The Exile Graph, 1: The 2001 Game as a Graph

From a world of views to a world of places

Guillaume Lethuillier
glthr.com

November 21, 2026 · part of the series “The Exile Graph”

> **Work in progress — November 21, 2026**
> I am still working in **Myst III: Exile**. I will update these articles over the coming weeks, so please feel free to come back later to see what’s new.

Abstract

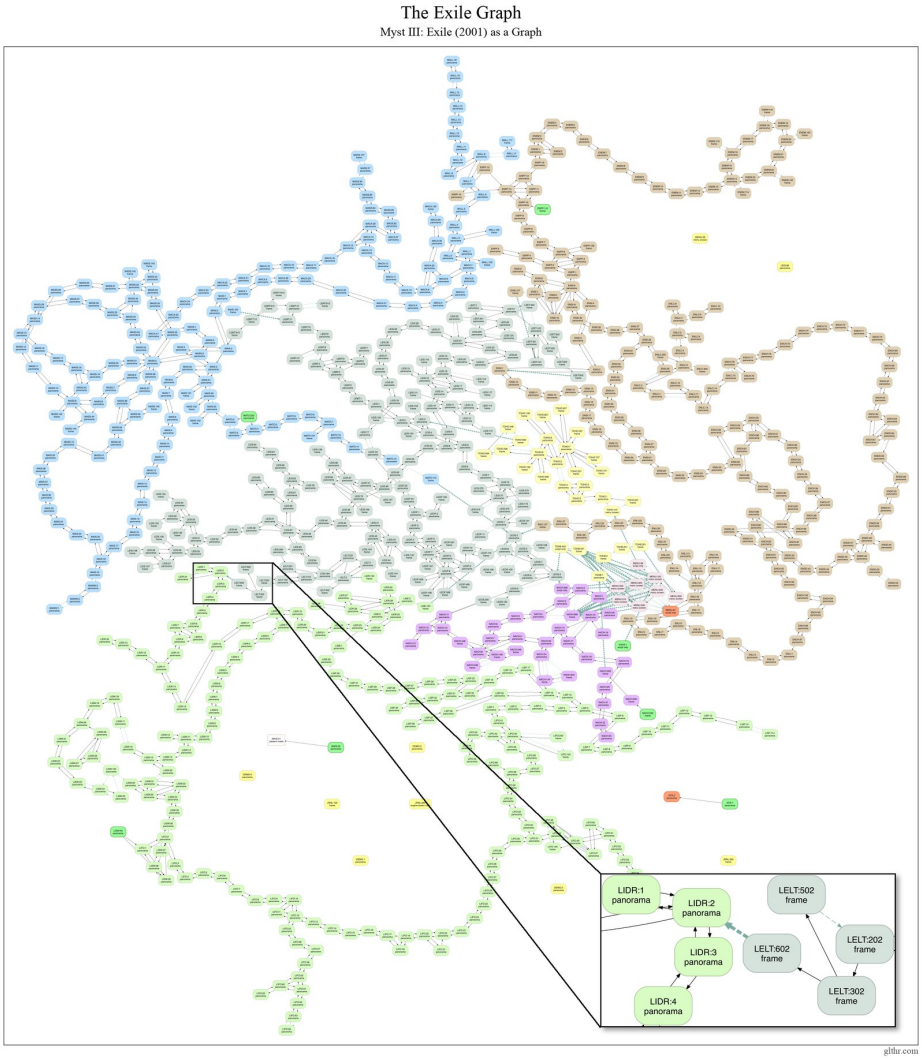
I represent the original Myst III: Exile (2001) as a directed graph of 820 places and one virtual target. Its 2,033 typed connections preserve script references, separately attributed engine transitions and continuations inside a script. The default view hides Zip shortcuts. This first article explains the graph and its browser view, and why a world of comparable size to Riven has a third of its nodes. It describes structural relationships, without executing the puzzle state that determines which ones a player can follow.

After Myst and Riven, the third game of the series was the obvious next step. It was also the one I expected to resist. Exile was not made by Cyan but by Presto Studios, with another engine, and its world is no longer a succession of still views: the player stands in a place and looks all around.

Exile can nevertheless be viewed as a graph, like the two games before it:

Version of the game: Myst III: Exile 1.0, the original English four-CD release, May 2001. The node numbers are those of this release.

Cite as: Guillaume Lethuillier, “The Exile Graph, 1: The 2001 Game as a Graph”, glthr.com, November 21, 2026. https://glthr.com/exile-graph-1



The Exile Graph can be **explored in the browser**, where any node can be found by its room and number.

Disclaimer

This is an unofficial, non-commercial study of Myst III: Exile (2001), independent of Cyan Worlds, Presto Studios and Ubisoft. The game artwork remains the work of its creators. The analysis contains spoilers and reads the released data without executing every possible world state. Structural references, inferred engine behavior and directly compared images are identified separately.

Introduction

Roadmap

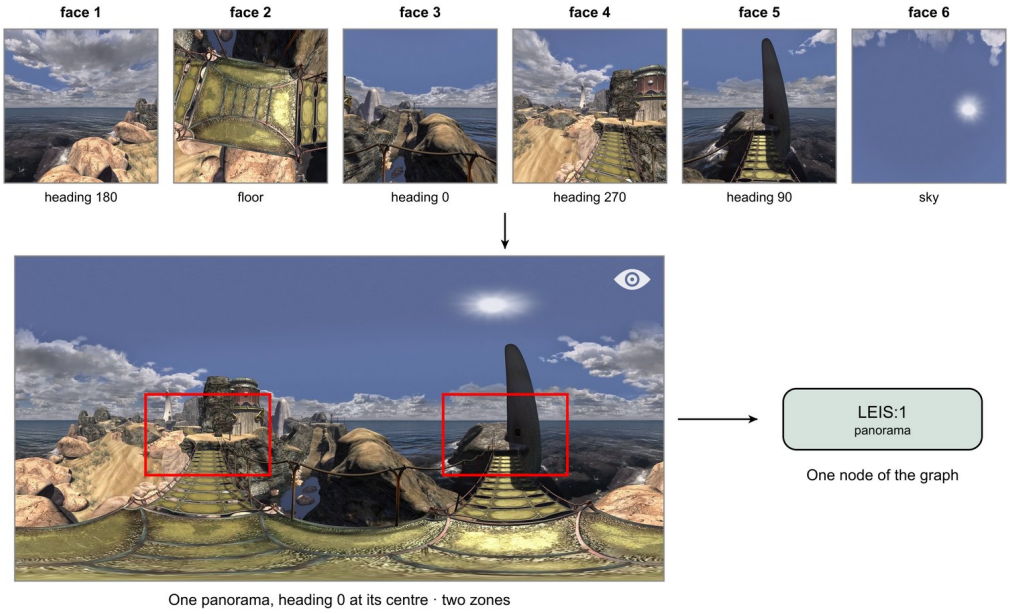
(1) In this article, I introduce the Exile Graph and explain its nodes, arrows and colors. The Scripts Are in the Program explains where the game keeps its data and how the graph was extracted.

(2) In the second article, I compare the graph with the game's pictures and scripts: theoretical paths, an ending that hangs from a book, panoramas that nothing selects, a node that does not exist, and two rooms left in the program without their pictures. The figures use the same node labels, graph excerpts and eye indicators as the Myst and Riven findings.

A Node Is a Place, Not a View

In Myst and in Riven, a card is one view: the player faces north, and turning to the east loads another card. A place where one can look in four directions is four cards, joined by turns.

Exile works differently. **Each place is a node**, and most nodes are panoramas: six square pictures, the faces of a cube seen from its center, that the game stitches around the player. Turning does not load anything. It happens inside the node.



A place of J'nanin as the game stores it (six faces), as this study displays it (one panorama, heading 0 at its center) and as the graph draws it (one box).

This explains a figure that surprised me. **Riven has 2,307 cards. Exile, four years later and on four discs, has 820 nodes.** The comparison is unfair to Exile: a Riven card is one view of 608 by 392 pixels, and an Exile panorama is a whole sphere, stored as six pictures of 640 by 640. Grouping the cards of Riven that a turn joins leaves about 950 groups. Exile has 692 panoramas and 113 flat frames. The two worlds have a comparable number of places where one can stand.

The figure for Riven is an estimate: it joins the cards linked by a hotspot named left, right or back in the 1.0 release, and counts the groups that remain.

It also changes what the graph says. In Myst and Riven, turning your head was a graph traversal; in Exile, it is just looking around, so **every connection is a move**: a step forward, a ladder, a door, a ride.

What Is in This Graph?

The game groups its nodes into rooms, and its rooms into Ages. A room has a name of four letters and a number: `LEIS` is room 501. The first two letters follow the Age:

`TO` for Tomahna, `LE` for J'nanin, `EN` for Voltaic, `MA` for Amateria, `LI` for Edanna and `NA` for Narayan.

Age	Rooms (places)	Places	Panoramas	Frames
Tomahna	TOHO (23), TOHB (6)	29	9	18
J'nanin	LEIS (108), LEOS (14), LEET (11), LELT (11), LEMT (18), LEOF (23)	185	134	51
Voltaic	ENSI (42), ENPP (18), ENEM (28), ENLC (17), ENDD (18), ENCH (47), ENLI (28)	198	187	11
Amateria	MACA (35), MAIS (36), MALL (21), MASS (57), MAWW (3), MATO (13)	165	149	16
Edanna	LIDR (33), LISW (41), LIFO (70), LISP (37), LINE (5)	186	181	5
Narayan	NACH (37)	37	26	10
Menu and books	MENU (10), JRNL (3), DEMO (4), ATIX (2)	19	6	2
System	XXXX (1)	1	0	0
Total		**820**	**692**	**113**

The fifteen places that are neither panoramas nor frames are nine menu screens, five nodes that only run a script, and one journal that the engine draws itself. One virtual target brings the total to 821 nodes. Room and Age are attributes used for labels, colors and filters: they have no vertices of their own.

The program holds more nodes than the graph: 1,169 in this release. The 349 others are never visited. They are **helper scripts**, numbered like nodes and called by the scripts of real places: `LEIS:10084` plays a movie and moves the player on behalf of `LEIS:84`. The graph follows these calls and attributes each move to the place whose hotspot started it.

Basic Properties of the Exile Graph

Node labels such as LEIS:1 have a thick underline, in the color of their Age, and open that node in the graph browser.

Nodes and Edges

The graph has 821 nodes: 820 places of the game and the virtual target described below. Its 2,033 typed connections are summarized in the table of connections. The default drawing shows 1,828 of them.

Paths

From the first view of the game (TOHO:101), the arrival in Narayan (NACH:11) is 23 edges away. The two most distant nodes of the graph are 189 edges apart (125 in Riven, 130 in Myst). Zip shortcuts and continuations are excluded from these distances. As in Myst and Riven, a path is a distance between places and says nothing about the puzzles that stand on it. The second article gives the routes.

Clusters

The drawing groups nodes by Age, and here the grouping is nearly exact: only 33 pairs of nodes belong to two different Ages, and 91 to two different rooms. Counting the isolated nodes, the graph has 12 weakly connected components, but one of them holds 808 of the 821 nodes, and 794 places can all reach one another. The first view reaches 797 nodes. The 24 that it does not reach include the node where the program starts, one screen of the menu, the journals, the end of the game and a few places that the second article examines.

J'nanin sits in the middle. Voltaic, Amateria and Edanna are entered from it and return to it, and Narayan is reached from it, which makes it the hub that Myst Island was in the first game.

Legend: How to Read the Graph

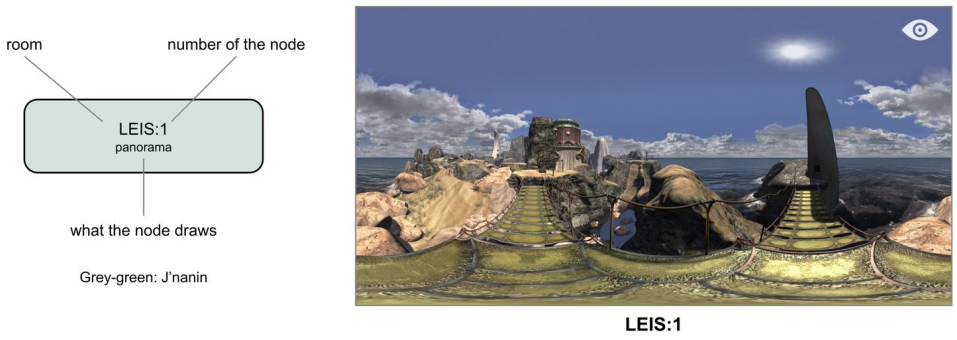
The browser graph uses a node's Age for its color, its references for its connections, and distinct borders or arrowheads for special cases. The following legend explains those conventions before the findings put them to use.

Reading a Node

Labels and Colors

Each Age has its own color. A label such as LEIS:1 means node 1 of the room `LEIS`, in J'nanin. A node number needs its room: LEIS:1 and ENSI:1 are different places. Unlike the cards of Myst and the pictures of Riven, the nodes of Exile have no names: the room and the number are all that the program stores.

Under the label, each box says what the node draws: a panorama, a frame, a menu screen, or nothing of its own.



The label of LEIS:1 and its view. The open eye identifies a view that the game displays, following the figure convention of the findings.

Panoramas, Frames and Several Views

A **panorama** is a place seen all around. A **frame** is a flat picture of 640 by 360 pixels: a close-up of a mechanism, a page, a movie that fills the screen. A frame cannot be turned in.

As in Riven, a node can hold several views. The 692 panoramic nodes name 901 panoramas (894 of them stored on the discs; the seven others belong to leftover

rooms and one missing node): 150 of them have more than one, chosen by a variable when the node is entered. LEOS:1, in the observatory of J'nanin, has nine. Opening a door or draining a pool changes the panorama without changing the node.

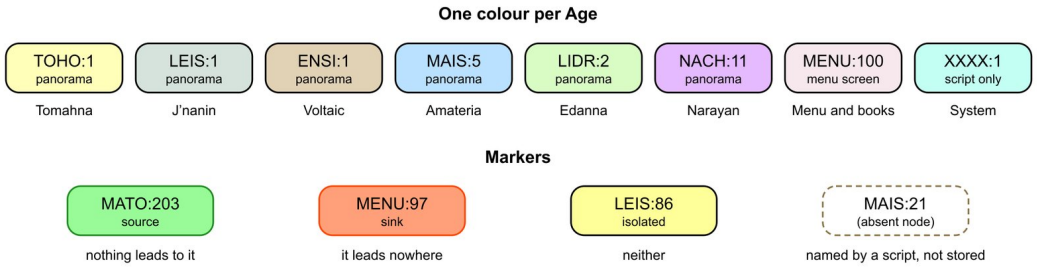
Sources, Sinks and Isolated Nodes

The usual graph terms apply. A **source** has no incoming connection from another node; a **sink** has no outgoing connection to another node; an **isolated** node has neither.

These properties describe a graph projection, without establishing access in the game. The journals are opened from the inventory and return to the place where they were opened. Leaving that mechanism unexpanded in the graph does not make them inaccessible.

Virtual Targets

One navigation instruction refers to a node that this release does not contain: MAIS:21. I keep that target as a virtual node, with a distinct border, rather than discard the reference. The second article comes back to it.



Reading a Connection

An arrow has a source and a destination. If both directions exist, the places are mutually connected. If only one exists, following it does not imply that the player can simply reverse the move.

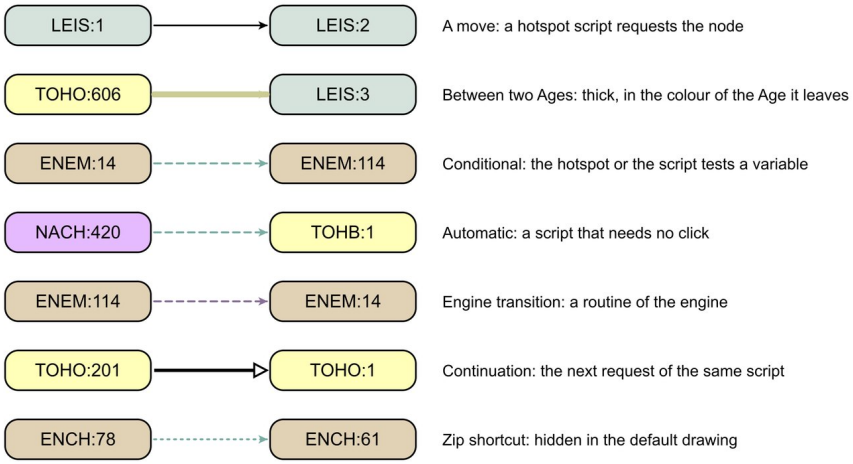
In Riven, the two directions did not mean the same thing: a turn was answered by a turn back (91.5% of the pairs), a step forward rarely by a step back (9.2%). Exile has no turns, and yet **90.4% of its 1,770 pairs of places have their reverse pair**, against 66.4% for Riven and 55.5% for Myst. The reason is in the panorama: after a step forward, the way back is simply behind the player, in the same node. The 170 one-way pairs are what a world really makes irreversible: a slide, a fall, a ride, a link.

There are several kinds of connection:

Kind	Typed connections	Meaning
Move	1,742	A hotspot script requests another node.
Automatic	91	A script that runs on arrival, or on every frame, requests another node.
Engine transition	11	Navigation handled by a reviewed engine routine.

Kind	Typed connections	Meaning
Continuation	6	One script enters an intermediate node, then requests the next one.
Zip shortcut	183	A potential shortcut whose destination must have been visited.
Total	**2,033**	All exported kinds together.

The default drawing shows 1,828 typed connections: Zip shortcuts are hidden, and so are 22 references from a node to itself (across 19 places). Path calculations also exclude the 6 continuations, leaving 1,770 unique ordered pairs of places.



From top to bottom: LEIS:1 → LEIS:2, TOHO:606 → LEIS:3, ENEM:14 → ENEM:114, NACH:420 → TOHB:1, ENEM:114 → ENEM:14 (through the engine), TOHO:201 → TOHO:1 (within the script of TOHO:101), and ENCH:78 → ENCH:61.

Several instructions can support the same edge. I merge references only when their source, destination, kind and conditional status agree, while keeping every contributing instruction in its evidence.

Local Conditions Remain Visible

A hotspot exists under a condition, and a script can test variables before it moves the player. The extractor keeps these tests beside the connection, with the instruction and its position in the program. Across all kinds, 316 connections are conditional. The tests are recorded and never evaluated: a conditional link is a structural possibility, not proof that its whole route can be played in one state of the world.

Scripts That Run by Themselves

A node of Exile has two kinds of script that need no click. One runs when the node is entered. The other runs on every frame for as long as the player stays: it watches a variable, a direction of the gaze, or the last frame of a movie. When such a script requests another node, the graph draws an automatic connection. TOHO:101, the first node of a new game, has no hotspot at all: its script plays the opening movie and moves on.

Hollow Arrows for a Continuing Script

Sometimes one script requests a node and then, straight away, another. The graph keeps both requests as ordinary arrows from the node that owns the script. It adds a third one, drawn with a hollow arrowhead, from the intermediate node to the next one, just as the Riven Graph does for continuing handlers. There are 6 such continuations. They explain why a view that is only passed through still has a way out. Path search ignores these arrows: a continuation belongs to its script and cannot become a shortcut for another node.

Some Links Belong to the Engine

Reading the scripts alone misses part of Exile's navigation. The main menu chooses its screens in the engine. Five panels of an electromagnet in Voltaic leave for "the node stored in variable 32", and no script stores anything there: a routine of the engine does, when it draws the panel. These eleven transitions are attributed to a pinned ScummVM implementation. They are separate from the scripts read from the discs.

The books receive different treatment. Opening a journal from the inventory saves the place where the player stands, and closing it returns there. Connecting the book to every possible place would introduce shortcuts through the whole world. I keep this mechanism as explicit metadata rather than invent such edges. The same holds for the main menu, which can be opened from anywhere. The second article shows that the end of the game depends on one of these books.

The Scripts Are in the Program

Myst keeps its scripts in HyperCard stacks, and Riven in Mohawk archives, beside the pictures. Exile's archives hold pictures and movies only. The Ages, the rooms, the nodes, their hotspots and their scripts are tables in the data of the program itself, `M3.exe`.

On the original discs, that program is protected against copying: two of its sections are encrypted. The ResidualVM project, since merged into ScummVM, had published the way to decode them. A table of ten internal Age entries (splitting Tomahna into opening and ending entries, with two system entries and one for menus and books) then leads to 33 rooms, and each room to lists of nodes with their conditions, their zones and their instructions: 18,338 instructions in all, for 2,486 hotspots.

A hotspot of a panorama is not a rectangle of the screen. It is a rectangle on the sphere, given by a direction (a pitch and a heading, in degrees) and two half-widths. The browser draws it on the unrolled panorama.

Conclusion

Exile changes what a node is, and the graph follows: fewer boxes, each of them a place, and arrows that all mean a move. The method is the one of Myst and Riven. What it shows is a world built differently: a hub, five Ages that each go their own way, and nine connections out of ten that can be walked back.

I updated DeMystify to generate this graph from the original 2001 game.

In the next article, I look at what does not fit: the places that nothing leads to, the panoramas that are stored and never shown, and the book that the end of the game depends on.

How to cite this article

> Guillaume Lethuillier, "The Exile Graph, 1: The 2001 Game as a Graph", glthr.com, November 21, 2026.
> https://glthr.com/exile-graph-1

For BibTeX:

```
@misc{lethuillier2026exile,
  author = {Lethuillier, Guillaume},
  title  = {The Exile Graph, 1: The 2001 Game as a Graph},
  year   = {2026},
  month  = nov,
  url    = {https://glthr.com/exile-graph-1},
  note   = {glthr.com}
}
```

Your questions and leads are welcome: myst@glthr.com

This document was typeset with LaTeX from the article published at https://glthr.com/exile-graph-1, which remains the reference version.