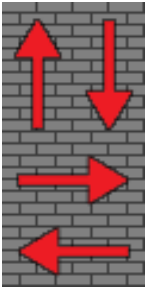


What concludes from the paragraphs above is that if you choose in a window an incorrect button type (for instance, the ground one instead of a ceiling one), nothing happens and using the rotation positioning you can easily turn that button to the desired position.



If two buttons are pressed at the EXACTLY same moment (in the same engine cycle) and both of them control the same objects, in the end their activities will mutually override! (One button turns on the object and the second one will therefore turn it off at the very same moment.) This also concerns the impulses which buttons send during their release.

4.4.9. Moving walls



This is another of basic elements of level construction. The abilities of its setting are really vast, so pay special attention to this chapter! And definitely do not overlook the warning placed at the end of this chapter.

In the window you can choose from four types of moving walls. The red arrow shows to which side a wall will **SLIDE OFF**, thus with this side (where an arrow points) the wall belongs somewhere “to the wall”.

Once you choose a respective wall in the editor window, the big red arrow will shift into a little triangular one which is always placed in the top left-hand corner of the wall, regardless of its size or slide direction.

A moving wall can only be controlled with levers and buttons.

Object controls in the placing or moving mode:



- change of the dimensions of a moving wall.



Dimensions of a moving wall can be changed **ONLY** if a wall is in the mode of object placing or moving. So if you want to change an overall dimension of an already existing wall in the level, click on it (in the standard editor mode) with a left mouse button. The editor will switch to the mode of object moving and now you can change dimensions of a wall using the arrow keys. Afterwards, place the altered wall with a repeated click of a left mouse button to the correct position in a level. If you, whilst changing wall dimensions in the mode of object moving, press the right mouse button or the key Escape, the wall will return back to its original place without any dimension alterations and the editor will revert to the basic mode.

Object controls:



- choosing an object for moving or deleting



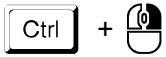
- switching the state of the wall slid/ejected (the slid state is indicated with a semi-translucent texture)



- switching the slide direction to the left/right at horizontally moving walls or top/bottom at vertically moving walls

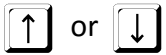


- switching between three types of a wall sound, it's indicated with the sign Sound 1/2/3 above the wall. The sound 1 is a classic stone wall, 2 is a “hydraulic voice” (for the walls of technical levels) and the sound 3 is suitable for chain bridges and gratings.



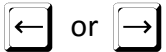
- switching MEM-OFF/ON, it's indicated with the sign above the wall. If MEM-OFF

(default) is set at a slid or blocked (for instance with a stone) wall, which was turned on and then turned off again, the wall WILL NOT MOVE after the obstacle is removed. On the contrary, if MEM-ON is set, the wall will "remember" the early and other signal from a lever/button, and once the obstacle is removed, it will carry out that action. I don't recommend using this function very often, and to let the setting at the default MEM-OFF, until you don't become real masters at level creating and until you find a sensible use for using the function of MEM-ON state.



- setting the thickness of a wall in a direction perpendicular to its movement (it's

indicated with a dark-blue shading as thick as the resultant wall)



- setting the limit of a wall slide (it's indicated with a yellow frame corresponding to

the size of the part of the wall which will not be slid)



These two last mentioned functions of parameter alterations of moving walls (limit of thickness and slide) could be used particularly whilst creating chain bridges or other vertically moving platforms. Usage at horizontally moving walls is disputable, nonetheless the editor allows changing all these parameters for all kinds of walls. Don't worry to experiment with these parameters, you will soon understand their logic.



- (Delete) removing an object, if the mode of object moving is active



- return to the basic editor mode, if the mode of object moving is active

Special functions of moving walls:



- turns on/off the style of a wall between a real and an "optic" one



What is an "optic wall"? Players of Quadrax X will surely remember miscellaneous gratings covering the exit or other passages or other moving elements covering active object. And this is exactly what an optic wall is. If it covers the entrance to the exit, passage or teleport, a figure won't be able to get there! Thus, it's an optimal tool for creating miscellaneous obstacles or optic illusions. An optic wall is always rendered in the foreground before passages/teleports, and therefore they are not visible behind it. (If, of course, the part of it doesn't contain translucent textures, such as at gratings). However, using optic wall is a "college" of Quadrax and not only does it require flawless construction of the level, but particularly good work in the graphics editor during the post processing of creating level textures, when this wall is needed to be placed correctly (especially, if it should be partially translucent). Hence, beginners shouldn't create levels with this kind of walls.

In the editor this wall is indicated with a big cross of yellow lines between the opposite corners of a wall. Whilst playing the level in editor mode, this wall is shown semi-transparently, so the objects beyond it are partially visible.



- turns on/off the style of “Pusher” (stone mover) style.



This style could be used for horizontally moving walls as high as two logical fields (they have to be the same size as stones). You can switch between a wall, a pusher (indicated with a big yellow arrow in the direction of a possible stone moving) and “force” pusher (indicated with a double arrow) by pressing the key P. There’s only one but significant distinction between a normal and “force” pusher: A “force” pusher can push a stone which is burdened with another stone! Its usage is particularly for pushing stone from vertical “reservoirs”. **Never use a force pusher where it is not necessarily needed!** If a top stone (which lies on a stone currently being moved with a pusher) doesn’t have on the side, where the bottom stone moves, a solid wall, it looks unrealistic! (The top stone doesn’t move!) Also if there was a figure standing on a stone being moved with a force pusher, it will slide below their legs! Keep this all mind, if you decide to use a force pusher, and assure that no such optic irregularities could happen.



Never place moving walls the way that one would cover another one, irrespective of their type! And not even when one is ejected, and the other one slid.

THE PATH DETERMINED FOR WALL MOVEMENTS CANNOT CROSS UNDER ANY CIRCUMSTANCES!

The editor doesn’t control this and it’s only up to the level creator to manage this. In case a crossing of paths of moving walls happened, it could cause unfathomable behavior of the entire engine when a level is launched, and it would surely disrupt the textures of moving walls! Despite the fact you can’t damage any data of the level or the game with it, don’t try it anyway!

4.4.10. Exit



The main finish place for the figures. It probably doesn’t need any further introduction. There can be a discretionary amount of exits in a level, and if all the figures are in one (one or more), the level is deemed finished (certainly, only if the figures have collected all the crystals contained in the level). There even needn’t be any exit at all, and the level would still be “playable”. Certainly, it won’t be able to be finished – this option is in the editor so that it would be possible to launch, play and debug level fragments or constructions of points.

Object controls:



- choosing an object for moving or deleting



or - cyclical change in the exit texture (7 forms)



- (Delete) removing an object, if the mode of object moving is active



or - return to the basic editor mode, if the mode of object moving is active

4.4.11. Crystals



A well-known and simple element. Crystals can be placed anywhere in the level, however in order to comply with the etiquette of good manners and good design, don't place them "into the air" or somewhere where they could happen to be in air. That's because it is an object that doesn't underlie the laws of gravity, and if you placed a crystal on a stone, for instance, once it's pushed, the crystal could be left "hanging" in space. So place it, if possible, on stable ground.

There can be a discretionary amount of crystals (until the limit of 255 is filled), but there needn't be any at all – just like in the level 90 in QX. It is not (unlike exit) a mandatory object of a playable level.

Object controls:



- choosing an object for moving or deleting



- (Delete) removing an object, if the mode of object moving is active



or - return to the basic editor mode, if the mode of object moving is active

4.4.12. Clue scroll



A well-known and simple element. The rules applying to placing clues are the same as for placing crystals. Unlike crystals, there can be clue in only one exemplar (the editor won't allow placing more than one object of this kind). If you don't place a helping scroll to the level at all, in the game there will be written on it (if shown by pressing F1): "There are no clue scroll available in this level."

Object controls:



- choosing an object for moving or deleting



or - cyclical change in the help texture (2 forms – lighter and darker)



- (Delete) removing an object, if the mode of object moving is active



or - return to the basic editor mode, if the mode of object moving is active

4.4.13. Global lighting



A special object "Global lighting" is controlled by levers or buttons. If this object is placed in a level, the level will support the function of "getting dark". If a lever or button is activated, which are connected to this object, the light will turn on/off in the level depending upon the primary setting of this object. Just like help, there can be only one exemplar of this object in a level. Whilst playing, the object is not visible.

Object controls:



- choosing an object for moving or deleting



- switching between the ON/OFF mode – if the mode ON is set, the level has the light on once started, if it's set to OFF, the level starts "in darkness"



- (Delete) removing an object, if the mode of object moving is active



or - return to the basic editor mode, if the mode of object moving is active

4.4.14. Sign of open edges



This special object was created for the tenth episode of QX, which, however, didn't get to the game. It turns on the function of so called open edges in a level. What does it mean? In the normal, classic setting a level edge is deemed a solid wall, which cannot be seen, but still encompasses the level from all its sides. Therefore, figures can even walk on the bottom of a level, just like stones will stop if they get pushed to an edge or fall to the very bottom of a level, even if there is no editing wall.

The function of open edges changes it. The level edge is now deemed that there's nothing beyond it, just empty space. Falling stones, stones pushed to the level edge – this all will irreversibly disappear in emptiness. Even a figure, if it doesn't watch out, can fall out down "outside the level" (what's deemed that it falls from a really huge height and is hereby instantly killed).

The function of open edges has its restrictions, though. Only stones can fully go beyond the edges, figures only when falling (so a figure can push a stone beyond the "edge" of the level, but won't get beyond it itself). For all other objects the edges of a level work the same as in the classic setting.

For enable the function of open edges, just place this object anywhere in the level. The object itself doesn't react with any other object in the level, and is not visible whilst playing.



This object and levels with the value of open edges should be used carefully. It was designed for the episode 10 of QX, which was supposed to resemble Laputa. So use it if you create only levels "in air".

Object controls:



- choosing an object for moving or deleting



- (Delete) removing an object, if the mode of object moving is active



or - return to the basic editor mode, if the mode of object moving is active

4.5. Stones (F3)

Stones are the main element of Quadrax – after all, from stones is derived his name (quadratic - Quadrax)... ;-)

For a better explanation, there's a numbering of respective types of stones in the following picture, the way they are shown in the window of the editor, and to these numbers the explanations related to respective types of stones will assign:



I assume the main values of all kinds of stones (apart from the anti-gravity one) all my players of Quadrax are familiar with, so I'm **not going to describe them here howsoever**.

4.5.1. Regular stones

They're stones with the numbers 1 – 4. 1, 2 and 4 are classic stones, differing from each other only with their basic texture (uncovered/mossy/dark). Each of them has fifteen other thematically same textures, which are chosen with arrow keys in the mode of modification of active object (so when a mouse cursor above the stone is translucent and the stone pulses).



This change of bitmaps (textures) of object generally applies to all objects than can have more forms and are always carried out using arrow keys left/right in the mode of modification of active object (so when a mouse cursor stands above the object).

Object controls:



- choosing an object for moving or deleting



- cyclical change of the stone texture (15 forms for each kind of stone)



- (Delete) removing an object, if the mode of object moving is active



- return to the basic editor mode, if the mode of object moving is active

4.5.2. Metallic stones

Stones with numbers 5 and 6. When pushed, they make a metallic voice and therefore are intended mainly for technically designed levels. Each of them has 30 different texture varieties.

Object controls:



- choosing an object for moving or deleting



- cyclical change of the stone texture (30 forms for each kind of stone)



- (Delete) removing an object, if the mode of object moving is active



- return to the basic editor mode, if the mode of object moving is active

4.5.3. Explosive stone

The stone with number 7. An old well-known explosive stone.

Object controls:



- choosing an object for moving or deleting



- cyclical change of the stone texture (3 forms – normal/mossy/dark)



- (Delete) removing an object, if the mode of object moving is active



- return to the basic editor mode, if the mode of object moving is active

4.5.4. Fragile stone

Number 8. Fragile.

Object controls:



- choosing an object for moving or deleting



- cyclical change of the stone texture (4 forms – normal/mossy/dark/volcanic)



- (Delete) removing an object, if the mode of object moving is active



- return to the basic editor mode, if the mode of object moving is active

4.5.5. Wooden stones

Stones with numbers 9 – 11. The stones No. 9 have Japanese characters carved on them, stones No. 10 are the same but without the characters and stones No. 11 are the same as the stones No. 10 but in a darker variety. Each of these three kinds of wooden stones has 15 different textures.

Object controls:



- choosing an object for moving or deleting



or - cyclical change of the stone texture (15 forms for each kind of stone)



- (Delete) removing an object, if the mode of object moving is active



or - return to the basic editor mode, if the mode of object moving is active

4.5.6. Anti-gravity stone

The stone No. 12, originally intended for the **episode 10** in Quadrax X. It has totally unique attributes and it has never been used in any other Quadrax before! It has virtually the same values as a regular stone, but reversed gravity applies to it! Thus, in open space it won't fall down but rise above! Also it doesn't burden any stone below it (as it logically pushes the top edge).



To fully handle to use all the values of anti-gravity stones means successfully handling the work with all other stone types. **The proof of that is that creating and testing the episode 10, which was supposed to contain these stones, was such a difficult and long process that I myself gave up on it!** Do not start constructing levels containing anti-gravity stones, until you create tens of good classic levels. Certainly, this is just a recommendation, you can fiddle with them in the editor, but without proper experience you may not be able to manage to create a sensible level using anti-gravity stones. Originally I didn't intend to add this type of stones to QX EFF at all, but in the end (due to Quadrax Neverending) you can encounter them here.

Object controls:



- choosing an object for moving or deleting



or - cyclical change of the stone texture (3 forms – normal/dark/mossy)



- (Delete) removing an object, if the mode of object moving is active



or - return to the basic editor mode, if the mode of object moving is active

4.5.7. Slippery volcanic stone

No. 13. Unlike previous stones it doesn't have a selection of texture, as it's fully animated. When moved, it makes a specific "crunchy" noise.

Object controls:



- choosing an object for moving or deleting



- (Delete) removing an object, if the mode of object moving is active



- return to the basic editor mode, if the mode of object moving is active

4.5.8. Slippery metallic stone

No. 14. When moved, it makes a specific creaky voice. It has 15 different kinds of textures.

Object controls:



- choosing an object for moving or deleting



- cyclical change of the stone texture (15 forms)



- (Delete) removing an object, if the mode of object moving is active



- return to the basic editor mode, if the mode of object moving is active

4.5.9. Magnetic stone

No. 15. It makes a metallic voice. It doesn't have an option of selecting other texture.

Object controls:



- choosing an object for moving or deleting



- (Delete) removing an object, if the mode of object moving is active



- return to the basic editor mode, if the mode of object moving is active

4.5.10. Metallic stone for magnets

No. 16. When pushed, it makes a metallic voice. It doesn't have an option of selecting other texture.

Object controls:



- choosing an object for moving or deleting



- (Delete) removing an object, if the mode of object moving is active



or - return to the basic editor mode, if the mode of object moving is active

4.5.11. Levitating stone

No. 17. It doesn't have an option of selecting other texture.

Object controls:



- choosing an object for moving or deleting



- (Delete) removing an object, if the mode of object moving is active



or - return to the basic editor mode, if the mode of object moving is active

Hereby we've come through the descriptions of all kinds of stones. The last position in the window of stones is unused and clicking it really won't make any other stone. ;-)

4.6. Movers/Ports (F4)

This window allows you to select from objects that somehow manipulate with stones or allow moving of the figures. In the following text they'll be described in the order in which the items occur in the window from left to right and from top to bottom.

4.6.1. Figure passages



Once placed in a level, each passage has a target object defined as itself. In the area of an entrance of passage you can see two squares in the mode of connection creating, the top one of which is an entrance and the bottom one is the target. Therefore, the connecting line goes from the top square to the bottom one at a newly created passage.

Each passage can have only one connection with another passage (start - target) or with itself. (So there can be only one way leading from each passage – either to another passage or back to itself.)

One passage, however, can be a target for more entrance passages!

Object controls:



- choosing an object for moving or deleting



or - cyclical change of the passage texture (8 forms)



- (Delete) removing an object, if the mode of object moving is active



or - return to the basic editor mode, if the mode of object moving is active

4.6.2. Figure teleports



They have the exactly same rules for placing into a level and creating connection like [Figure passages](#).

Object controls:



- choosing an object for moving or deleting



- cyclical change of the teleport texture (3 forms – normal/rusty/sci-fi)



- (Delete) removing an object, if the mode of object moving is active



- return to the basic editor mode, if the mode of object moving is active



If a teleport leads to another teleport, signal lights above its entrance will blink red-green (or a green stripe at the sci-fi texture). If it leads to itself, these lights are red. This is managed directly by the engine of the game and it's not necessary (and possible) to set it.

4.6.3. Entrance stone teleport



This is a special object defining the position of an entrance place for a stone teleport. Its depiction in the editor differs from its real look in the game, and it will be described below.

If you hover with a mouse cursor above the object of entrance stone teleport, an additional three-row-long text will show up above it indicating the state and the type of a teleport.

The description of meanings of respective signs:

- ON/OFF – indicates whether a teleport, after a level is launched, is turned on/off (set ON by default)
- SHIMMER – whether a teleport will “buzz” and shimmer like a broken fluorescent tube
- HORIZONTAL/VERTICAL – horizontal or vertical placing of teleport sides
- GRAPHIC 1/2/3 – the type of used texture (technical/ancient dark/ancient light)

In the mode of object connecting, it's possible to connect an entrance teleport with one outlet teleport.



Object controls:






- choosing an object for moving or deleting






- switching ON/OFF

 +  - turning on/off the value SHIMMER

 +  - switching HORIZONTAL/VERTICAL

 or  - cyclical change of the texture GRAPHIC 1/2/3

 - (Delete) removing an object, if the mode of object moving is active

 or  - return to the basic editor mode, if the mode of object moving is active



Teleport sides are shown only when a level is played! And this depends upon the setting of parameters HORIZONTAL/VERTICAL and GRAPHIC 1/2/3. Also sides are set to be a solid wall! Therefore, it's necessary to consider this option of entrance teleports and adjust a level design to it.



If an entrance teleport is not connected with any other outlet teleport, it certainly won't react to a stone that would get into it. The same applies at the moment when to the outlet teleport a teleportation from another teleport is being carried out when more than one entrance teleport is connected to the outlet teleport. Also, when a place of an outlet teleport is blocked with any object apart from figures, the teleportation won't initiate.



When a teleportation initiates, but during it the outlet teleport gets blocked (before a stone appears in it), the stone will be blocked in "out-space" and waits. When the outlet teleport gets released, the teleportation will finish.

4.6.4. Outlet stone teleport






Outlet (target) point of a stone teleport is a place to which a stone from an entrance stone teleport teleports, according to the mutual connection of these teleports. Outlet point of teleport can be connected with a discretionary amount of entrance teleports.

An outlet teleport can also be connected with a stone shifter – in the mode of connection selecting, click with a left mouse button on an outlet teleport and afterwards click on a stone shifter. This way, you'll create a connection between these two objects and the type of outlet stone from the teleport is then dependent upon the state of the stone shifter (explained in detail in the chapter below).

Object controls:

 - choosing an object for moving or deleting

 - (Delete) removing an object, if the mode of object moving is active

 or  - return to the basic editor mode, if the mode of object moving is active



If there's more than one entrance teleport connected to the outlet teleport, then during the parallel request to teleport the stones the stone that was placed in a level first will be teleported first! It's the value of the game engine when a cycle FOR is carried out to gradually process the request. The entrance teleport whose turn is first, is initiated and the other teleports have to wait.

4.6.5. Stone shifter



This special object allows changing a stone type in an outlet teleport. After creating a mutual connection (see the previous chapter), the stone type in the outlet teleport is influenced according to the state of the stone shifter these ways:

- If a shifter is turned off, no change will happen to the teleported stone,
- If a shifter is turned on, the type of the teleported stone will change during its outlet from the teleport to the type graphically shown in the stone shifter.



A shifter can be controlled by levers and buttons, and it depends upon the value direction or value switch of a respective controlling element. When it comes to a regular lever or button, they only turn on/off the changer. But if a lever or button has the value direction or value switch defined, they control switching the stone type inside the changer. So if you want to control, just like turning on and off the changer, also the stone type inside, it's needed to always have two controlling elements assigned, and one of which must be standard and the second one must have the **function direction or value switch** set!

Stone shifter has rather rich options of setting, so it's needed to thoroughly study the following information.

In the mode of object modifying an informing text created from three rows will show above the shifter. **The first line** contains a sign ON/OFF, whether a shifter is turned on/off during a level start (ON by default). Behind that sign there's a number of stone type which should be set by default in the level start. Respective numbers of the stones mean:

- 1 – Classic
- 2 – Explosive
- 3 – Brittle
- 4 – Wooden
- 5 – Slippery
- 6 – Magnetic
- 7 – Metallic for magnets
- 8 – Levitating
- 9 – Anti-gravity

In the same order a stone type changes cyclically during a level playing controls of the change of a stone type with a lever or button depending on which types of stones are allowed (see below).

In the **second** line you'll see nine digits. Each of them indicates which type of **main texture** the stone shifter should be used for respective types of stones. Position of the digits (from the left) corresponds with the number of the stone type (1 - 9) stated above and is controlled with keys  1 to  9. For each kind of stone there's a different amount of texture selection!



If a digit has a value of the texture 0, that stone cannot be assigned any type of texture!

Description of the texture digits for a respective stone, created in an outlet teleport using a shifter:

Classic (position 1 from the left)

- 1 – Stony normal (in the further description of other stones only as “normal”)
- 2 – Stony covered with moss (in the further description of other stones only as “mossy”)
- 3 – Concrete
- 4 – Stone dark (in the further description of other stones only as “dark”)
- 5 – Metallic normal
- 6 – Metallic rusty

Explosive (position 2 from the left)

- 1 – Normal
- 2 – Mossy
- 3 – Dark

Fragile (position 3 from the left)

- 1 – Normal
- 2 – Mossy
- 3 – Dark (with blue shades – for technical levels)
- 4 – Volcanic

Wooden (position 4 from the left)

- 1 – Normal with Japanese characters
- 2 – Normal
- 3 – Dark

Slippery (position 5 from the left)

- 1 – Tinny
- 2 – Volcanic

Magnetic (position 6 from the left)

0 – This kind of stone cannot have its texture changed

Metallic for magnets (position 7 from the left)

0 – This kind of stone cannot have its texture changed

Levitating (position 8 from the left)




0 – This kind of stone cannot have its texture changed

Anti-gravity (position 9 from the left)

1 – Normal

2 – Mossy

3 – Dark

The last **third line** indicates whether a set stone type should occur in the shifter. If so, the digit value is **1**, otherwise it's **0**. Switching between values is controlled using  +  to . Each type of stone has 1 set by default. The order of digits corresponds with the stone type, just like setting textures.

Object controls:



- choosing an object for moving or deleting



- switching ON/OFF turning on/off the shifter during a level start



- change of the stone number that will be set by default during a level start



- gradual cyclical switching of the texture for respective types of stones



- switching options of stone appearance in the shifter



- (Delete) removing an object, if the mode of object moving is active



- return to the basic editor mode, if the mode of object moving is active



Unlike force fields and belts, the setting element for switching of stone types HAS NO influence, if a shifter is turned off!



Within one type and main texture, a specific texture for a stone is chosen randomly in the interval which is set for that type of stone and texture. I.e., for the type of stone 1 and main texture 1 (normal stone) one of fifteen textures of this stone type will randomly choose.



During a level playing, the items turned off in the shifter (which have in the third row on their position number 0) are automatically skipped.

4.6.6. Force fields



Force fields have the same options of connecting with other elements as shifter (see the chapter above). A regular lever/button turns on/off the field, a lever/button with a defined value direction or value switch allows changing a direction of the field to the right/left.

In the mode of object modify, informing text created from two parts will show above the force field:

- ON/OFF – defines the state of the field turned on/off during a level start
- RIGHT/LEFT – the starting direction of a field to the right/left during a level start

Object controls:



- choosing an object for moving or deleting



- ON/OFF turning on/off the field during a level start (ON by default)



+ - switching RIGHT/LEFT move of the field during a level start



- (Delete) removing an object, if the mode of object moving is active



or - return to the basic editor mode, if the mode of object moving is active



Force fields can be arranged on one another and therefore create a bigger field like 2 logical cubes. The fields, however, mustn't overlap each other (so the resultant height of a field must always be a multiple of the height of one field) and all must be connected with other setting elements!



Unlike stone shifters, the setting element for switching the direction of the field does have influence, even when a field is turned off!

4.6.7. Lifts



Another standard object which has been here with us since the very beginnings, and where any further description is futile... A lift can be controlled with levers or buttons.

Object controls:



- choosing an object for moving or deleting



- switching of initializing direction of a lift during a level start (indicated with a little yellow triangular arrow in the middle of a lift).



or - cyclical change of a lift texture (11 forms)



- (Delete) removing an object, if the mode of object moving is active



or - return to the basic editor mode, if the mode of object moving is active

4.6.8. Entrance stone mover



Once again, another of already classic objects. Just like at teleports, you connect the source mover with a target with one, and just like at teleports, a target teleport can have more entrance ones assigned to it. And at this object also applies the rule “first come, first served”, just like at stone teleports.

Object controls:



- choosing an object for moving or deleting



or - cyclical change of a mover texture (4 forms)



- (Delete) removing an object, if the mode of object moving is active



or - return to the basic editor mode, if the mode of object moving is active

4.6.9. Outlet stone mover



Once again, another of already classic objects. Just like at teleports, you connect the source mover with a target with one, and just like at teleports, a target teleport can have more entrance ones assigned to it. And at this object also applies the rule “first come, first served”, just like at stone teleports.

Object controls:



- choosing an object for moving or deleting



or - cyclical change of a mover texture (4 forms)



- (Delete) removing an object, if the mode of object moving is active



or - return to the basic editor mode, if the mode of object moving is active

4.6.10. Crane




Simply a crane. It raises any stone and be controlled with levers/buttons.

Object controls:



- choosing an object for moving or deleting



or  - cyclical change of a crane texture (7 forms)



- (Delete) removing an object, if the mode of object moving is active



or - return to the basic editor mode, if the mode of object moving is active



A crane grappler isn't visible during level editing. Once a level is played (even in the editor mode) the grappler will show up.

4.6.11. Magnet



Very alike to crane, however it lifts only metallic stones for magnets. Using magnetism. Incredible... ;-). In the mode of object modifying, a sign ON/OFF is shown above the magnet indicating in which state the magnet will be during a level start (set OFF by default).

Object controls:




- choosing an object for moving or deleting



- switching of the state OFF/ON in which the magnet will be during a level start



or  - cyclical change of a lift texture (2 forms)



- (Delete) removing an object, if the mode of object moving is active



or - return to the basic editor mode, if the mode of object moving is active

4.6.12. Transporting belt



It's used for horizontal transport of stones. A transporting belt has the same setting of its length as the setting of the size of moving walls is. This means that its length can be changed (using arrow keys left/right) only if an object is in state of inserting or state of moving. Text information about its settings is also shown above the belt.

- ON/OFF – the state of belt during a level start (OFF by default)
- RIGHT/LEFT – initializing direction of belt during a level start (RIGHT by default)

Transporting belt can be controlled with levers and buttons, and it depends upon the value direction or value switch of a respective controlling element. When it comes to a regular lever or button, they only switch on/off the belt. If, however, a lever or button has the value direction or value switch defined, they can also control switching the direction of belt movement. Thus, if you want to control with turning on and off a belt also its movement direction, it's needed that it has always two controlling elements assigned, where one of which must always be standard, and the second one must have a function direction or value switch set!

Object controls:



- choosing an object for moving or deleting



- switching OFF/ON of a belt during a level start



- switching RIGHT/LEFT belt movement during a level start



- change of the length of a belt if the mode of object moving is active



- (Delete) removing an object, if the mode of object moving is active



- return to the basic editor mode, if the mode of object moving is active



Unlike stone shifters the setting element for switching the direction of a belt does have influence, even when the belt is off!

4.7. Mortal (F5)

This window of the editor contains “mortal” objects. This means the object that could harm the figures – by killing them, what will harm them a lot. ;-)

4.7.1. Arrow shooter



Can be launched with a lever/button. In a level it's always deemed a SOLID WALL!

Object controls:



- choosing an object for moving or deleting



- switching direction right/left




- (Delete) removing an object, if the mode of object moving is active



or - return to the basic editor mode, if the mode of object moving is active



The engine allows a level to process up to twenty active (flying) arrows or lasers in a level at one time! If there are currently twenty arrows/lasers “in the air”, a next arrow shooter will not be activated!

Whilst playing in the editor, an arrow shooter is shown as a special wall , whilst playing in the playing mode it is not visible, and thus it must be correctly textured!

4.7.2. Laser



Can be launched with a lever/button. Unlike an arrow shooter, it IS NOT deemed a solid wall in a level!

Object controls:



- choosing an object for moving or deleting



- switching direction right/left



- (Delete) removing an object, if the mode of object moving is active



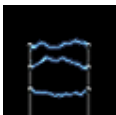
or - return to the basic editor mode, if the mode of object moving is active



The engine allows a level to process up to twenty active (flying) arrows or lasers in a level at one time! If there are currently twenty arrows/lasers “in the air”, a next laser will not be activated!

Whilst playing in the editor, a laser is shown with the same image as during the editing, but whilst playing in the playing mode it is not visible, and thus it must be correctly textured!

4.7.3. Electric fence



Can be controlled with a lever/button. Above it there's an informing text ON/OFF shown, which indicates whether a fence is on during the level start (ON by default).

Object controls:





- choosing an object for moving or deleting



- switching OFF/ON during a level start



- (Delete) removing an object, if the mode of object moving is active

 or  - return to the basic editor mode, if the mode of object moving is active

4.7.4. Mortal flame



It has no special values or options of setting apart from that it can burn a figure to soot, if not quenched with a stone.

Object controls:



- choosing an object for moving or deleting



- (Delete) removing an object, if the mode of object moving is active



- return to the basic editor mode, if the mode of object moving is active

4.7.5. Target



This special object in a shape of a green target is a novelty in QX EFF. It is used for detecting a hit of arrows/lasers to a specific place and for controlling other objects. If an arrow or laser during its flight hits such an object, the target will behave the same as a lever or button switch!

Even though, this object isn't any mortal for the figures, it is listed in the window of objects Mortal, as it's directly dependent upon arrows or lasers with which (and only with them) it can react.



Creating a connection of this object with other controlled objects is exactly the same as creating a connection at levers and buttons.



Whilst playing in the editor mode, the target icon is visible just like during level editing. Whilst playing in a regular playing mode, however, the target symbol IS NOT shown – so it's only up to the level creator to create some meaningful graphics on that place!

4.8. Animations (F6)

The window of animations that will pop up could scare with its size. But you needn't be afraid. Most of animations have exactly identical behavior and setting. So, I'm not going to describe settings of each single object of animations (like describing all approximately 75 animations would take a rather huge amount of pages), but I'll describe the summary values, and then I'll talk about few non-standard animations that have expanded/other options of setting or connectivity.



Animations to a level should be placed after its final debugging and testing. They're only additional objects with no influence to the playability of a level, they only "revamp it" and visually enrich.

4.8.1. Basic controls

All animations are classic objects, so these basic functions will always work at them:



- choosing an object for moving or deleting



- (Delete) removing an object, if the mode of object moving is active




or - return to the basic editor mode, if the mode of object moving is active

Most of animations have two parameters of setting (the exact position in a level and animation speed) which are shown as text information above the animation. First, it's two two-digit numbers which are set by default at each animation to the value 00, and eventually the third number of two digits, which is in a bottom row below the first two and always contains a value **bigger** than 00.

4.8.2. Shift of animations in the axes x & y

Two values in a top row mean shift in axes x (first number), and y (second number) in pixels in comparison with the basic raster 30x30 pixels to which all other objects are inserted. With shifting of animation you can exactly set its position (to pixel) towards the level texture. Animation can be shifted in both axes by up to 29 pixels (what makes sense as on the thirtieth pixel there's another beginning of a basic raster).

Animation shift is carried out in the mode of object parameter modifying – so when a mouse cursor is above the animation (it's semi-translucent and the animation pulses) – simply by pressing and holding the key  and pressing cursor arrows



+



- animation shift in pixels in axes x and y



Exact animation positioning is handy particularly at flames when you need a flame to fit exactly with the upper boundary of a pot (or cauldron) in which it glows. Using this function, you can also set for a flame a specific location in a level.



If an animation DOES NOT HAVE these two numbers shown above it, its correcting shift is not possible.



BEWARE! Even if you shift an animation, for the editor its existence is still deemed to be there where it was originally placed without shifting, i.e. in the alignment to the grid 30x30 pixels. In case your animation seems shifted too much (e.g. by 25 and 25px), it might seem to you that a mouse cursor above it doesn't react (particularly at small animations). All it takes is move a mouse cursor to the place where the animation is really "in data" placed and everything will be alright (now it might seem that a cursor is outside the animation). This little optic snag is, though, only a little cost for the option of being able to utterly precisely set the resultant position of animations.

4.8.3. Animation speed setting

Third number (which is shown **below** the values of animation shift) sets the speed of animation in the cycles of the game. Each animation has its own limits of speed depending on its average framerate. This number sets how long (how many cycles of the game) the animation will be waiting, before it shows another frame of its animation. Also the same rule applies that a lower number means faster animation movement and vice versa. The alteration of the value of that number is easily carried out in the mode of object parameter modifying with a single press of arrow keys up or down.



- modification of the number of animation speed




It's good to set animation speed as the very last thing during level creation (after it's been textured) according to the personal feeling from how an animation feels in a level.





Each animation has its own pre-set limits of minimal and maximal speed and these limits cannot be overlapped during setting. If an animation DOES NOT HAVE a number of animation speed shown, the speed animation is set without being able to be changed.

4.8.4. Animations without animations (sounds)

There are three special types of animation that have no visual look. In fact, it's only voices that are placed where you want the sounds to sound. They are respectively:

-  Sound of transformer 1 (place to electrical substations, big electric devices etc.)

-  Sound of transformer 2 (higher tone and slightly more silent than 1, can be used as a sound source of broken fluorescent tubes and alike devices)
-  Sound Geiger (snapping of Geiger's computer, place to the places that could be a source of radiation)

These three special sound “animations” are certainly not visible whilst playing a level and only a respective sound can be heard on their place.

4.8.5. Paper wall



This animation actually isn't an animation either, as it's not “animated”. Nonetheless, it's created as a “one-sprite” animation. Paradoxically, unlike most of other animations, it has a higher amount of parameters that can be set. Also, unlike other animations that are in the background beyond all active objects, this object is in foreground and therefore covers all other active objects.

In the mode of object placing and moving a paper wall can have its dimensions changed simply (just like at moving walls) using arrow keys.

After placing a wall in a level, it is possible in the mode of object parameter modifying using left and right arrow keys to change the visualization of a stronger side on the left or right side of a paper wall. Above paper walls, there's always indicated with informing texts LEFT and RIGHT which side will be shown whilst playing the game. **This function is not available for a wall as high as one field.**



Paper walls have limited maximal size. Even so, they can be easily connected to seamless tiles and create a discretionarily big area with a discretionary shape. Just don't forget to correctly set side parameters to walls!

4.8.6. Bulky generator



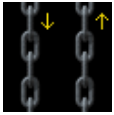
This animation works entirely on its own. If there's an object [global lighting](#), placed in a level, then according to its state it manages its animation. If a light is off in a level, the generator idles. After a light is turned on, it gradually accelerates and if it is turned off, it gradually ceases to move. There's no need to create any connections, the work of generator is entirely automatic and dependent solely upon the state of the object of global lighting.

4.8.7. Light bulbs



These animations are also dependent on the object [global lighting](#). If a light is off in a level, the bulbs are off as well, and vice versa.

4.8.8. Moving chains



These animations can in the placing or moving mode have their height changed using arrow keys up/down.

4.8.9. Clock



This animation has no special settings. Even so, clock always shows the current system time on the computer where the game runs ;-). Animation exists in four varieties differing only with the direction of lighting and therefore with the shadows on a dial.

4.8.10. Water of fountains



graphics.

These animations contain only the animation of water of fountains and are intended for advanced users who create custom level textures. The respective texture of the fountain must be added to the source files of level

4.8.11. Luminous beacons



For the end I've kept probably the fanciest animation ever – light beacons. Despite its littleness, this animation has vast options of editing! You will understand them once you switch the editor to the mode of connection creating between objects. That's because beacons are the only interactive animation and can be connected with virtually any other active (moving) object. So, whilst playing, a beacon monitors the state of an object, and if it starts moving/working, the beacon will start blinking! Particularly at technical levels this means livening the entire level!

4.9. Create bitmap

If the editor is in the basic mode, clicking this item of the menu **will save all the changes in the level** and a special editor of level textures will start up where you can carry out basic texturing of an entire level to the way it will look like in a regular gaming mode.

The complete description of the controls of this editor is listed in the chapter [Level texture editor](#), where you can find all information about work in this special editor.

4.10. Lock level

This choice in the level editor will lock away the entire level and prevent it from being altered or changed. In addition, locking will automatically erase utterly all saveslots (reasons for this will be explained subsequently).

This choice is ensured with a warning dialog box whether you really want to carry out this step. If you really wish to lock away the level, click with a mouse on the button OK, the level will be locked away and all the saveslots will be erased. All the items of the top menu, apart from Play and Create bitmaps, will disappear. Also in the right-hand bottom corner a red symbol of a padlock will show indicating that the level has been locked. If you wish to unlock a level, click this padlock symbol and in a subsequent dialog box click on OK.

If a level is locked away, in the insight of all levels in the main level editor there will be a small red icon of a padlock in the middle of it. **A locked level cannot be manipulated in the main editor and it also protects it from being rewritten. (Which is very handy!)**

What is level locking actually good for?

First, it's a protection before unwanted damaging of your ready and debugged level. Certainly, the main meaning of locking a level is for the case that you **want to publish your project publicly for other players!**

4.10.1. Level playability check

In order to prevent unplayable levels from getting to the publicly published projects, levels that you want to publish in a project need to be checked for their playability. And this is conducted by the author himself by playing and finishing a locked level in a regular playing game menu. (So NOT in the editor!)

Once, a locked level has been played and finished in a regular gaming mode, it will be highlighted as a playable one. The padlock icon will color green indicating that the level has been finished in a regular playing mode. This is also the reason why locking a level erases all the saveslots that could contain positions on a level that was being worked on. So the level must be finished without using any old saveslots. Certainly, whilst playing a locked level in a gaming mode you can use new saving and then loading from saveslots just like in a regular game. But this assures that data from the saveslots have been made from already locked version of the level.

Once the project contains at **least one** playable level, it is possible in Project manager to export it in a special mode, where a second player who imports his project to his QX EFF will have access only to playable levels and will have to subsequently finish and solve all playable levels in the project (so it will be alike to playing any other full Quadrax). Further information about functions of level exporting is listed in the chapter [Export project](#).

A locked level which has been marked as playable can be unlocked in the editor by its author at any time – and it's entirely the same as with regularly locked level by clicking a green padlock icon in the right-hand bottom side in the level editor. A dialog box will follow asking whether you really wish to unlock a level which has been marked as playable. It is apparent, because if you unlock a level marked as playable, this mark will be erased! If you want to make a level marked as playable after unlocking (and additional alterations) you need to go through all the steps of locking a level and finishing it successfully in a gaming mode.

4.11. Clear level

If you click this menu item in a basic editing mode, a dialog box will show up asking whether you really wish to erase all level data. After clicking OK, all the data from the level, saveslots, level textures and statistical data will be erased. (Reach the state of an utterly new level.)




Be very cautious with this choice! All the level data will really be **irreversibly** rewritten with data of a clear level and there's no way to recover them afterwards! So always think very thoroughly if you **really wish** to erase all level data.

4.12. Special editor functions

Here you can find description of some special editor functions that make creating levels easier.


4.12.1. Moving an entire level

If the level editor is in the basic mode, then pressing and holding the combination of keys CTRL + ALT + SHIFT on the **left** side of the keyboard will switch the editor to the special mode of level moving. On the screen there will be a symbol **MOVE** , indicating the mode of level moving. Whilst continuously holding the combination CTRL + ALT + SHIFT you'll be able to, using arrow keys, move the entire content of the level to the left/right/top/bottom.



BEWARE! Whilst all static objects (walls, trap-cellars, ladders, ropes, lift block) can be rolled and are shown when level edges are overlapped automatically on the other side, active objects are not capable of doing this, and once a level edge is reached, they retain on the same place, i.e. on the level edge. During big movements using this function, it can easily occur to you that all active objects will "congregate" in one stack. So be wary whilst using function and shun situations when some of active objects get to the level edge.

4.12.2. Level mirroring

If the level editor is in the basic mode, pressing and holding the combination of keys CTRL + ALT + SHIFT on the left side of the keyboard will switch the editor to the special mode of level moving. (Just like in the previous chapter.) However, instead of using arrow keys, press the key . The level will be completely mirrored! (Tilted according to the middle vertical axis.)



Mirroring level is intelligent and will correctly assign to all directional objects (direction of switching of levers, belts, force fields, etc.) these directional parameters also in a mirror way. Thus, it's a complete level mirroring including all parameters of its objects.



Make the choice of moving or mirroring a level before you start texturing it! The editor of texture doesn't allow to make similar operations and the data structure wouldn't fit with the optical one! **In general, a rule applies that you should first fully finish and debug a level in the level editor, then lock it away, and lastly carry out level texturing.**

If you have already carried out in the texture editor the rendering of ladders or ropes to some of the layers, it's likely that both these functions (movement and mirroring) are blocked!

4.12.3. "Object copying"

If the level editor is in the basic mode, hover with a mouse cursor upon any **active** object, press and hold the key **[C]**, and then click with a left mouse button on the object. The original object will remain on its original place and the editor will switch to the mode of inserting a new object. **However, it is not legitimate copying of an object with all its parameters!** (That's why the name of the chapter is in quotation marks.) The editor will only switch to the mode of inserting a new object of the same type, just like if you just chose that object from menu windows of objects.



This function is intended mainly for speeding up the working process for advanced builders, if they want to insert another object of the same type to a level that already exists in it, they needn't open the respective window of groups of objects where that object is placed from the menu, but using this function they will choose a respective object as if they chose from the menu window.

4.12.4. Visual display of the ancillary raster grid

Pressing the key **[G]** (for **Grid**) will show in the editor the ancillary raster grid with lines as large as 30x30 pixels that precisely corresponds with the positions of logical cubes in a level. It's basically a coordinate web of positions of logical fields in a level. This grid can be turned on in any mode of the editor apart from the gaming mode. The grid is a good tool for precise placing of objects that are to have a specific position in comparison with another object in the level (e.g. inlets and outlets of stone teleports etc.). The grid can be turned off at any time by pressing the key **[G]**.

4.13. Careful with the saveslots

It's vital to remember one basic and essential fact whilst working in the editor. **Complete** data are always saved into the saveslot together with information about all static and active objects in the level and their current state at a moment a saving to the saveslot is used. So, if you make any changes in the level, they will not show up in the previously made existing saveslots! So if you load an older saveslot whilst working in the editor, **the level will be loaded in the exactly same state it was in when the saveslot was made regardless of the current look of the level!** So always consider this when using saveslots!

Whilst playing in the editor mode, all the saveslots are “infinite”, i.e. you can write data to them at any time. Each saveslot, however, may contain only one record, so each new writing to the respective saveslot will always rewrite the old data from the previous save. Whilst working in the gaming mode, saveslots work the same as usual, i.e. writing to one saveslot can be used within one play only once.



Saveslots whilst playing in the editor mode are saved to different files than saveslots played in the regular gaming mode, and are, therefore, independent. But the same rules apply for saveslots created in the regular gaming mode as for the saveslots from the editor. If you created a saveslot at some version of the level, after the data from it has been loaded, it will be in the exactly same state it was in when a creation of the saveslot happened, irrespective of its current look. Now it's hopefully clearer for you why all saveslots are erased, once the level has been locked away.

4.14. Level statistics

When a level is being developed, all the steps, times, launches etc. you've made during a process of level development are counted to the statistics. Now and then it's interesting data. But if the level is played in the level editor mode, the data of best steps/times and values of the first level solution is never enrolled, once it's been finished. That's logical, because when a level is being debugged, these values can rapidly fluctuate and wouldn't make any sense. Data enrolment about finishing of the level and the best steps/times will happen only if the level is locked away, and it is played in the regular gaming mode. Although, if you unlock the level once again, this data (best steps/times, steps/times until the first finishing) will be reset!

But if you play a locked imported project, all data of statistics work the same as in the regular full Quadrax game.

4.15. Finishing work with the level editor

If the level editor is in the basic mode, pressing the key Escape will carry out **automatic save of all the changes in the level** and the return to the main levels editor will follow. If you **don't wish** to save any made changes, press the combination of the keys SHIFT + Escape. The return to the main level editor follows **WITHOUT** saving the changes in the level. Please, pay attention to the information below!



Keep in mind that if you've launched playing of the level in the editor mode or launched the level texture editor, both these functions before their launch **always save the level data automatically!** So if you, for instance, carry out some changes in the level, then test it out by launching the playing in the editor, and in the end leave using SHIFT + Escape, the state of the entire level will be the same as it was at the moment when the gaming in the editor mode was launched which had automatically save the level data!

4.16. Important information about object collisions during level creating

Whilst creating a level, always keep in mind that I assume a level creator to possess some kind of intelligence. The editor doesn't contain any sophisticated controlling mechanisms for detecting of overlapping and collisions of placed or moved objects and will have no problem with placing a figure to a wall, a stone, illogical overlapping of active objects etc. Do not try to test out or use this "weakness". Also, do not write to me mails claiming that when you placed two stones the way they overlapped, the game engine behaved unusually. As it's been said, the editor and game engine assume some eligibility and intelligence of a user, and it's supposed that a level will be created sensibly without any collisions between objects!

So, always create a level the way that no objects are overlapped the way that cannot happen whilst playing a level! Otherwise, it might occur that objects behave illogically, or an attempt to launch of such a mistakenly designed level can cause the game to crash!

Some kinds of overlapping, though, during advanced creation and with sufficient knowledge of the creator are possible. A classic example might be a moving wall which should block the lift and is placed the way that during a level start there is a top or bottom side of a lift. In this case, however, the moving wall must **always be set to be slid** (hidden)! The same rules apply to the situation moving wall versus stone etc. So, if you happen to use overlapping of active objects, **you always have to know what and why you are doing it!**

5. Level textures editor

5.1. Introduction to the level texture editor

This is the third and last part of the entire complex QX EFF. However, I would like to warn all potential creators hoping they'll be able to create the same levels they saw in Quadrax X. The texture editor for QX EFF is only a VERY SIMPLE bitmap editor allowing basic texturing of the level and mid-ground. It's needed to realize it was not my intention to create an editor with the same abilities and options as the advanced bitmap editors (such as GIMP or Photoshop), apart from the fact that it wouldn't be possible anyway, because it would take some (decades of) years for one person to create something as similar as Photoshop.

All the levels you saw in Quadrax X had been prepared (or their basics) in this very simple level texture editor, but a few tens of hours of work in "real" editors followed until they were revamped to the look you saw in QX.

In the chapter [Advanced options of editing the level layers in external graphics programs](#) it's thoroughly described how to work with the layers creating the resultant picture of the level, and what tricks you should use to reach approximately the same look as in QX.

5.2. The difference between a level in the level editor and its real graphics look in the game


First, you need to realize that everything you've created in the level editor is just data defining how the level and objects in it will behave. What the level will look with regard to the graphics side depends solely upon how you texture the level! You can try it out by launching a ready but non-textured level in the regular gaming mode and you'll see (or won't see ;-)) that apart from the episode background and active objects, nothing else is visible in the level! Therefore, it's needed to create a final graphics look. And this texture editor is intended right for basic texturing.



The game engine doesn't render any of static objects of the level either. This basically means that ladders and ropes ARE NOT visible in the graphics look of the level! See more in the chapter [Ladders and ropes](#).

5.3. Editor and level texturing controls

After entering the editor, you'll see the special interpretation of all static and active level objects shown above the background (layer 0) of the respective episode. The image of the episode background is dependent upon the position of the level placement in the main level editor. For the levels 1-10, it's the episode 1, for 11-20 episode 2 etc. Instead of a classic mouse cursor you'll see a

square object for erasing the elements of the foreground , that moves exactly in the raster grid of the level.

The editor has four main modes:

- Erasing the foreground (layer 2) – the basic mode (default when the editor is entered)
- Drawing the foreground (layer 2)
- Choosing graphics elements for drawing the foreground
- Drawing/erasing the mid-ground - layer 1

In order you to understand what is drawn where, it's needed to first expound what layers the resultant image consists of.





5.3.1. Layers of the image

For better understanding how drawing of the textures work, it's good to imagine that the level image consists of three different layers placed one upon another. The bottommost layer – backdrop – is the picture of the background of each episode. Let's call it the layer 0. Upon it there are two more layers placed that you can imagine as clear glass panes on which you can independently draw on. (Imagine a picture, e.g. a countryside by a painter (layer 0) glassed with two glass panes – and you can draw on these). The layer right above the background is the layer 1 and the topmost layer placed totally in front is the layer 2. Sometimes I prefer to call the layer 1 mid-ground, as it's placed between the background and top layer. If you've correctly imagined layers 1 and 2 as glass panes, it will be clear for you what covers what when they're being drawn on. If you draw something to the layer 1 (mid-ground – the mid layer that is placed right above the bottom image of the episode), what you draw will cover the original bottommost picture. And if you draw to the layer 2 (the one at the top), everything you draw on it will cover the bottommost layer 0 and the mid layer 1 (the one BELOW the layer 2).

Ugh, I hope everyone understands! Further in the description, I'll refer to the layers strictly with their numbers 1 and 2 which are the only ones in the editor that can be “drawn on”.


5.3.2. Showing and hiding of individual elements

For more precise and more comfortable drawing the editor allows turning on and off the visualizations of all the elements on the picture. This is controlled with number keys, where:


-  - Hides/shows the layer 0 (when hidden, the layer 0 is replaced with a black area)
-  - Hides/shows the layer 1
-  - Hides/shows the static objects of the level apart from trap-cellars
-  - Hides/shows active objects in the level + trap-cellars

 **5** - Hides/shows the layer 2



 **6** - Hides/shows the contours of graphic elements inserted to the layer 2

 **7** - Hides all the layers apart from the layer 1 and turns on the contours of elements of the layer 2

(this is a special mode for drawing to the layer 1 – the explanation can be found in the chapter [Drawing/erasing of the mid-ground \(layer 1\)](#)).




 **0** - Resets all the settings to the default look (i.e. everything is visible except for contours of objects of the layer 2)



All static and active objects of a level which are shown and hidden using keys  **3** and  **4**, are placed on a virtual position BETWEEN the layers 1 and 2. So, if you draw to the layer 1, things you draw are shown below them, whereas things you draw to the layer 2 overlaps them. **This order of showing active objects only applies to the texture editor**, when playing in a level, active objects are placed before the layer 1 and the layer 2! See more in the chapter [The principle of image arrangement in the game](#).

5.3.3. Drawing the foreground (layer 2)

First, it's good to start drawing the foreground, i.e. the layer 2. Here all the textures of walls, moving walls and trap-cellars are placed.

Pressing the key Tabulator () will switch the editor to the mode of graphic elements selection for drawing the layer 2, and the mouse cursor will change to the classic one. Using arrow keys  and  on the keyboard you can browse between different kinds of graphics elements for drawing the foreground. Those who finished the entire Quadrax X will know all graphic elements for sure. After choosing a respective list of graphics elements you can choose a desired element by clicking with the left mouse button on it, and the editor will automatically switch to the mode of drawing to the layer 2. If you currently don't want to choose any object, press the key Escape or Tabulator or the right mouse button and the editor will revert back to the basic mode of erasing the layer 2.

If you have chosen a graphics element in the previous step, you can now move with it on the screen using mouse. Pressing the left mouse button allows you to repeatedly draw a respective element to the layer 2.



The editor guards the respective graphics elements not to overlap, and won't allow placing any graphics element the way they would collide with another one.



Whilst drawing walls to the level, it is beneficial to turn off showing of the active objects (4) and therefore see only the walls of the level where you're drawing their graphic look. On the contrary, for drawing moving walls and trap-cellars it is convenient to turn off showing of the walls (3) and keep the showing of the active elements.

If you've already drawn (placed to the layer 2) as many appearances of a specific graphic element as you need, you can return to the mode of selecting graphics elements (it will be opened automatically on the last chosen list) by pressing the key **Tab** and repeat it all until you finish the entire layer 2.

If you press the right mouse button or the button Escape in the mode of placing graphics elements to the layer 2, the editor will revert back to the basic mode.

If the editor is in the basic mode – thus if instead of a cursor a square with the red sign DEL is showed, **you can erase all graphics elements that you have drawn to the layer 2 in previous steps.** All it takes is move the cursor (square) WHEREVER on any graphic element and by pressing the left mouse button this element will be erased from the layer 2.



When drawing to the layer 2, always keep in mind it's only a graphic look of the level and has nothing to do how the figures and active objects can move in the level! If you draw somewhere to the layer 2 a wall that is not defined with a static element of wall in the level editor, it won't be an obstacle for any object! On the contrary, if you don't draw a wall anywhere in the level, but it's been defined in the level editor, despite the fact it will seem in the gaming mode that there's nothing, the figures and active object won't go through! **Complete carefulness and surgical precision whilst drawing the layer 2 is therefore a key condition in order to create a good and flawless level.** When drawing to the layer 2, always check from time to time using the key **5** that you've drawn the walls only there where they should be, and on the contrary that you haven't drawn a wall somewhere it shouldn't be!

If you have drawn the entire layer 2, it's convenient before drawing to the layer 1 to test it out in the gaming mode to ensure the layer 2 is fine and that you haven't forgotten to draw a wall anywhere, a moving wall, trap-cellar etc.

Now you may be asking where some graphics elements you saw as walls in the episodes 2, 5, 6, or 9 are. The answer will probably let you down. Nowhere. Completely all walls in the levels of episode 2, 5, 6 and 9 were drawn "manually" directly in the advanced graphic editor such as GIMP and Photoshop. (At episode 5 and 6 I mean the texture of brick walls.) Nonetheless you can still create in this QX EFF editor a masking basic for definitions of walls for the following work in external editors. If the editor is in the mode of menu of graphic elements, press the key **G** and the active list will switch to the showing of a grey mask of graphic elements. These masks can then be placed the very same way as the rest of graphic objects until the moment when you've created in the layer 2 the entire grey mask for current walls.

This is exactly how I did the drawing of all levels of episodes 2, 5, 6 and 9 in Quadrax X!

Further information about how to work with such a mask afterwards can be found in the chapter [Using the wall masks for creating custom wall textures](#).




If you want to change the current graphic element for another one of the same size, you needn't erase the previous element from the layer 2. Simply choose another element of the same width and height and simply redraw with it the original one.



Graphic objects that don't have the exact rectangular or square shape (skewed or rounded off) they still take up the square and rectangular size corresponding with their longest edges! This needs to be thought of when placing elements to a level! It can be pronouncedly seen, if you turn on by pressing the key 6 the showing of contours of graphics elements – the contour shows how much space each elements really takes up.


If you are finally gratified with the main layer of the foreground, we can get to drawing the layer 1.

5.3.4. Drawing/erasing of the mid-ground (layer 1)

If the editor is in the basic mode, pressing the key  will switch it to the mode of drawing to the layer 1. The mouse cursor will change to the “aiming cross” with the symbol + in top right-hand quadrant and with the number 1 in the bottom right-hand quadrant. Symbol + indicates that **drawing** to the layer 1 is active and the number 1 in the bottom right-hand side sets that the texture number 1 will be drawn on.

Press and hold the left mouse button. Subsequently by moving the mouse you'll see a rectangle being created that shows where the texture will be drawn. After releasing the left mouse button this texture will be rendered whose type is defined with the number at the mouse button to the layer 1. Simple, isn't it? ;-)






When drawing to the layer 1, it is convenient to set up a schematic mode of showing using the key . Thus, no other elements apart from the layer 1 and the contours or graphics elements of the layer 2 will be shown. So you can draw to the layer 1 the way you need.

If you want to erase what has been drawn in the layer 1, you can do it the same way as drawing but with the right mouse button. The symbol in the top right-hand quadrant will change to – indicating that the mode of **erasing** is active. By holding the right mouse button and moving the mouse you can highlight the area you wish to erase. After releasing the right mouse button, the entire area will be erased.




For those who have grown used from other graphics programs to using for erasing the combination of the keys ALT and the left mouse button this option is here as well! When you press and hold ALT, the symbol + at the cursor will change to – and you can erase in the layer 1 with the left mouse button. The key ALT needs to be held only until pressing the left mouse button, then you can release it, and as far as you hold the left mouse button, the erasing mode will be active.

In the mode of drawing to the layer 1 all functions of hiding/showing other objects are active apart from the key . When drawing the mid-ground, the layer 1 cannot be hidden.

When the mode of drawing to the layer 1 is active, you can, using arrow keys  and  on the keyboard, cyclically change the texture type with which you'll be drawing. The number in the bottom right-hand corner of the mouse cursor always indicates what texture number (type) will be used.





Do not combine more textures to the layer 1, unless it is a level that is supposed to symbolize the move to another episode (e.g. the level 80 in QX) and if you have thought it over! Cramming a level with more types of textures of the mid-ground is from the standpoint of graphic design counterproductive.

Return from the mode of drawing to the layer 1 is done by pressing the key  or the key Escape. The editor will therefore return to its basic mode (erasing the layer 2).

5.3.5. Ladders and ropes

Since the game engine doesn't render in the gaming mode any static object, ladders and ropes wouldn't be visible at all. So they need to be drawn either to the layer 1 or layer 2. From the standpoint of rendering in the game itself it doesn't matter, just like it's explained in the chapter [The principle of image arrangement in the game](#), both the layer 1 and 2 will both merge up to one backdrop. However, if you've constructed the level the way that a moving wall crosses (overlaps) one of these two static objects, it's inevitable to draw them to the layer 1! (As picture data for moving walls is taken from the layer 2.)

The texture editor has a special function allowing a disposable rendering of ladders and ropes to the chosen layer. After pressing the key  the dialog box will show up setting to which layer ladders and ropes should be rendered. Choose the button with the layer label or the button cancel, if you don't wish to carry out any action.


If you choose one of the layers, the rendering of basic graphics look of ladders and ropes will carry out. If you have the showing of static objects (the key ), turned on, **at first glance nothing will change!** But turning off this image will show you that the ladders and ropes are now drawn to the same layer you've chosen in the dialog box (you can check this by hiding the respective layer).



If you want ladders and ropes to have a look differing from the default one, you have to draw them yourself. See [Ladders and ropes for experts](#).



Using the function of rendering ladders and ropes to some of the layer will **block** the ability to work with these objects in any way in the level editor! It's in order to prevent conflict between the level data and its graphic look. Also functions of level moving and mirroring will be blocked.

If you have ladders and ropes rendered in a layer and decide to erase them from it, press the key . A dialog box will show up asking you whether you really wish to erase these objects and from which layer they should be. After clicking OK an erasure of ladder and rope objects from the respective layer will carry out.



Erasing ladders and ropes will completely erase all picture data on all logical fields (i.e. always 30x30 pixels) from the layer where ropes and ladders occur. So carefully consider before you start rendering ropes and ladders whether you have finished work in all other layers!



By erasing ladders and ropes from the layer they were on, an option to work with these objects in the level editor **will be enabled** again.



If there's no moving wall covering ladders and ropes in the level, I recommend beginners to always draw to the layer 2. The risk you will erase some picture data is a lot lower, because the layer 2 contains mainly pictures of graphics objects of solid walls, and these cannot collide with the ladders or ropes.

If you mean to edit the layers of a level in an external graphic program, it is better not to use this function of QX EFF at all, but carry out the drawing of ladders and ropes "manually" – see the chapter [Ladders and ropes for experts](#).

And that's it. Really! Nothing more can be done in this simple texture editor. So you may be asking the question that is the name of the following chapter:

5.3.6. Where are the shadows and lighting from the sun, flames and torches? Where are the statues and other decorations?

Only in your hands and your skills of working with advanced bitmap editors such as GIMP or Photoshop! All other lighting or other effect or changes of layer 1 and 2 need to be done externally with post processing in a sophisticated bitmap graphic editor.

If you have no experience with working with such editors, but would like your levels to look similarly to the ones in QX, I recommend you to learn to work with a great bitmap editor GIMP that is very

close to Photoshop with regard to the options of editing and altering images and in some cases (for example, precise work with the values in alpha channel) it outdoes it! Its pros are unequivocal – it's absolutely free. GIMP can be downloaded directly from the website of its developers:

<http://www.gimp.org/>

More skilled graphic designers are certain to have their favorite editor already, so I leave that choice up to them. ;-)

How to work with the respective layers in an external graphic programs is mentioned in the chapter [Advanced options of editing the level layers in external graphics programs](#).

5.3.7. Finishing work in the texture editor

If the editor is in the basic mode, it's enough to press the key Escape or right mouse button. All changes made to the level **will be automatically saved** and the program will revert back to the level editor. If you don't wish for some reason to save the changes (for instance, you've unwittingly erased something in an earlier saved level), press the combination of keys SHIFT + Escape or SHIFT + the right mouse button. The program will return back to the level editor **without changing** the textures.

5.4. Advanced options of editing the level layers in external graphics programs

In the subsequent chapters you'll find a thorough description how to work with respective physical files of level layers. The following text is intended for those who want to master the looks of the graphic layers to the perfection. Unfortunately, in this text I won't shun using more technical terms, still I'll try to find the best way so that a diligent layman would understand it, and an expert wouldn't be outraged. ;-) I will also assume some kind of PC literacy so I'm not going to explain the terms such as folder, file path, etc. If I use in this text a collocation "graphic programs" I will always mean the complex project of one level file for GIMP or Photoshop.

5.4.1. Placing files of the level layers

Open the folder **projects** in the root folder where you have installed QX EFF. You'll see in it the folders of respective projects (0000 , 0001 etc.). If you don't know which of these project is currently active, you'll find it in Project manager. Open the folder of the respective project and go further to the folder named **levels**. There you will see all the files of all layers of each level, indexed from 00. (I.e. the level 1 uses the **00_layer1.png**, **00_layer2.png**, **00_layer3.png**... etc. until the **99_layer3.png**)

Don't pay attention to the file layer3 right now, it will be explained later. If you open the files layer1 or layer2 in your graphic program, you'll see in them exactly what you've drawn to them in the texture editor in QX EFF. In order to prevent any unwanted damage, make a backup of these files, and then you can start working.

5.4.2. The principle of image arrangement in the game

You can skip this chapter so far if you don't gather everything.

The principle of image arrangement in the game was described in detail in the chapter [Layers of the image](#). When playing the level, the picture itself is arranged slightly differently. Before the level start this preparatory work will be done:

- Layer1 will be moved first to the bitmap of the episode background using alpha blending.
- Then, again using alphablending, layer2 will be moved, APART FROM the textures of trap-cellars and moving walls that are saved aside to the separate bitmap that afterwards the game engine uses for animating of moving walls and trap-cellars.

The resultant bitmap (i.e. the episode background + layer1 + layer2) will be saved to the memory also to the hard drive (where it's used as an immediate source for loading in case of starting/restarting the entire level). During the game itself, this bitmap is the first (or the second if scudding skies are used as a background) data source saved to the backbuffer of video memory. After that, more elements of the level are shown such as animations, active objects etc. in such an order that as a result they will together create the overall image of one level frame.

5.4.3. Data format of PNG files and their dimensions

Basically, it's not required to fully comprehend how the game works with data! What, however, is unequivocally important is that graphic files of the type PNG used by the engine of QX EFF as sources from the game **ALWAYS have to contain the Alpha channel**, and it also **MUST BE a 32bit PNG**! The graphic program with which you'll be editing the images must be able to make and fully support creating of 32bit PNG files with Alpha channel and work with them! Programs I've mentioned several times GIMP and Photoshop can, certainly, do this.



If you save PNG files for usage in the game, **ALWAYS set when saving a PNG file in your graphic program the maximal possible compression**! If you export a project using Project manager, it automatically assumes that all source PNG images in the project are already zipped and unlike the other files, it doesn't zip them! See more in [Further information about file exporting](#).



The dimensions of PNG images of respective layer MUST BE EXACTLY 1920x1080 pixels! If you save an image of the level layer in another dimension, an error will occur when loading the data of level layer and their arranging during the operation with memory, and the game will crash!

5.4.4. An example of a graphics project for the level one in QX

It's said that one example is better than 1,000 words. So I'll try to save a thousand words. ;-)

In the root folder of the game go to **docs\examples** where you'll find the pictures of all three layers for the first level of Quadrax. What's more important, though, is you can find there the file **sources.zip**, containing the files for GIMP (version 2.8.16) and Photoshop (version CS6). Unpack the file **sources.zip** then you can either in GIMP or in Photoshop open the respective file **00_layers.***

As you can see, the source graphics project for the level 1 in QX contains all needed sources for all three layers of the level 1. For each level of your project I recommend creating such a separate graphics project, load the layers 1 and 2 into it to separate groups and then work with them mutually. Always, when you afterwards want to create/update a layer of the file for the game, turn off the visibility of other layers (groups) in your graphic project and export the project as a correctly indexed PNG file to the folder **levels** of your current project. (If you, for instance, work on the level 15 and the layer 2, export the image as **14_layer2.png**).



Keep in mind that first two digits under the underscore always represent the level INDEX and not its number! Level index is always smaller by 1 than the level number. Also don't forget that indexes smaller than 10 must be 2-digit. The file name **4_layer1.png** is wrong, the correct version is: **04_layer1.png**!

The same thing that was said at the beginning of this manual in the chapter [Start slowly](#) applies to creating graphics too. Start with simple levels and simple graphics. Once you get used to working with respective layers, you can start creating more and more complex, and more and more perfect graphics creations.

Since each level is created with a separate image, it depends only on your skills of working with graphic what the resultant level will look like. There are virtually no restrictions! I myself am not fully gratified with the graphics of entire Quadrax X, as I am not a professional graphic or painter. I am aware that if a skilful person capable of drawing had collaborated, each level could have been a separate work of art!

5.4.5. Tips and tricks for advanced layer editing

In this chapter I automatically assume that you have defined a source file for each level (graphic project of the level) where you have saved two original layers (layer1 and layer2) in separate sectors/folders. I also assume you have some skills and knowledge with your graphic software, so I'm not going to describe all graphic operations in detail.

5.4.5.1. Using the wall masks for creating custom wall textures

In the chapter [Drawing the foreground \(layer 2\)](#) there's at its end a way for creating layer2 the way that instead of textures of graphic objects, it would contain a grey mask on the places where walls

are planned. Use this simply as a basic mask for your full-screen textures which should represent “walls” in the level. After applying the masks (it needs to be altered depending on what graphic program you use) you’ll obtain the basic layer of walls represented with your texture. Then it’s just painstaking work with debugging this layer to the perfection. If you have understood how layer2 works in the game, it will definitely be no problem for you to create similar textures of walls just like in Quadrax X in the episodes 2, 5, 6 and 9.

5.4.5.2. Shading

In order to increase the level plasticity, it’s convenient to create the shades casted by the walls in layer2 in the mid-ground in layer1. First, you need to extract all textures of moving walls and trap-cellars from layer2 to the separate layer. Once you have the layer containing ONLY wall textures (without textures of moving walls and trap-cellars), duplicate it and use the respective shade effect for the new copy. Set the filling of the layer to 0% (so that only the shade will remain) and apply the effect to the layer. Move this layer above the layer1 and apply to it the mask layer1 (e.g. in Photoshop simply by choosing Creating Clipping Mask). Similarly, using the next layer, you can create darkening of dark corridors etc. You’ll see this all very well in the example of the previous chapter.



It’s vital to realize that all textures for moving walls are taken from the layer 2 during the level preparation by the game engine, and always in the full extent of the defined size of a moving wall and in the state of its slide! Here it doesn’t matter, what thickness or slide limit the moving wall has defined! If there should be any static image UNDER the moving wall, it’s needed to place it to the layer1! Because of this, it’s also needed to create the contingent shades from the layer which DOESN’T contain moving walls, as these shades would be shown also at slid moving walls what might look very strange at least!

5.4.5.3. Ancillary layers in the graphics project

It’s good to add to your graphic program an ancillary layer showing all active level objects (you can see them clearly in the example from the chapter [An example of a graphics project for the level one in QX](#)) in order to have a good overview at what and where exactly is in your level. You can create this layer easily using the following steps (I enclose the procedure I used when creating QX):

- In the texture editor of QX EFF turn off absolutely everything apart from active objects (only trap-cellars will be visible from static objects – exactly for this usage)
- Save the screenshot of the game (in the graphic settings in the game it’s needed a screenshot type to be BMP!)
- Leave the texture editor of QX EFF using Shift + Escape
- Insert that screenshot as a separate layer in your graphics project of the level
- In this layer remove all black pixels (those with RGB values 0,0,0)

And that’s it, the ancillary layer of active objects and trap-cellars is ready. This way you can insert any ancillary layer – I personally also used the current background of the respective episode with the sky so that I could assess the overall look of the level, what it would look like in the gaming mode.



Don't forget when saving the layer 1 or 2 as PNG files for the game to turn off the showing of other groups and all ancillary layers!

5.4.5.4. Lighting from the sun, torches, flames etc.

In the folder **docs\sources** I've enclosed the basic version of textures of the mid-ground (layer1). You can use these textures as another layer in your graphic program and create the effects of lighting of the mid-layer (layer1) from torches, sun, etc. with respective effects and masks. These textures correspond with their drawings exactly with the textures used by the editor QX EFF as sources of [drawing the mid-ground](#), but are not colored. With appropriate color filters you can therefore easily create a set of different layers which afterwards apply to the mid-ground in your graphic project with their respective masks.



In the demonstrative project from the chapter [An example of the graphical project for the level one in QX](#) there are all effect layers already finalized, i.e. all needed light effects have been applied to them. Nevertheless, study them thoroughly so that arranging of these layers to the final layer1 and layer2 will be more obvious to you.

5.4.5.5. Ladders and ropes for experts

In the chapter [Ladders and ropes](#) it was described how to render ladders and ropes to layer1 or layer2. However, I myself never used this when creating Quadrax X! If you really want your level to be truly graphically unified and immaculate it's better not to use the built-in function of the editor QX EFF for rendering these elements, but rather draw them yourself to the separate layer in your graphic program. It's good to have this layer in the group layer1 – then you can simply create at ladders a shade casting to the mid-ground etc. Certainly, you can draw ladders and ropes the way they will fit with the environment of the respective episode.

5.4.5.6. Trees, statues, greenery and other graphics objects

After a little consideration I decided that QX EFF would not contain any graphic sources I used when creating Quadrax X. First of all, it's because searching them on the Internet, modifying and editing, or perhaps completely entire creation took me hundreds and hundreds of hours, and second of all, because many of these objects I intend to use in my subsequent projects. But for those who fall to doing complete alterations of the graphics look of the levels, it definitely won't be a problem to create their own collection of graphics elements which they'll be using in their levels. After all, you don't want to create a copy of levels of Quadrax X but particularly your own levels, do you? ;-)

5.4.6. Mysterious layer 3

If you've opened and thoroughly studied [An example of the graphical project for the level one in QX](#), the meaning of layer3 is obvious to you. It's simply a layer working as a mask of shades for covering the figures (dynamic lighting) in the level. This defines how much "darkened" the figures will be in a specific part of the level. The rules apply that if Alpha is 0, it has null effect on the figure. With the

number value of Alpha increasing, the figures get darker and darker (until they turn into black silhouette at Alpha value 255).

You don't have to be very "accurate" when creating this layer, as it has effect only on the figures in the level. When you study this layer of the project from the example, everything will clear up for you.



Do not use in layer3 a value of Alpha bigger than 39% (value 100 from the scale 0 - 255). At this level of Alpha, figures are dark enough even for the darkest nooks of the level.

5.4.7. The caveat for the end

When you once start editing the respective files layer1/2 in an external graphic program, then don't use for that level the texture editor in QX EFF (apart from creating ancillary layers to your external graphic program). Although, it would have no trouble loading the modified files, it would be more harmful than useful. For instance, erasing a graphics element will erase the complete rectangular area of data defined with the respective element in layer 2 (all pixels in that area of layer2 will be replaced by the ARBG value 0,0,0,0), etc.

6. Advanced options of game modding

Or your own Quadrax easily and fast!

The engine of QX EFF allows you to use in your project custom background, music and background image of statistics of each episode! And it's simply by adding respective files to your project. An entire episode (and basically the whole game) can look **utterly different** to Quadrax X!

This is why you can set the detailed behavior of the environment of each episode in Project manager. I.e., if and what sky texture should be used in a respective episode, what additional animations should be used, what effects and much more! The description of all options of the episode environment can be found in the chapter [Advanced](#) of Project manager.

And of course, you can write custom texts of clue scrolls and diary chapters.

6.1. Custom episode background

In QX EFF images of level backgrounds are the same in each episode, just like in Quadrax X. However, if you want to use as a level background for some of the episodes a custom image, it is no problem! Just create your own picture you wish to be a background of the respective episode and then place it to the folder **backgrounds** in your project.



Picture must be in a PNG format and must have dimensions EXACTLY 1920x1080 pixels!

The filename must be written as **bkgXX.png**, where **XX** is a 2-digit number of the episode for which you wish to use the image. For example, for the episode 4 the file must be named **bkg04.png**.

The game engine, before the level starts, searches the folder **backgrounds** and if it doesn't find a file named by the criteria above, it will use the default image of the episode from QX.

6.2. Custom level statistics background

The picture of level statistics is the one where final statistics are shown once the level has been successfully finished. Also this one can be changed in your project for any episode!

Just like the image of episode background, all it takes is place it to the folder background in your project. The filename must be **bksXX.png**, where **XX** is a 2-digit episode number for which you wish to use the image. It's exactly the same process as at the image of episode background.



Picture must be in a PNG format and must have dimensions EXACTLY 1920x1080 pixels!



It's appropriate to comply with the same theme of the episode – i.e. if you want to change the background image of episode levels, you should also change the image of statistics – and they should be thematically equal.

6.3. Custom episode music

And lastly the music. Even the default music from Quadrax X playing whilst playing levels can be changed by you with your own!

Just insert a file **qXX.mp3** (or **qXX.ogg**) to the folder **music** where, just like at pictures of backgrounds and statistics, **XX** is the number of the episode for which you want to use the music. So, music for the first episode must be named **q01.mp3** or **q01.ogg** etc.



The engine of QX EFF is capable of working with two formats – MP3 and OGG. So you can choose the one you prefer.



If you have more than a few simple levels in the episode, it's good the music to have the corresponding length so that it soon won't become repetitive and pesky. For full episodes (i.e. all 10 levels), the minimal recommended length is 15 minutes, but more is always a plus. If you want the music to make playing more comfortable and not vice-versa, it's also vital to choose the convenient genre or style corresponding with the episode environment. Examples of suitable songs are ambient ones etc. Definitely avoid using vivid and dynamic music or songs with singing! Music should enrich the episode but not disturb it!



You have to be aware that during the project export all this data is exported too! So if you intend to add custom songs for the episode to the project, it's suitable to choose songs in the bitrate 128kbit/s, maximally 192kbit/s. Even so, such a song can have tens of MB and the resultant size of the exported project will increase exactly by the overall size of all music files!

6.4. Custom texts of clue scrolls and diary

6.4.1. Texts of clue scrolls

In the folder texts of your project you can find these files:

scrolls_cz.txt

scrolls_en.txt

They contain simple unformatted text of all clue scrolls for all hundred levels of the project. Like their names imply, the first file is intended for Czech texts and the second one for English. I don't assume that a creator will immediately start creating the bilingual variety of his or her project, so during the editing choose the file in which language your project will be.

The file of clue scrolls has rather strict rules about how and what you can write into it. These rules are listed as notes in the file itself at its beginning, nevertheless here you can find them in an expanded version:

- **The maximal length of a line in the clue is 50 characters!**
- **The maximal amount of rows in one clue is 8!**
- The character TABULATOR is **forbidden**
- Encoding of Czech is CP1250
- Each hint **must** be indexed with a hash symbol (#) and a 2-digit index (00 - 99) after which the name of the level follows in quotation marks and on next rows there's a contingent text of the clue (max. 8 rows!)
- Clues must be sorted gradually 00-99!
- The very last row of the entire file **must** contain only sequence **#EOF**

If you open the file of the clues, you'll definitely understand on your own how it's "pre-filled".



Do not use programs such as Microsoft Word or similar for editing the hint texts! Use an editor which is able to work with plain TEXT files! I personally recommend an immaculate editor PSpad, <http://www.pspad.com/>, which I myself have been using for many years. But any plain text editor will do, even the Notepad in Windows.



Be really wary to comply with the rules above! If you infringe them, it could easily occur that during an attempt to show a clue scroll not abiding by the strict rules could make the game crash!

6.4.2. The text of the diary

The files of the diary text are in the same folder **texts** as the clues:

story_cz.txt
story_en.txt

They also have rather strict rules, so here they are:

- **The maximal length of a line in the diary is 50 characters!**
- **Each chapter must begin with the row with the operative character hash (#)**
- The character TABULATOR is **forbidden**
- Encoding of Czech is CP1250
- The very last row of the entire file **must** begin with the two hash symbols in a row (##)

Under each line with the character hash (#) the text of a custom chapter starts. It can have a discretionary amount of rows.



The engine of QX EFF cannot add line/word breaks automatically. So when you type to the diary text, you must from time to time to check whether some rows “overflow” behind the edge of diary lists. This applies mainly for the rows where wide letters occur – mainly uppercase ones, letter such as w, m etc.

The amount of rows one page of the diary is 20. This is important to know to prevent the situation happening that for instance, there will be only the last line of the paragraph on one page. In this case it’s better to increase the line spacing by 1 (add empty line) so that the next page there would be two lines from the paragraph. But you’ll find this all out yourselves during playing and the following control directly in QX EFF.



Be really wary to comply with the rules above! If you infringe them, it could easily occur that during an attempt to show a helpful hint not abiding by the strict rules could make the game crash!



Since tabulators in the diary are forbidden, if you need to add an indent to the first line of the paragraph, use four spaces.

A note for the end: Originally I intended to implement to QX EFF an option to insert to the diary custom pictures, but in the end I rejected it. The options of modding of the entire QX EFF are already remarkable. In addition, inserting images to the diary is again highly bonded with the inner rules of the game engine and it’s so intricate that apart from me no one would actually comprehend them, without mentioning that the entire program could crash very easily and undesirably.

6.5. Custom graphics of active elements



This chapter is intended only for utter experts in graphics! If you don’t know what a 32bit PNG format, or alpha channel means, do not out try out anything from this chapter!

QX EFF allows you to completely replace graphics of virtually all active elements of the game! The instructions are intended for experts only and I’m not going to expound here any technical details. **It’s made only for user who know exactly what they do!**

- Copy files **anim.qxg** and **anim_g.qxg** from the folder **data\graphic** to the folder **backgrounds** in your project.
- Whilst copying (or after it) change the suffix of both these files to **png**.
- Change the graphics of the elements you want.

- Keep in mind that this is the key source file of the game not only for the graphics of active elements but also for other game graphics! So be very wary whilst modifying these PNG files and change only these objects you intend to use for your own project! The rest of the parts must retain the way they are!

6.6. Lastly about modding of QX EFF

If you have carefully read the previous chapters, it's become obvious to you that you can create **your own and unique Quadrax** with even a hundred of full-featured levels where from the original QX EFF only the screens of menu, settings and a few other things will retain, but the rest of it will look the way you create it. Although, I don't really believe that someone will fall to doing such a huge project, but perhaps I'm just underestimating you. The time will show. Anyway, QX EFF allows this and if somebody wants to use the full scale of its features, they will have no obstacles to do so.

7. Project manager

QX EFF Project manager is a separate app which, as the name “implies” is used for complex project administration for QX EFF. In the following chapters the entire work with the app and the functionality of each menu item will be expounded in detail.

The application can be found in the folder of the game and its link is also placed, if you choose it during the installation, on the desktop.



In the following text QX EFF Project manager is talked about just as “Project manager”, whereas by the labelling QX EFF I mean the game itself and level editor.

7.1. What a “project” is

A project for QX EFF is one complete collection of data which contains within itself everything needed to work in QX EFF. Thus, it contains all files, folder and subfolders required for launching the project in QX EFF. Even if someone is not familiar with the file systems (and doesn’t intend to ever work with the modding of the game), they needn’t feel any worries. Project manager will allow them to work with respective projects completely comfortably without them having to have any professional skills with working on PC.

All it takes is read this guidebook, and then you’ll see that project administration for QX EFF is simple and trivial. So we’re not going to be lingering here and get straight to the description of the entire application.

7.2. After launching

After launching the application Project manager, a standard window of Windows app will pop up in whose upper part you’ll find the menu using which the entire Project manager is controlled and in the area of the window you’ll see the static information about an active project. Once installed, Project manager (and entire QX EFF) contains a default project with the id 0000.

ID is a unique 4-digit project identifier – a number that is unique for each project. It can gain values from 0000 to 9999. Hence, in QX EFF you can have up to ten thousand custom or imported project! Project ID is assigned **automatically** according to the lowest available number in the row of project IDs. With this ID an **equally named folder** is created in the folder **projects**, and in this folder with the same name as the project ID, you will find all other folders with all project files.



ID is a unique 4-digit project identifier – a number that is unique for each project. It can gain values from 0000 to 9999. Hence, in QX EFF you can have up to ten thousand custom or imported project! Project ID is assigned **automatically** according to the lowest available number in the row of project IDs. With this ID an **equally named folder** is created in the folder **projects**, and in this folder with the same name as the project ID, you will find all other folders with all project files.



Project manager can be launched only if QX EFF itself isn't running and vice versa. That's because both the apps work with the data of an active project, and therefore cannot be launched at the same time. Project manager and QX EFF guard this mutually and don't allow their launch if one of the apps is already launched.

In the following part we'll thoroughly scrutinize the meaning of respective rows shown in the app window.

7.2.1. Active project (ID)

An identifier of an active project, see above. In one moment you can set only ONE project to be active, i.e. the project which is currently being worked on in Project manager or QX EFF.

7.2.2. Project name

Your custom text how you will name your project. Its length can be up to 32 characters and that's particularly because the project name is shown on the page of best results in QX EFF (instead of a user password). Project 0000 is named "Default".



Choose an obvious and logical label for the project name. If possible, not general. Names such as "My project", "Project 1" etc. aren't very suitable, because if you exported a project and someone else imported it, the name "My project" wouldn't help very much to identify it, would it...

Fortunately, the project name can be changed at any time, even at an imported project. So if you import someone's project and the name doesn't reveal much about, you can change it easily e.g. to "project XY from the sir/madam XYZ". ;-)



Project name is a mandatory item. Project cannot be created without a name!

7.2.3. Project description

Your custom text providing more descriptive information about the project. Unlike the project name, it can be up to 1024 characters long. It's used for more detailed description of the entire project, and it's an optional item.

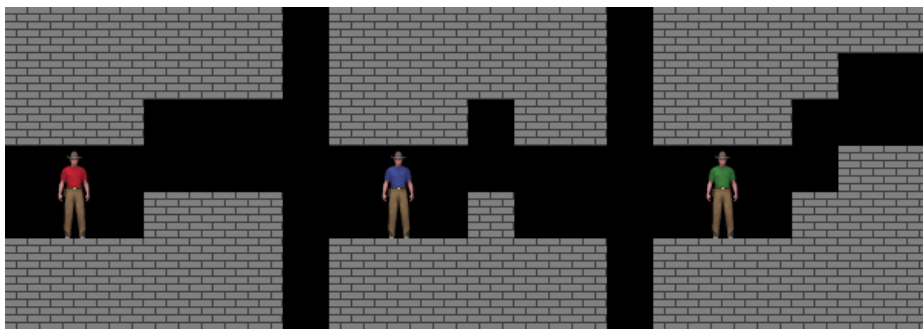


On the introductory window of Project manager only the shortened one-line-long insight is shown from the project description. You can view and edit the full project description in the basic project settings, see the chapter [Basic](#).

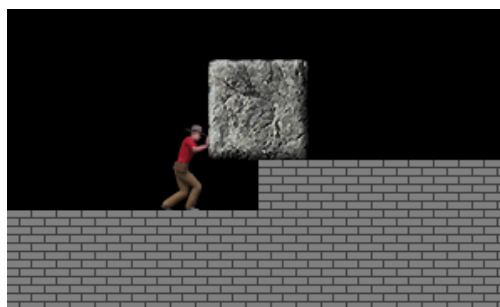
7.2.4. Allowed “Quadrax Neverending” moves

Here follows a sign YES/NO. Maybe it really is time to explain what that Quadrax Neverending is. Or rather what it will be. Well, it’s my very last project in the Quadrax series. There will be no QXI, QXII etc., but just QN (**Q**uadrax **N**everending). However, it will have one interesting and unique feature; there will be absolutely no limitations on the amount of episodes, levels, graphics, animations etc.! As the time will pass by, more and more levels will be added without any limitations. So, it truly will be **Neverending**.

So far, the project QN is in an early phase of development, even so I’ve already decided on two key changes compared to the entire present series of my Quadraxes. Two rules determining the movement of figures in level will change in a way to the better realness of their movement. The image will tell more than words:



In all other previous Quadraxes, none of the figures could pass to the right section of its part. But not in QN. Therein the figure will be able to get to the right – always if there is at least some gap and the place for a standing figure on the other side. And of course, they will also get back. I reckon it’s a lot more logical behavior than it was until now.



The second (and last) change is the ability of pushing all stones even if they’re not in the same position as the figure, but if they’re 1 field higher. Again see the picture above. In QX and lower the figure would be just powerlessly pushing the stone, but it wouldn’t move. In QX this behavior will be more logical again. If a stone can be pushed (i.e., it abides by all the conditions for a stone that can be

pushed), it will be possible in the situation in the picture above to move it to the right by 1 logical field. (Certainly, not any further, as the “stair” is hindering on the floor.)

And that’s all. It’s only these two changes which differ figure movements in QN from all previous Quadraxes, and even though it may not seem so, they’re rather crucial and allowing a much bigger flexibility of figure movements and their behavior in a level. And since a level that could be without this new element of behavior unfinishable (and vice versa), it’s always needed when establishing a new project to strictly choose whether these new rules of figure movements should be applied in QN or not. So, you can create projects that are with regard to figure behavior compatible with older Quadraxes or choose the modern behavior of figure movements just like in QN.



Once you have in a project at least one playable level, the style of figure movement cannot be changed! Neither can it be at changed at an imported locked project.

Now this question has definitely appeared in everyone’s mind – “Will I also be able to add my project from QX EFF to QX?” Well, it won’t be that simple. First of all, QN will have entirely different data structure and working with it will differ too. Since it will have to be able to work with theoretically infinite aggregate of levels, all level data including animations and level graphics will be uploaded dynamically always for one specific level. Second of all, in QN I want to retain some “mark of quality”, and I definitely won’t upload episodes that don’t comply with specific (read “my”) criteria of quality – from the level design and the point contained within them through the graphics and musical site to the good story of the episode.

So there you have it. If you manage to create a complete episode that’s worth to be placed to QX, I can carry out data conversion from the format of QX EFF to the format of an QN episode, and place this one to Quadrax Neverending. But I warn that it really must be a good and with regard to each side perfected episode – otherwise it would be futile to list it into QN.

Further information about how to proceed if you wish your project to be listed in QN can be found in the chapter [Project publications from QX EFF in Quadrax Neverending](#).

7.2.5. Allowed full editing mode

It can have a state YES/NO. Each project can have two states in which it is:

- **Fully editable** – This is how each new project in Project manager is set by default. It means that all the options of editing are available in the project.
- **Locked** – This the project state created by an import of a locked project. No editing of the project is allowed apart from its name and description. You’ll find out more about locking project in the chapter [Lock exported project](#).

7.2.6. Number of playable levels in project

The number announcing how many playable levels are contained in the project. (See the chapter [Level playability check](#).) If there isn't at least one playable level in a project, it cannot be locked during the export. See more in the chapter [Lock exported project](#).

7.2.7. Date and time of creating and the last modification of the project

The last two rows contain data when the project was founded and when the last change was made into it using Project manager or QX EFF.

In the following part of this guidebook you'll find a complete description of the entire setting menu of Project manager and all its items.

7.3. Projects

This menu contains items for administration of all projects that you have created or imported in QX EFF.

7.3.1. Create new project

If this item is chosen, a dialog box will appear where you can insert parameters of the new project:

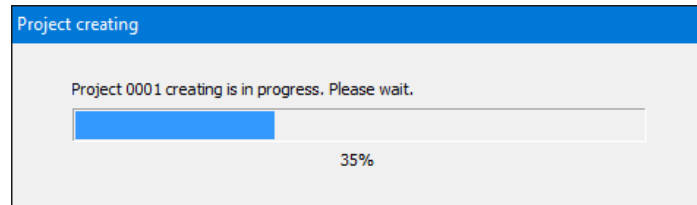
- Project ID – is chosen automatically as the first free ID within all existing projects
- Project name (explained in the chapter [Project name](#))
- Project description (explained in the chapter [Project description](#))
- Allow “Quadrax Neverending” moves (explained in the chapter [Allowed “Quadrax Neverending” moves](#)). This choice is checked by default.
- Set this project as active after creating – checking this option causes that after creating a new project it will be automatically set as active. This choice is checked by default.

After filling in the fields and choosing the options, a new clean project will be created after clicking the button If you've checked “After creating a project, set it as active”, then after creating such a project, Project manager will set it as an active project. Otherwise, the project will be created but an active project will still be the one that has been until then. (This applies to all dialog boxes containing the similar choice “...set it as active”.)

Clicking the button will close the dialog box without making or saving any changes. (The function of this button is completely same in all dialog boxes in Project manager, thus I'm not going to describe it in other descriptions of dialog boxes.)



Creating all data structures of a new project may take up some time. At that moment there's a window with graphic and text information about percentage progress of creating the new project shown on the screen – see below.



7.3.2. Select active project

After choosing this item from the menu, a dialog box will open where you can choose which of the projects you have in QX EFF will be active. By selecting from the project list and clicking the button the selected project will set as active.



At one moment you can set only **ONE** active project, i.e. the project with which both Project manager and QX EFF work. The project you set in Project manager as active will also load during the start of QX EFF and then you can work on it (create levels and everything else from the chapters 4 – 6). If you want to start working in QX EFF on another project, abort QX EFF, launch Project manager, choose other project as active, abort Project manager and launch QX EFF again. Project you've set in Project manager as active can now be edited in QX EFF.

7.3.3. Import project

With this choice you can import a project to your list of projects, which has been exported with the choice [Export project](#) from Project manager. If a project has been exported as locked (see more in [Lock exported project](#)), you won't have an option after its import to alter or edit it, apart from changing the project name and description.

After choosing a file for project import in a standard window for choosing files in Windows, a dialog will pop up where project file and choice will be listed whether you want after its import the project to be active. After clicking the button project import will initiate. Progress of the import is shown, similarly as its creation, in a window showing a percentage import status.

After a successful import, the project is added with a new ID to your project list, and if you've chosen during the project import the project to be set as active, it will be set (unexpectedly ;-)) as active.

7.3.4. Projects maintenance

This is a very special choice. A dialog box will pop up asking you to confirm whether you really want to launch project maintenance. This function searches the project folder and carries out the diagnostics of each found project. All void projects or projects with critical errors will be erased! **Use this function only if you have trouble creating/importing/copying projects!**



When working normally with Project manager you will probably never need to use this function. But if data damage of some of the projects occurred (PC crash, your own meddling with source data project files etc.), this function could be the last rescuer to repair the structure of project folders and all projects contained within it.



If you work on an extensive and important project, make its backup using function [Export project](#) regularly! Exported project files can then be archived on CD/DVD or USB flash memory and in case of critical PC crash or OS failure, you can easily restore them with a reverse import.

7.3.5. Exit

This option ends and closes Project manager.

7.4. Active project

In this menu there are choices listed for working with a currently active project.

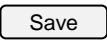
7.4.1. Setting

This choice contains two items of the submenu for setting the current active project.

7.4.1.1. Basic

Opens up a dialog box where you can change the name, description and setting of allowing figure movements just like QN. The project name and description can be changed any time at any project.

The status of the item Allow “Quadrax Neverending” moves is dependent upon the project state. If the project is locked and prevented from being edited or if it contains at least one playable level, the option is grayed and cannot be changed. (It’s a protection against creating project with checked playable levels that would have this value changed.)

Clicking the button  saves all the changes you’ve made in this window.

7.4.1.2. Advanced

This setting is determined mainly for those who decide on a complete overhaul of the game using custom episode pictures etc. (See more in the chapter [Advanced options of game modding](#)).

Opens up a dialog box where you can change attributes of environment for each episode. In new project are all settings for each episode set default as they used in Quadrax X.

The description of respective window elements:

Environment setting for – Selection of an episode from the select list for which you set the rest of the parameters in this window.

Use “night” environment textures – When this option is checked, the textures of all active elements and figures suitable for night environment will be used in the entire episode.

Horizontal moving texture – Choice of the texture of sky/lava that will be used in the whole episode.

At the end in the texture definition you can find two labels:

- Underlay – it means that **firstly** the texture of the sky is rendered, and **afterwards** the main backdrop episode picture follows. Thus, the episode picture lies above the texture of the sky and in order to make it visible, the top part of the episode picture must be transparent in these places (i.e., the pixels must have an Alpha channel set to 0). Certainly, the lowest transparent pixel mustn't be lower than the height of sky texture. The heights of respective underlay textures are listed at the end of this chapter.
- Overlay – it means that the sky texture is rendered after the rendering of the main backdrop episode picture. This one, on the contrary, cannot contain any transparent pixels.

Birds animation – the choice specifying what animations of birds are set in the chapter by default.

The description explains itself.

Other settings:

- Animated crater of volcano – the animation of the volcano peak will be used at the same position as in the episode 8
- Earthquake effect – earthquake effect will be used like in episodes 8 and 9
- Echo for sounds – instead of classic basic sounds, the sounds with echoes will be used (usable for spacious interior episodes)

Clicking the button  saves all the changes you've made in this window.

The heights of the underlay's of skies:

Jungle: 345px.

Night: 345px.

Post-apo: 450px.

Lava: 68px. (This texture is, unlike other textures, aligned to the bottom side of the screen).



If the project is locked, the access to advanced settings is not possible.

7.4.2. Create copy

Opens a dialog box where the ID of the current active object and the ID of the copy created from this project will be listed.

Checking the choice “Set this copy as active project after creating” it sets the project as active project right after it’s been created. The process of creating a project copy is shown in a standard window with graphic and text information about percentage progress of copy creating.

Clicking the button  initiates the copy creating.



The function of copy creating copies entirely all files contained within the folder of the active project. So, even if you have in it your own files or files for game modding, they will be copied too. The new project created as a copy will have exactly the same parameters as the original one.



Copies cannot be created from a locked imported project! See more in the chapter [Export project](#).



System and key data files, however, will have whilst creating a copy a new hashed unique identifying number of the project assigned! Copying any files from the folder **pqxd** between respective project, apart from using the function of copy creating is not allowed! If you carry out copying of these system files, the project will be highlighted as erroneous with critical errors in data identification! So, never ever mess with the folder **pqxd**!

7.4.3. Export project

This is one of the most powerfully function of Project manager! Even if it might seem that the dialog box for options selections is relatively simple, it allows you to create 5 different types of export!

The description of respective selections of the dialog box of project exporting:

- Lock exported project – locks an exported project and prevents it from being edited
- Export saveslots – in the project file all saveslots will be contained that you’ve created whilst testing and playing the project
- Export levels statistic – in the project file all statistics data will be contained to all the levels of project

Thorough description of the respective selections:

7.4.3.1. Lock exported project

If there’s at least one playable project in the level, then the exported file will be locked against editing when this option is checked. This means that if someone imports your exported project, they’ll have no options of editing it apart from the project name and description. Also in QX EFF they’ll have no choice for entering the editor level in the game menu. They’ll only have the ability to launch the first playable level in the project and play it in a regular gaming mode, etc. until the last playable level. The whole progress in the game is, therefore, similar to the one in full Quadrax.

This mode (locked project) is therefore used for project presentation that should be used as “final game”, and the player who imports the project will have to solve all playable levels listed in the project. Levels different to the ones listed as playable will be ignored (regardless of whether they’re empty or contain some levels or your trials).



Playable levels don’t have to be exactly sorted in the project, because the engine of QX EFF chooses only the ones listed marked as payable. Nevertheless, **I strongly recommend** in order not to perplex a contingent player to **sort all playable levels of respective episodes consecutively one by one**. If you, for instance, have in each episode five playable levels, it would be more natural if players saw the levels in the way 1, 2, 3, 4, 5, 11, 12, 13, 14, 15 etc... than 1, 3, 5, 6, 8, 12, 14, 15, 17, 20... etc.



If there is no playable level in the project, Project manager will not allow exporting project as locked and if an attempt occurs, it will show a respective error message.



Active selection “Lock exported project” automatically closes next two selections in the dialog box. In the export of locked project there may be no saveslots or level statistics contained. (Otherwise it wouldn’t make sense, would it...)



A locked imported project cannot be again re-exported, nor can copies be created from it. It’s used only for normal gameplay in the gaming mode.



Important note! A locked project has all file layers and all texts of the clue and diary encrypted and saved in a proprietary data format! Player who plays the project cannot therefore (even if he or she wanted to cheat) look either at pictures or at texts of hints and diary.

7.4.3.2. Export saveslots

Checking this option inserts all saveslots of all levels that the project contains. Use this option only for creating project backups or if you really want to, to give the counterparty all your saveslots.

7.4.3.3. Export levels statistic

Checking this option inserts all statistics data of all levels that the project contains. Use this option only for creating project backups or if you really want to, to give the counterparty all your statistics data.



Both the options mentioned above I recommend using if you want to keep the project export as its backup. After importing, you'll have the project exactly in the same state it was in during the export. To the exporting project file COMPLETELY ALL files are saved that are listed in the folder of the respective export. So if you carry out game modding, the export will contain all files or folders that you create in it. This is why project exporting is suitable for creating complete project backups.


7.4.3.4. Further information about file exporting

Project manager automatically compresses all files of the entire project whilst creating the project, apart from **source PNG** files for level layers and added musical compositions in the formats MP3 and OGG. That's because these files are compressed on their own.



The size of the resultant export file is strongly dependent on the amount of data in the project. If the project contains many levels with full graphics and additionally some of your own music, the resultant size of exported file can be measured even in hundreds of MB! (Just to clarify, the graphic files of levels and music of the episodes in Quadrax X have the size of approximately 530 MB!)

7.4.3.5. Creating an export file

If you have chosen the options of the dialog box for project exporting, click the button  and creating of the source file of project export will initiate. Depending on the amount of data in the project, the creating of the exporting file can take a rather long time, since Project manager carries out the compression of all source files of the project. The progress of creating the exporting file is, as always during the similar situations, shown in a window informing about the percentage status of creating the export file.

After successfully creating the exporting file, an informing window is shown and after it's closed, a standard dialog box of Windows for saving exporting file is shown. It'll have a name looking like **QX_EFF_projekt.qxprj**. Choose an opportune name for the export file and save it where you find it convenient. After a successful save, a message about the correct and flawless finish of the entire procedure of the project is shown.

And with this you've successfully exported your project. Easy, isn't it?

7.4.4. Check project

This option launches (after clicking the enquiring window) complete diagnostics of the active project. If this situation occurs to you that QX EFF shows a message during its launch that the active project is damaged, you can, using the diagnostics project tool, find out where the error is. Project diagnostics checks whether there are all necessary project files listed in the project folder and checks their lengths and content. It also checks the integrity of data of key system project files and checks the integrity of unique UID of all objects in data of the files. If it finds any critical error in the data (missing file, damaged integrity etc.), it will announce it with a respective message. If the project is all right,

it'll only show the message "Checking of project XXXX was successful and there were no critical errors."



Critical errors can be created if you for instance erase some of the files **XX_layerX.png** that the game uses for showing level graphics. This can be easily fixed if you just load that file to the project. It's worse with the errors of integrity of system project files. This can happen only rarely, when, for example, your PC or hard drive crashes or when someone meddles with the system project files to find out "how it works". So I strongly warn you not to perform any operations with system project files that are saved in the folder **pqxd** of each project!



We can conclude from what was mentioned above that it's good from time to time to back up the entire project by creating an export of it. In case an accident with a computer or hard drive occurs to you, you can easily back up your project with the import.

7.4.5. Delete project

This option shows the dialog box with the double enquiry asking if you really wish to delete the project. If you really wish to irreversibly erase and delete a project, click the button and then the button . The active object will be deleted and Project manager will set the project with the nearest higher (in case it was the last project in a numeral queue, then with the lower) ID as active.



Deleting project irreversibly erase the entire project folder with all files, folder and subfolders that are inside it! Use this function only if you really wish to completely delete the project with all its data.



The default project with the ID 0000 cannot be deleted. It's a protection to assure that there will always be at least one active project in the entire QX EFF.

7.5. Language

Project manager will launch in a language you've chosen during the game installation or in QX EFF.



If Project manager (accidentally) launches in Czech, so the menu has an item called "Jazyk", selecting "Přepnout do angličtiny" from this menu switches the entire interface to English.

7.6. Info

Once again another simple item of the menu with one choice: “About”. If it’s chosen, a dialog box with concise information about the program will pop up. The most important data is probably the version number at the end of the first line amongst the window. The version of Project manager always must be the same as the version of QX EFF that is usually shown in the top right-hand corner of the basic game screen.

8. Project publications from QX EFF in Quadrax Neverending

If you decide to create levels and episodes in the mode that allows the figure movements like in Quadrax Neverending (QN), you'll definitely consider publishing your project and sending it to QN. But like I said at the end of the chapter [Allowed "Quadrax Neverending" moves](#), QN can be enriched only with the levels that comply with rather demanding criteria:

- Resourceful and innovative levels and points in them
- Flawless graphic visualization (including the episode backdrop and results of statistics)
- Good and convenient background music
- Interesting story corresponding with the overall storyline of QN
- Good translation of all texts into English (i.e. clue scrolls and the story)
- Overall consistency of the style of the entire episode – i.e. from level outlines through graphics to music

And the last, but most important criterion is that when a project author wants to assign his or her project to QN, he or she mustn't provide it as a separate project for importing to QX EFF to anyone! If I decide on assigning an episode to QN, it will be published only there and nowhere else! If someone begs to differ with this decision, it is no use trying to assign your project to QN.

The last arbiter deciding on assigning/not assigning a contingent project to QX is me. As the author of Quadrax II, III, IV, V, VI, VII, VIII, X, QX EFF and at last of QN I'm going to keep that right ;-). I reckon that during those twenty years I spent with Quadrax I've gained enough experience to assess the quality of other creators. ;-)

If you, despite this all, desire to create levels/episodes for QN, the best thing to do is introduce me to your idea. My email contact is on all websites of all my Quadraxes.

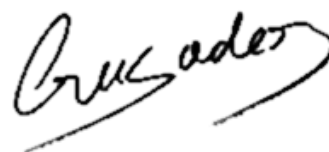
9. Conclusion

When I started typing more than three years ago the first lines of the source code of Quadrax X, I decided that the level editor with which I would be creating the levels for QX would be “a little bit” more comfortable to use and more perfect than the editors for previous Quadraxes. I kept adding pretty many functions, and since I was an active user of the editor, I quickly came to grips with all the settings.

But now when I finished writing this user guide I realized how vast and complex (compared to the editor of QVIII and below) the project got throughout those three years. An inexperienced user who first encounters QX EFF could be left disillusioned at the “size” of this manual. Even so, I can assure that with regard to computer programs and editors, QX EFF is a dwarf. And even with the 85 pages of this manual. What should the users learning to use programs such as CAD say when their manuals often exceed the size of a thousand pages?! Or did someone really think that the editor would contain only one colored button with the sign “Make me a perfect and graphically enchanting level”? Probably not, right? ;-)

Just like at all areas of human activity, if someone wants to create something, they have to learn to use the tool that creates what they want. A musician can have wonderful melodies dingling in his or her head, but if he or she doesn’t learn to play a musical instrument, they’ll stay in his or her mind only. A beginning architect can have a heap of ideas for the most amazing houses of the world, but if he or she doesn’t learn something about constructions of houses, materials and many of other fineness, these houses will stay in his or her mind only. And if your mind is full of ideas for creating entrancing levels for Quadrax, then without learning to use this editor, they’ll stay only in your mind.

At the end I would like to wish you good luck with creating your own levels and don’t get discouraged by first failures. And most importantly, endure. Rome wasn’t built in a day.

A handwritten signature in black ink that reads "Crusader". The signature is stylized with a large, sweeping 'C' and a long, horizontal stroke extending to the right.