Inside Myst V's 2005 Neural Network

Why a chalk line sends a slate across Laki'ahn and an umbrella can make it rain in Noloben

Guillaume Lethuillier
glthr.com

November 22, 2026

Collection: Myst V: End of Ages (2005) — Behind the Console (alongside The End of Ages Graph)
Next: Ghosts of Uru (glthr.com/end-of-ages-hidden-worlds)
Interactive browser: glthr.com/end-of-ages-graph
All 5 Myst games: glthr.com/myst-graphs (Myst, Riven, Exile, Revelation, End of Ages)

Abstract

In *Myst V: End of Ages* (2005), Cyan trained four small neural networks to read the symbols on your slate, one for each Age. I drew the same N on all four slates. In three Ages it sent the slate to a pedestal. In Noloben it called for rain. You can try your own handwriting with the four readers below.

On the beach in **Myst V: End of Ages** (2005), I drew an umbrella on the stone slate and stepped back. One of the creatures that read the slate, a Bahro, came over, looked at my drawing and left it on the sand. I drew the umbrella again, this time with one raindrop. The Bahro raised its arms and the rain came down.

When the game came out, one of its makers said the slate "*works like any good handwriting recognition technology*".[1] Four years later a player on the Guild of Writers forum recognized the kind of network Cyan had used.[2]

I used the Limited Edition for Mac sold by GOG, with its September 2024 update.

The four networks were still there in my copy of the game. I could give each one the same drawing and compare its answer with what the Bahro did on screen.

The files belong to a version of Cyan's engine that stayed closed when the source for *Myst Online: Uru Live* was released. I found no published account of how this version reads a drawing beyond that forum remark.

The Slate, Step by Step

To see what happens with a symbol the Bahro is supposed to recognize, start in Laki'ahn.

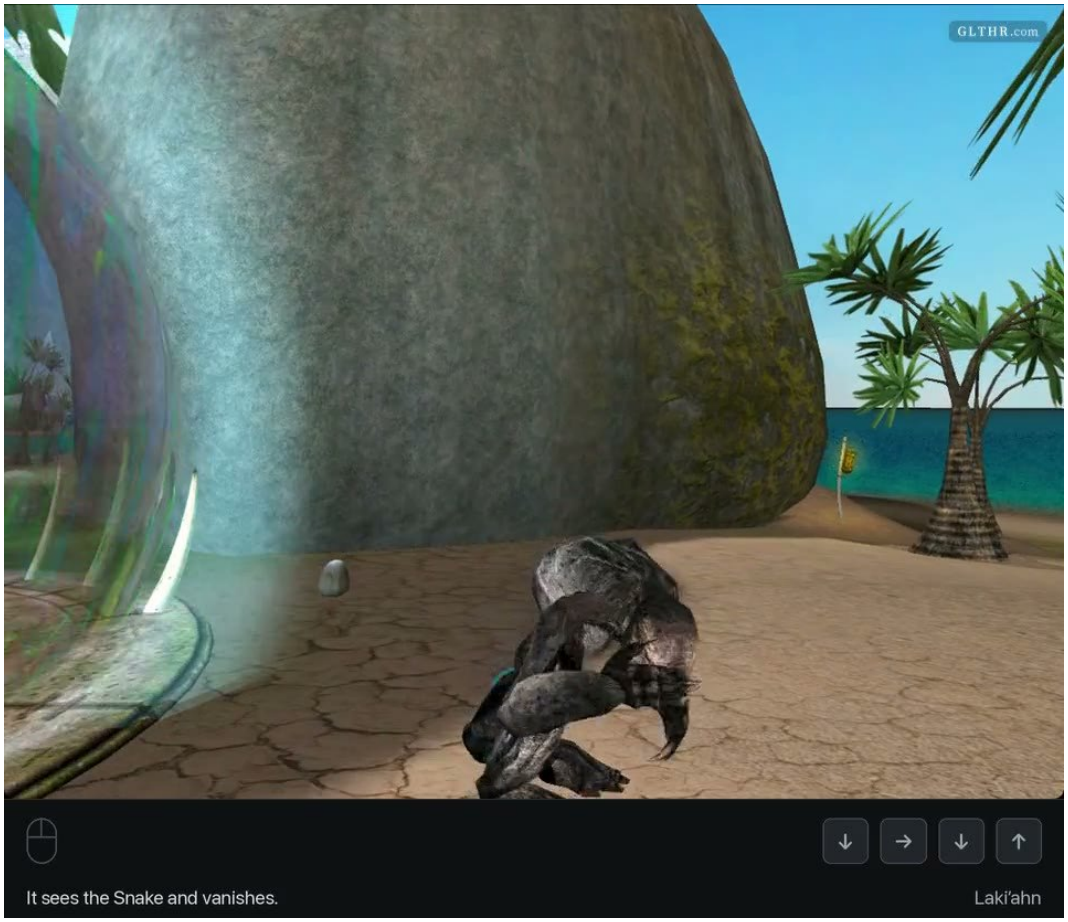
Cite as: Guillaume Lethuillier, "Inside Myst V's 2005 Neural Network", glthr.com, November 22, 2026. https://glthr.com/end-of-ages-slate-neural-network

Interactive figures, animations and videos of the web page are shown here as still images or addresses.

[1] Rainier, "'Myst V: End of Ages' Developer Q&A #2", Worthplaying, July 30, 2005. The answers are signed "RM": "players can write anything they want on the Slate, although only certain commands will be registered by the system and recognized."

[2] ddb174, "Re: Converting Myst 5", Guild of Writers forum, September 8, 2009, in a thread about Drizzle, a tool that converts the Ages of *Myst V* to run in *Uru*. The post explains why the puzzles of the converted Ages cannot be played and says the slate "uses a simple perceptron ai to recognise symbols".

1. **Pick up the slate.** The stone tablet rests on a pedestal. Lifting it opens it full-screen.
2. **Draw.** Sketch a symbol with the mouse. In the clip below, it is the winding Snake.
3. **Drop it.** Click the carved border, and the slate goes back to the ground.
4. **Wait for the verdict.** Walk away from the slate, and a Bahro comes to read the drawing. It either carries out a command or walks away.



Video: https://media.glthr.com/end-of-ages-slate-neural-network/eoa-slate-intro-4.mp4

The Bahro comes up to the slate, sees the Snake and vanishes. The game gave my drawing a score above 0.99 for Snake.

The game also scored my Snake as Wind, Friendship and every other symbol in Laki'ahn. Each gets its own number between 0 and 1. The Bahro acts only when the winner has a clear lead.

Each Age has its own network, and each network knows ten or eleven symbols. For reference, here they are as the game itself shows them: carved on the pedestals of Direbo, on the walls of the Ages, or sketched in the journal.



The symbols that each network was trained to read, from the game's own pictures. Compare the pedestal and Keep signs across the four columns. The Snake is the only one without a picture in the files: it is shown as the wavy line drawn in the first clip.

How Myst V Reads a Drawing

TL;DR

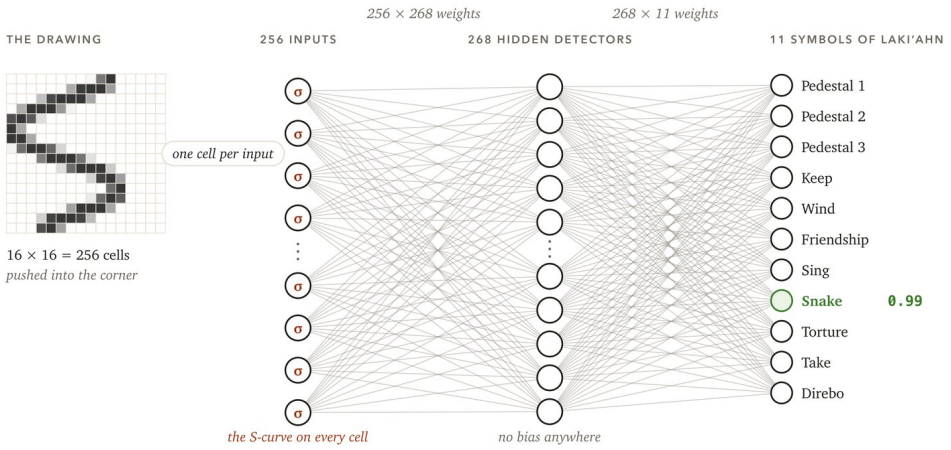
The game shrinks your drawing onto a small grid, throws out scribbles, lets a neural network score every symbol of the Age, and obeys only a clear winner.

When you drop the slate, your sketch passes through four stages before the Bahro decides what to do:

1. **Shrink onto a 16 × 16 grid.** The game draws a tight box around everything you sketched, scales the longest side to 16 cells while keeping the drawing's proportions, and pushes the result into the **top-left corner**. A narrow vertical stroke ends up against the left edge, and a flat one against the top edge. Each cell holds a brightness from 0 (blank stone) to 1 (ink).
2. **Catch scribbles first.** Before the network runs, the game counts how many of the 256 cells hold any ink. At **40% or more**, it gives up at once and calls the drawing a **Scribble**, whatever the network might have answered.

3. **Let the network score every symbol.** The 256 cells feed **268 hidden detectors**. Each detector weighs every cell, for or against, and reports how strongly a certain mix of curves, corners and empty space is present. The detectors then vote on the **10 or 11 symbols** of the Age, and each symbol gets a score between 0 and 1.
4. **Require a clear winner.** Scoring highest is not enough. The game subtracts the score of the runner-up from the best score, and accepts the drawing only if the gap is at least **0.70**. Anything closer, and the Bahro shrugs: **Unknown**.

A network of this shape, with its cells, its hidden detectors and its symbols, is a **three-layer perceptron**.



The network of Laki'ahn: every cell of the drawing reaches every hidden detector, and every detector votes for every symbol. Each line is one weight. Only a few of the 256 inputs and of the 268 detectors are drawn.

You can try it below. Everything runs in your browser, from the weights that Cyan shipped and from the game's own way of shrinking a drawing. Try a vertical line in Laki'ahn, or load a dense scribble in Tahgira to watch the 40% cutoff overrule a confident answer of the network.

This figure runs in the browser: draw on the slate, and watch the grid, the S-curve, the hidden detectors and the scores of the Age change as you draw.

An S-Curve at the Entrance

Look at the little S on each input in the diagram. It stands for a curve called a *sigmoid*, which changes the brightness of each cell before the detectors get to see it. That is an unusual place to put one.

A blank cell (0) enters the network as **0.5**, and a fully inked cell (1) as about **0.73**. Instead of white ink on black stone, the network only ever sees pale grey strokes on medium grey stone.

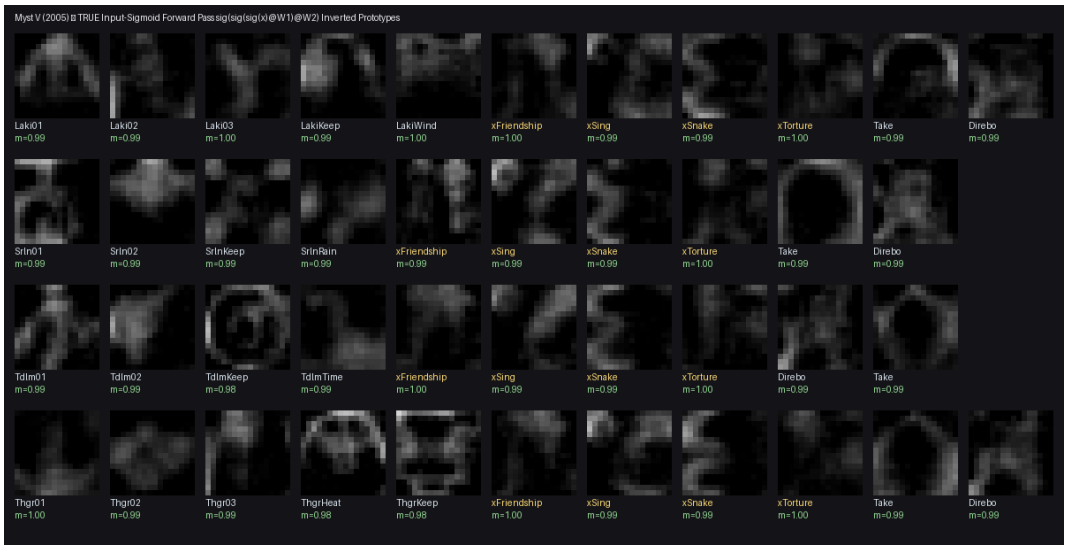
All four files give the first layer the same curve as the later ones. My guess is that the first layer inherited it along with the others. The files record the choice and give no reason for it.

The game works with it all the same, and it may even gain something. Cyan's network has none of the constant offsets, called biases, that such networks usually

carry. A background of 0.5 on every cell can play that part. You can switch the step off in the figure above, on Todelmer's empty slate: with the sigmoid off, Friendship rises to about 0.3; with the sigmoid on, no symbol rises above 0.04. The comparison is not quite fair, since the weights were trained with the step in place. I repeated the comparison with freshly trained weights in the handwritten-digit experiment.

What the Weights Remember

What would the network draw if I asked it for a Snake? I started with a blank grid and changed its cells a little at a time, raising the score for Snake and lowering the others. I did the same for each symbol in each Age. Each square below is a drawing that satisfies one reader. The original sketches used to train it are a different question.



The 42 symbols, drawn by the weights themselves (row 1: Laki'ahn, row 2: Noloben, row 3: Todelmer, row 4: Tahgira). Tall narrow glyphs such as the Snake sit against the left edge, and wide flat ones such as Laki'ahn's Wind against the top edge: the mark of drawings pushed into the top-left corner.

The six commands that the four Ages share (Take, Snake, Sing, Friendship, Torture and Direbo) come out with the same shape, at the same place, in all four networks, although each network was trained apart.

Strokes That Fool the Bahro

TL;DR

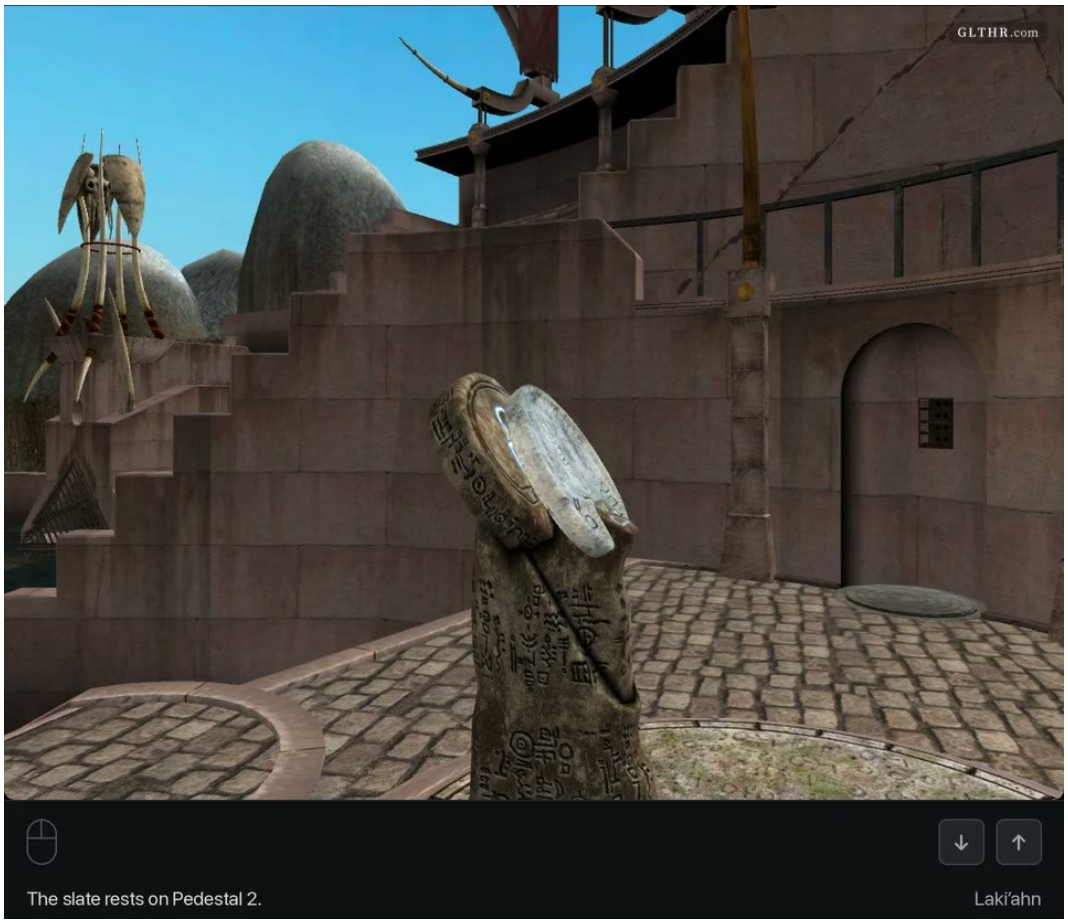
A single stroke can give a different order in each Age. Four dots are enough for Torture in Todelmer, and an umbrella with one raindrop brings the rain in Noloben.

The game can answer "Unknown". The network itself only scores the symbols it knows. If a doodle happens to give one symbol a big enough lead, the game accepts it. Here are four you can try.

- A single vertical line in Laki'ahn is read as **Pedestal 2**: a narrow stroke is pushed against the left edge of the grid, where that symbol lives.

- A plain V in Laki'ahn is read as **Torture**.
- This plain circle gives **Take** in Noloben and Laki'ahn, and **Unknown** in Tahgira. In Todelmer I got **Keep** once and **Unknown** on my other tries.
- A plain square means **Take** in Noloben and **Snake** in Todelmer.

The shortcuts work in the game itself:



Video: https://media.glthr.com/end-of-ages-slate-neural-network/eoa-slate-shortcuts-4.mp4

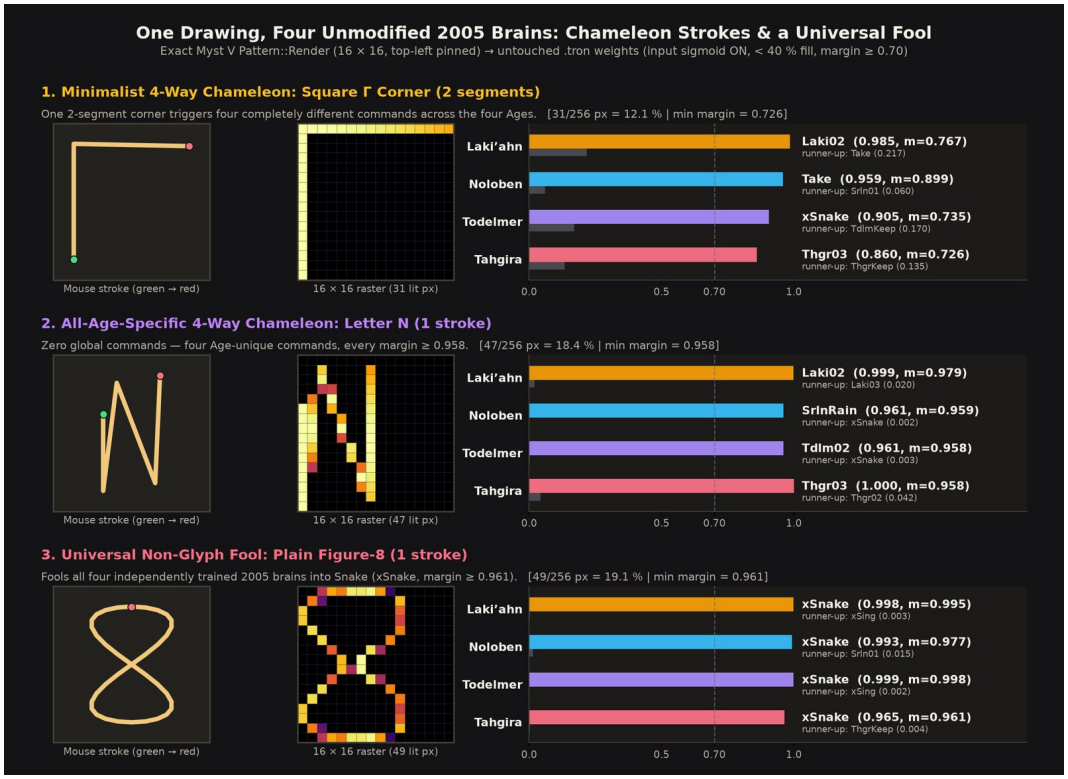
In the running game: a single vertical line on Laki'ahn's slate scored 0.96 for Pedestal 2, far ahead of the runner-up, and a Bahro took the slate away to that pedestal.

One Stroke, Four Meanings

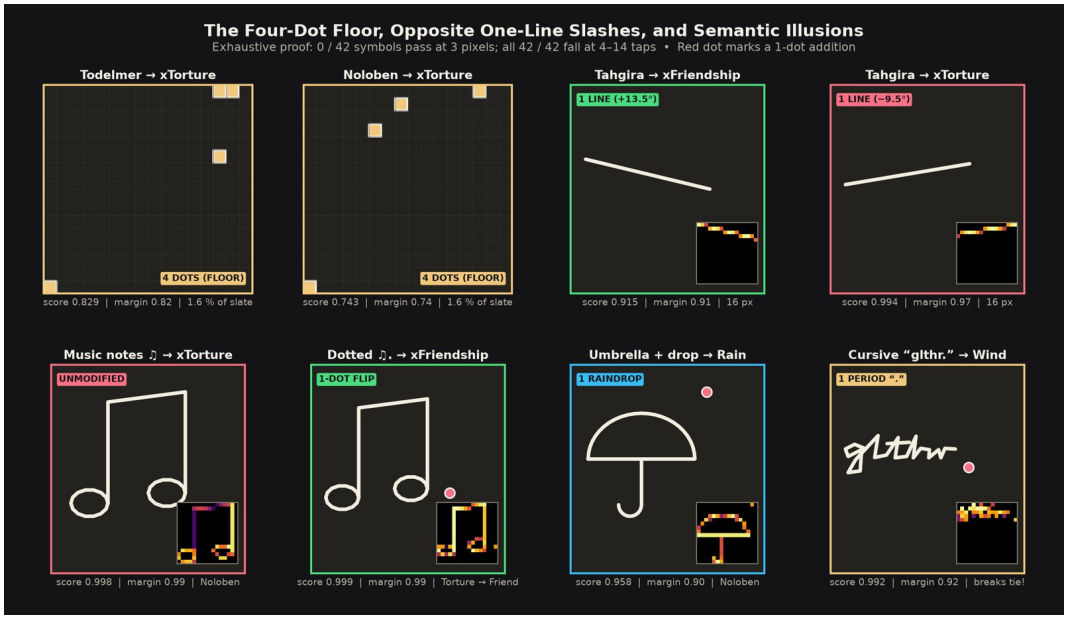
The four networks were trained apart, on the same small grid, and they do not draw the borders between symbols at the same places. One stroke can then mean a different command in each Age:

- A square corner of two segments (Γ) means Pedestal 2 in Laki'ahn, Take in Noloben, Snake in Todelmer, and Pedestal 3 in Tahgira.
- A letter N in one stroke sends the slate to a local pedestal in three Ages and changes the weather in the fourth: Pedestal 2 in Laki'ahn, Rain in Noloben, Pedestal 2 in Todelmer, and Pedestal 3 in Tahgira.

- A plain figure 8 goes the other way: it is a **Snake** for all four.



The same strokes, read by the four networks: a Γ corner and an N give four different commands in Laki'ahn, Noloben, Todelmer and Tahgira, and a figure 8 is a Snake for all four.



Top: four dots that command a Bahro, and the two opposite slashes of Tahgira. Bottom: doodles and words as the game shrinks them (insets), with a red dot where one added dot changes the answer.

I drew the same N on each of the four slates in the running game. The four commands came out as computed:



One N, four Ages, in the running game: Laki'ahn (Pedestal 2, 0.99), Noloben (Rain, 0.72), Todelmer (Pedestal 2, 0.90) and Tahgira (Pedestal 3, 0.99).

Doodles and Words

Everyday drawings land somewhere too:

- An umbrella with one raindrop beside it brings **Rain** in Noloben. Without the raindrop, the Bahro shrugs.
- Beamed musical notes (♫) read as **Torture** in Noloben, and one dot beside them turns Laki'ahn's answer from Torture to **Friendship**.
- **"AI"** in Todelmer and **"<3"** in Laki'ahn both read as **Torture**. I wrote Yeesha's name with one E, **"YESHA"**, and got **Friendship** in three of the four Ages. **"CYAN"** tells the Bahro of Noloben to **Sing**.

Here is the umbrella in the running game, on Noloben's beach. The bare umbrella returns Unknown. With one raindrop, the slate is read as Rain, and a Bahro comes onto the sand to bring it:



Video: https://media.glthr.com/end-of-ages-slate-neural-network/eoa-slate-umbrella-3.mp4

An umbrella and one raindrop on Noloben's slate: Rain scores 0.97, far ahead of the runner-up, and the Bahro obliges.

The margin rule is what stands between the Bahro and plain nonsense. As a joke, I signed "glthr" in cursive on Laki'ahn's slate. My signature scored 0.98 for Wind and 0.96 for Pedestal 1. I have been called worse. The two scores were too close, and the game answered Unknown.

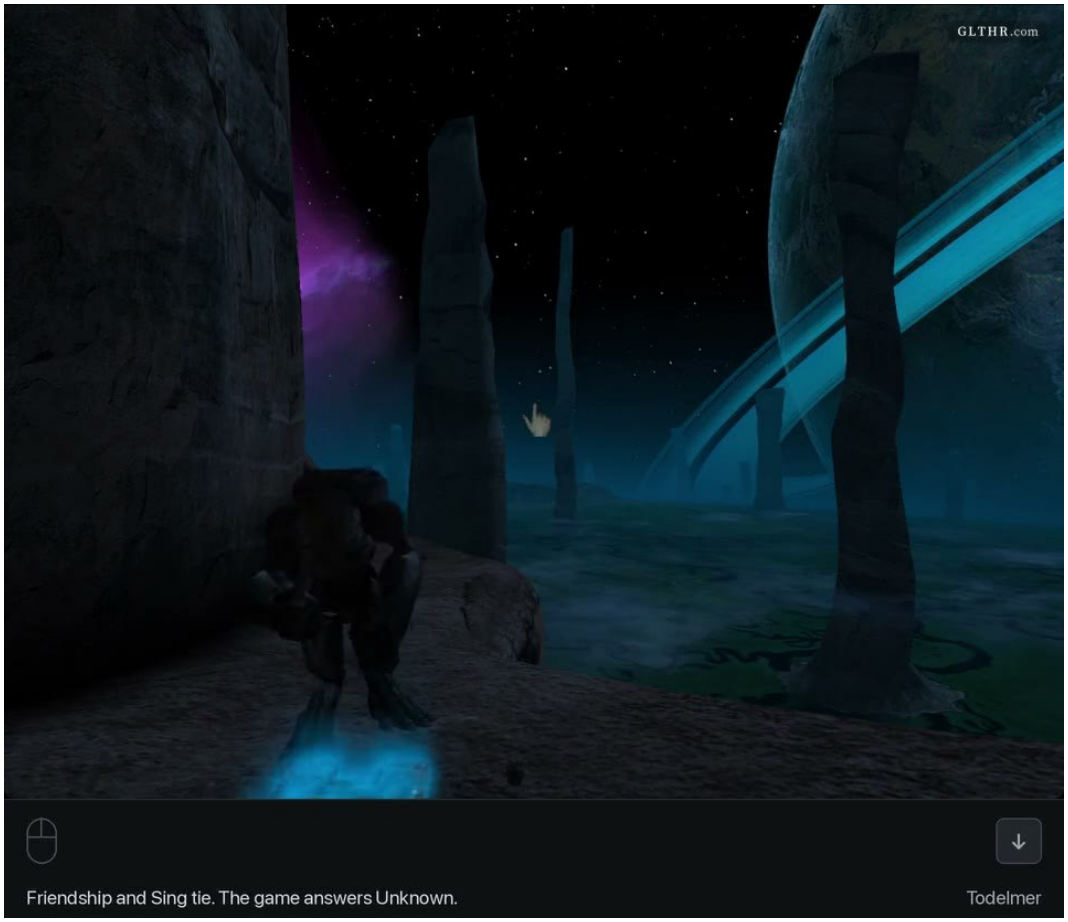


Video: https://media.glthr.com/end-of-ages-slate-neural-network/eoa-slate-glthr-3.mp4

Signing "glthr" in cursive on Laki'ahn's slate: Wind scores 0.98 and Pedestal 1 scores 0.96 at once, so the game answers Unknown.

One dot can settle such a tie, in either direction. In the figure, a period after the signature tips it to Wind. In the running game, with a more compact signature, the same period did the opposite: "glthr" was Wind, and "glthr." was Unknown. The box around the drawing grows with the dot, and every cell of the grid moves with it.

When the answer is Unknown, a Bahro still comes, but with no order to carry out. In Todelmer, "CYAN" scored 0.998 for Friendship and 0.997 for Sing, a near-perfect tie. The Bahro hesitated by the slate and left it where it lay:



Video: https://media.glthr.com/end-of-ages-slate-neural-network/eoa-slate-cyan-bahro.mp4

"CYAN" on Todelmer's slate: Friendship and Sing both score above 0.99, so the game answers Unknown, and the Bahro leaves the slate behind.

A Hidden Eraser

The slate carries an unfinished feature: an eraser gesture, named after Zorro inside the engine, meant to let players slash a Z across the stone to wipe it. It was left switched off in the shipped game. Slash a Z today, and the chalk stays. Clearing the slate takes a click on the erase button.



Video: https://media.glthr.com/end-of-ages-slate-neural-network/eoa-slate-scribble-and-zorro-2.mp4

A Z slashed across Laki'ahn's slate leaves the chalk in place, so clearing it takes the erase button. Filling half of the stone then trips the 40% scribble cutoff when the slate is dropped.

I tried the gesture again with the eraser enabled. A wide Z across the stone then wipes both the drawing and the Z itself, as soon as the mouse button is released:



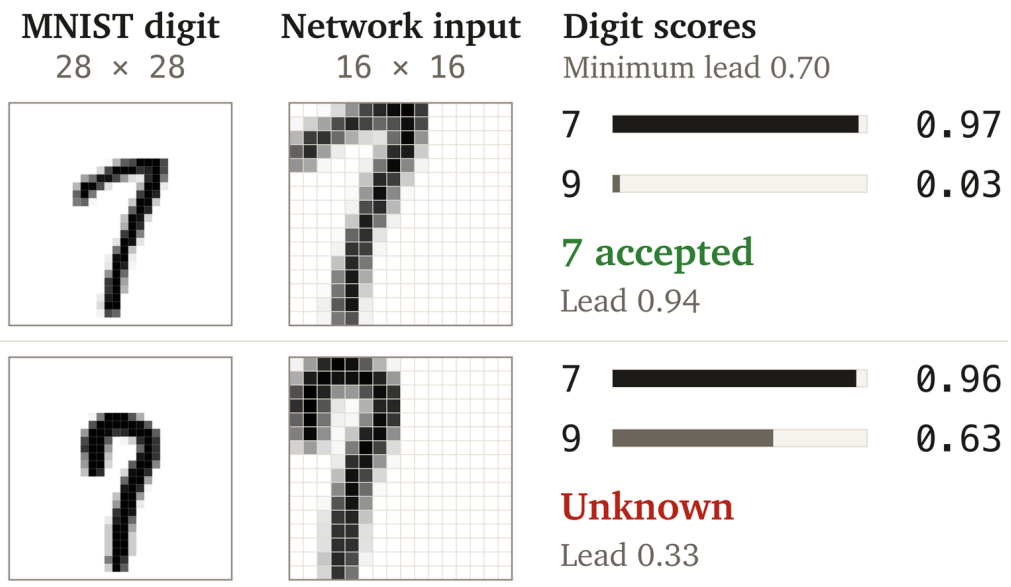
Video: https://media.glthr.com/end-of-ages-slate-neural-network/eoa-slate-zorro-live-2.mp4

The Zorro eraser, switched back on in the running game, at its default sensitivity of 0.30: one wide Z across the stone clears the drawing on release.

The Same Design, Reading Digits

Could the slate reader tell a handwritten 7 from a 9? I kept Cyan's design and trained a new set of weights on 60,000 handwritten digits from MNIST. I shrank each digit to fit the slate's 16 × 16 grid. Then I tried the reader on the 10,000 digits in the test set, none of which it had seen during training.

On about 96% of the test digits, its first choice was right. I then applied the game's rule: the best score has to beat the next one by at least 0.70. The reader gave an answer for 86% of the digits. Just over 99% of those answers were right. Here is what happened to two sevens.



The digit reader you can try above gives both drawings a high score for 7. The second drawing also scores 0.63 for 9, close enough to make the reader answer "Unknown".

You can try your own handwriting by switching the first figure to Digits 0–9. Try leaving the slate blank, too. With the extra S-curve that Cyan used on the inputs, no digit scored above 0.4 on a blank grid. When I trained the network without that curve, it read more digits correctly, but it also looked at an empty square and called it a 5. I would need Cyan's training drawings to see whether the same trade-off held for the Bahro signs.

Technical Details

The boxes below give the layout of the files, the addresses in the program, the equations and the scores behind each section. They were checked against the four `dat/*.tron` files, the game program `eoa.exe`, which the Mac edition runs inside its own copy of Wine, and the game's Python scripts, decrypted from `python.pak`. The program keeps no names for its functions: each is named here by what it does and by its address.

The symbols and their internal names

The four networks are `laki.tron` (Laki'ahn), `srln.tron` (Noloben), `tdlm.tron` (Todelmer) and `thgr.tron` (Tahgira). libHSPlasma, the community's library for the files of Plasma, does not read `.tron` files.

Name used in this article	Output of the network
Pedestal 1, 2 or 3 of an Age	`Laki01`, `Laki02`, `Laki03`; `Srln01`, `Srln02`; `Tdlm01`, `Tdlm02`; `Thgr01`, `Thgr02`, `Thgr03`
Keep, one for each Age	`LakiKeep`, `SrlnKeep`, `TdlmKeep`, `ThgrKeep`
Wind (Laki'ahn), Rain (Noloben), Time (Todelmer), Heat (Tahgira)	`LakiWind`, `SrlnRain`, `TdlmTime`, `ThgrHeat`
Snake, Sing, Friendship, Torture	`xSnake`, `xSing`, `xFriendship`, `xTorture`
Take, Direbo	`Take`, `Direbo`

The pictures of the symbols come from the game's textures: `xpedsymb-*` in `Direbo_Textures.prp` (the pedestals, Keep, Take and Direbo), `lakiwindsymbolcarving`, `srlnraindropsymbol`, `tdlmtimesymbol` and `thgrheatventsymbolbright_slate` in the textures of each Age, and `xfriendshipsymbol`, `xsingsymbol` and the journal sketch `xjournalsketch_painsymbol` in `GUI_BkBookImages.prp`. No texture shows the Snake.

The layout of the network files

Each file holds one perceptron of three layers (256 → 268 → 10 or 11): a matrix of 256 × 268 weights, then one of 268 × 10 or 268 × 11. The function that loads a network (`eoa.exe 0x6e36e0`) reads each file to its last byte: 290,064 bytes for `laki.tron` and `thgr.tron`, 288,979 bytes for `srln.tron` and `tdlm.tron`.

- **Header (9 bytes):** format version `6`; four flags `(1, 1, 1, 0)`, stored at `+0x44..+0x47`, for `antiAlias`, `scaleToFit`, `keepAspect` and `center`; then `n_in = 256` as a `uint32`.
- **Input layer (256 neurons):** each neuron has an empty name (with a `uint16` length), an activation type `2`, and 268 outgoing weights as little-endian `float32`: the matrix `W1`, 68,608 numbers.
- **Output layer (10 or 11 named neurons):** Laki'ahn and Tahgira have 11 outputs, with three pedestals; Noloben and Todelmer have 10, with two. The difference of 1,085 bytes between the files is that neuron: 268 weights of 4 bytes, and 13 bytes for its header and its name.
- **Hidden layer (268 neurons):** `n_hidden_layers = 1` and `n_hid = 268` as `uint32`, then each neuron with an empty name, an activation type `2`, and 10 or 11 outgoing weights, the matrix `W2`.

The format has no bias neuron.

How a drawing becomes 256 cells

The function that turns the strokes into the grid (`eoa.exe 0x6e63e0`) computes the box `(minX, minY, maxX, maxY)` of all the strokes, and uses `span = max(maxX - minX, maxY - minY)` on both axes, since `scaleToFit` and `keepAspect` are set. `center` is `0`, and the centering branch at `0x6e65e0` runs only when `center && !scaleToFit`. Every point is therefore `(x - minX) / span` and `(y - minY) / span`: a tall stroke starts at column 0, and a flat one at row 0.

Each segment goes to the function that draws a line (`0x6ed180`), which multiplies the coordinates by 15, steps along the longer axis by 1.0, rounds `(cx, cy)` to `(ix, iy) = (floor(cx + 0.5), floor(cy + 0.5))`, and, through `0x6ed570` (`antiAlias = 1`), sets the cell to the larger of its value and `max(0, 1 - 2 * ((cx - ix)^2 + (cy - iy)^2))`.

The sigmoid on the input layer

Every neuron of the four `.tron` files carries the same activation type, `2`: in the output layer, in the hidden layer and in the input layer. It is the logistic sigmoid, `1 / (1 + exp(-x))`, that the hidden and output layers need. The function behind `recognize()` (`eoa.exe 0x6e2e70`) applies it at each layer:

1. At `0x6e3064`, each of the 256 rendered values, between 0 and 1, is copied into the value of an input neuron.
2. At `0x6e3089`, the input layer is fed forward (`0x6e5470`), one neuron at a time (`0x6e4960`).
3. An input neuron has no incoming connection, so the weighted sum is skipped, and the neuron's activation (`0x6e4930`) is applied to the value of the cell itself.

The whole calculation, for a drawing x, with σ the sigmoid:

$$a_0 = \sigma(x), \qquad a_1 = \sigma(a_0 W_1), \qquad y = \sigma(a_1 W_2)$$

With the weights as shipped and the sigmoid removed from the input layer, an empty slate gives `xFriendship` a score of 0.305 in `tdlm.tron`: all 268 hidden neurons then sit at 0.5. With the sigmoid, an empty slate and a slate filled with ink score between 0.001 and 0.036, in the four Ages.

The two acceptance rules in the script

`RecognizeSketch()` is in `xBahroSlateGUI.py` (line 489), in the decrypted `python.pak`. `GetScribblePercent()` is in the program (`eoa.exe 0x6e36a0`).

1. **The scribble rule:** `scribblePercent = count(pixel != 0.0) / 256.0`. A value of `0.4` or more (103 cells) returns `'Scribble'` at once.
2. **The margin rule:** the results are filtered to the symbols of the Age and to `('xFriendship', 'xSing', 'xSnake', 'xTorture', 'Take', 'Direbo')`. The script then requires `bestMatch[1] - nextBestMatch[1] >= 0.7`, and returns `'Unknown'` otherwise.

How the 42 pictures were recovered

Each of the 42 pictures is one output neuron of `laki.tron`, `srln.tron`, `tdlm.tron` or `thgr.tron`, inverted by gradient ascent from the weights, with the forward calculation of the game, sigmoid on the inputs included. `xSnake` sits against the left edge and `LakiWind` against the top edge. `Take`, `xSnake`, `xSing`, `xFriendship`, `xTorture` and `Direbo` have the same shape and position in the four files.

The scores of the simple strokes

Shapes of one stroke, through the rasterizer and the weights of the game:

Drawing	Network	Best output	Runner-up	Margin	Lit cells
Straight vertical line	`laki.tron`	`Laki02` (0.998)	`xSing` (0.002)	0.996	6.2%
Plain `V`	`laki.tron`	`xTorture` (0.908)	`xSing` (0.001)	0.906	12.1%
Plain circle	`srln.tron`	`Take` (0.995)	`SrlnRain` (0.003)	0.992	18.0%
Plain circle	`tdlm.tron`	`TdlmKeep` (1.000)	`Take` (0.034)	0.966	18.0%
Plain square	`srln.tron`	`Take` (1.000)	`Srln01` (0.056)	0.944	23.4%
Plain square	`tdlm.tron`	`xSnake` (0.994)	`TdlmKeep` (0.267)	0.728	23.4%
Zigzag of three bars (`E`)	`tdlm.tron`	`xSnake` (1.000)	`Tdlm01` (0.001)	0.999	14.5%

The plain circle also scores 0.989 for `Take` in `laki.tron`.

In the clips, the game itself answered. `laki.tron` gave the Snake of the first clip 0.995 for `xSnake`, against 0.004 for the runner-up, and the vertical line 0.960 for `Laki02`, against 0.002. The cursive "glthr" scored 0.983 for `LakiWind` and 0.961 for `Laki01`: a margin of 0.022, hence `Unknown`.

The plain circle of Todelmer is a borderline case. In one test, `tdlm.tron` gave it 0.999 for `TdlmKeep`, against 0.284 for `Take`. Drawn again eight times, in two sizes, a plain circle

kept `TdlmKeep` above 0.99, but `Take` rose to between 0.53 and 0.82, and the game answered `Unknown`.

> **The switched-off eraser in the code**
>
> The control of the slate has a "Zorro guesture" (so spelled), with `enableZorro` and `setZorroSensitivity`. The shipped `xBahroSlateGUI.py` calls `enableZorro(0)`. Calling `sc.enableZorro(1)` on the control with tag `100` of `lakiSlateGUI`, after `PickupSlate()` has set it back to `0`, brings back the four stages of the recognizer in the program (`eoa.exe 0x709307` to `0x70a6b8`). A wide `Z` across the stone, with `u` between 0.08 and 0.92, fires `kSketchErased` when the mouse button is released, and wipes the drawing and the `Z`. The default sensitivity is 0.30.

The scores of the chameleons, the dots and the doodles

The same stroke through the four files:

Stroke	`laki.tron`	`srln.tron`	`tdlm.tron`	`thgr.tron`
`Γ` corner of two segments (31 lit cells, 12.1%)	`Laki02` (0.985)	`Take` (0.959)	`xSnake` (0.905)	`Thgr03` (0.860)
`N` in one stroke (18.4%)	`Laki02` (0.999)	`SrlnRain` (0.961)	`Tdlm02` (0.961)	`Thgr03` (1.000)
Plain figure 8 (19.1%)	`xSnake`	`xSnake`	`xSnake`	`xSnake`

The `N` gives four outputs that belong to one Age each, with every margin at 0.958 or more. The figure 8 scores between 0.965 and 0.999 for `xSnake`, with every margin at 0.961 or more.

The least ink:

- **Three cells.** None of the 11,054,080 patterns of three cells passes, in any of the four networks. The best margin is 0.557, in Noloben.
- **Four cells and more.** All 42 symbols can be triggered with 4 to 14 cells (1.6% to 5.5% of the slate; the median is 7). `tdlm:xTorture` (0.829, margin 0.824) and `srln:xTorture` (0.743, margin 0.741) fire from four dots.
- **Straight lines.** 41 of the 42 outputs can be triggered with two segments or fewer (`ThgrHeat` needs an arc), and 8 with a single line. In Tahgira, a line tilted 13.5° downward scores `xFriendship` (0.915, margin 0.909), and a line tilted 9.5° upward scores `xTorture` (0.994, margin 0.973). At 8° upward, three Ages score `xTorture`.

The doodles and the words:

- Beamed notes (♫) score 0.998 for `xTorture` in Noloben. With one dot beside them, Laki'ahn goes from `xTorture` to `xFriendship` (0.999, margin 0.99).
- The umbrella with one raindrop scores `SrlnRain` (0.958, margin 0.90).
- `"AI"` scores 0.987 for `xTorture` in Todelmer, and `"<3"` 0.997 in Laki'ahn. `"YESHA"` reads as `xFriendship` in three of the four Ages, and `"CYAN"` scores `xSing` (0.92) in Noloben.
- The cursive `"glthr"` reads as `xFriendship` in Todelmer (margin 0.91) and in Tahgira (margin 0.83). With a period, `"glthr."`, the tie of Laki'ahn breaks into `LakiWind` (0.992, margin 0.92).

The scores measured in the running game

For these checks, each stroke was projected through the camera of the slate of its Age (`dv/du` between 1.14 and 1.17 on the texture of 512 × 512), drawn on the slate in the running game, and the answer of `perceptron.recognize()` was read back. This rules out an error of my own copy of the calculation.

Test in the running game (1024 × 768)	Age	Answer of the game	Best output	Runner-up	Margin
`N` of five points	Laki'ahn	`Laki02`	0.987	0.002	0.985
`N` of five points	Noloben	`SrlnRain`	0.716	0.002	0.714
`N` of five points	Todelmer	`Tdlm02`	0.899	0.002	0.897
`N` of five points	Tah-gira	`Thgr03`	0.989	0.039	0.950
Umbrella, no raindrop	Noloben	`Unknown`	0.372 (`Take`)	0.090 (`Srln02`)	0.282
Umbrella and one raindrop	Noloben	`SrlnRain`	0.965	0.049	0.917
`"YESHA"`	Tah-gira	`xFriendship`	0.997	0.090	0.906
`"CYAN"`	Todelmer	`Unknown`	0.998 (`xFriendship`)	0.997 (`xSing`)	0.001
Straight line, 13.5° downward	Tah-gira	`xFriendship`	0.744	0.003	0.741
Straight line, 9.5° upward	Tah-gira	`xTorture`	0.886	0.023	0.863
Compact cursive `"glthr"`, then `"glthr."`	Laki'ahn	`LakiWind`, then `Unknown`	0.768, then 0.413	0.041, then 0.003	0.727, then 0.410

In the umbrella clip, the bare umbrella returns `Unknown` (`SrlnRain` at 0.012), and the raindrop, one dot at `(681, 155)`, takes `SrlnRain` to 0.965.

A Bahro comes only for a slate that lies off every pedestal, and only once the player has walked far enough from it: the game waits for the player to leave an area around the slate (`RunBahroLinkIn()` in `xBahroHandler.py`, in the decrypted `python.pak`). In my tests, no Bahro came for a scribble. It then plays `BahroLinkInOutFriendly` for Friendship, `BahroLinkInOutTortured` for Torture, `BahroLinkInScaredOut` for the Snake, `BahroSing01` for Sing and `BahroLinkInOutConfused` for `Unknown`.

The four versions of the handwritten-digit reader

The digit reader has 256 inputs, 268 hidden detectors, 10 outputs and about 71,000 weights. With Cyan's design it uses no biases. The weights take less than 300 KB on disk, and a reading takes a fraction of a millisecond. The table compares all four versions on the same 10,000 MNIST test digits.

Variant	Right answers	Accepted (margin of 0.70)	Right among the accepted	Best score on a blank square
As in the game: no bias, sigmoid on the inputs	93.80%	80.16%	99.04%	0.399
The same, with the corner step of `Pattern::Render`	95.67%	86.02%	99.28%	0.211

Variant	Right answers	Accepted (margin of 0.70)	Right among the accepted	Best score on a blank square
No bias, no sigmoid on the inputs	98.40%	96.03%	99.57%	0.896 (digit 5)
Standard network: biases, no sigmoid on the inputs	98.50%	96.18%	99.60%	0.974 (digit 5)

The second row can be written as a `.tron` file and read by the game's own code. Without the sigmoid on the inputs, the weights are free of a contrast squeezed to 0.231, and a blank square is read as a 5.

The design has limits that the game works around. It scales a drawing by the box around all of its ink, so one stray dot in a corner shrinks the symbol and moves every cell. It keeps no trace of the order or the direction of the strokes: a circle drawn clockwise and one drawn the other way are the same picture. And its outputs are independent sigmoids, trained on valid symbols only, with no output for "other": hence the scribble cutoff and the margin rule, written by hand.

Conclusion

When you find the sign for the Snake in Laki'ahn, you still have to draw it. Your snake will bend in different places from mine. A trained recognizer gives us room to draw the same sign in our own hands. It also gives us room to be misunderstood. On the beach in Noloben, I could draw my umbrella well enough for another person to recognize it and still have the Bahro walk away.

Was that impressive in 2005? Sixteen years earlier, Yann LeCun and his colleagues had already trained a neural network to read handwritten digits from US ZIP codes.[3] What impresses me is the job Cyan gave it. The designers decided what the Bahro would do for each sign. They left it to the network to decide whether the marks I made with a mouse looked enough like one. All I changed on my second try was to put a drop beside the umbrella. This time the Bahro raised its arms and it rained.

Series and Companion Articles

Myst V: End of Ages (2005): The End of Ages Graph & Behind the Console

Group	Article	Address
Graph	The End of Ages Graph, 1: The 2005 Game as a Graph	glthr.com/end-of-ages-graph-1
Graph	The End of Ages Graph, 2: A Way In, No Way Back	glthr.com/end-of-ages-graph-2
Finding	Commands Beyond the Slate: Myst V: End of Ages (2005)	glthr.com/end-of-ages-commands-beyond-the-slate
Finding	**Inside Myst V's 2005 Neural Network** (this article)	glthr.com/end-of-ages-slate-neural-network
Finding	Ghosts of Uru	glthr.com/end-of-ages-hidden-worlds

[3] Yann LeCun and colleagues, "Handwritten Digit Recognition with a Back-Propagation Network", 1989. The team tested its reader on handwritten digits supplied by the US Postal Service.

The 5 Myst Studies on glthr.com (glthr.com/myst-graphs)

Game	Graph Study & Browser	Findings
Myst (1993)	The Myst Graph (3) · Browser	The Library Ceiling Shortcut, Behind the Ages, Trapped in Myst
Riven (1997)	The Riven Graph (2) · Browser	The Riven Diffs, The Contradictions of the Fifth Age, Catherine Is Still in Her Cell, The Sixth Knock, What Changed Between Riven's CDs and DVD
Myst III: Exile (2001)	The Exile Graph (2) · Browser	Linking Without a Book, The Valve That Refuses to Turn, The Two Realities of the Edanna Tusk
Myst IV: Revelation (2004)	The Revelation Graph (2) · Browser	The Vision the Amulet Never Showed, 53,921 Files: Watching Ubisoft Montreal Build Myst IV
Myst V: End of Ages (2005)	The End of Ages Graph (2) · Browser	Commands Beyond the Slate, Inside Myst V's 2005 Neural Network, Ghosts of Uru

How to cite this article

> Guillaume Lethuillier, "Inside Myst V's 2005 Neural Network", glthr.com, November 22, 2026.
> https://glthr.com/end-of-ages-slate-neural-network

For BibTeX:

```
@misc{lethuillier2026inside,
  author = {Lethuillier, Guillaume},
  title  = {Inside Myst V's 2005 Neural Network},
  year   = {2026},
  month  = nov,
  url    = {https://glthr.com/end-of-ages-slate-neural-network},
  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/end-of-ages-slate-neural-network, which remains the reference version.