PROJECT SUMMARY · OCTOBER 2026 · ROM BUILDER V0.1

Viking Child LX for the Atari Lynx

The Viking at twice the speed – same jumps, same levels, same enemies



The title screen of the original game. Viking Child LX changes nothing about the look of the game; only the Viking moves faster.

At a glance

Original	Viking Child (also known as The Prophecy), Imagitec Design, published by Atari Corp. for the Lynx in 1991
Credits	Design Nigel Cook, programming David Chiles, graphics P. Goldsmith and Nigel Cook, music Barry Leitch, producer Juli Wade
Target	Atari Lynx, 65C02 at 4 MHz, 64 KB RAM, 160 × 102 pixels, 256 KB cartridge
Change	The Viking runs two original game steps per step: 60 instead of 30 pixels per second
Unchanged	Graphics, levels, enemies, timers, energy clock, sound, music, frame rate
Modes	FAST (default) and CLASSIC, switched in the pause with A or B; CLASSIC is the original game
New code	119 bytes in the zero page, a 26-byte loader and four 3-byte jumps
ROM Builder	Windows tool and Linux command line, v0.1: builds Viking Child LX from the user's own cartridge image
Development	23–25 October 2026, by Claude in Claude Code, directed and checked by the project's author

Overview

Viking Child is one of the best-looking games on the Atari Lynx, but many players find it too slow. The project began with a question: is the game slow because it was badly programmed, or by design? The analysis answered it: the frame rate is steady and fine; the Viking himself moves only one pixel per game step. The author then decided that only the Viking should get faster – not the whole game, and not just the picture.

Viking Child LX runs the Viking's control routine twice per game step. Each run is exactly one step of the original, so every walk, jump and fall follows the original path – in half the time. Everything else keeps its original pace. The pause switches between FAST and CLASSIC, and CLASSIC is identical to the original game.

The **ROM Builder**, a single Windows program (and a command-line version for Linux), makes Viking Child LX from the user's own cartridge image. It contains only the new code and the steps to put it into the game, no game data.

AI vs. human: who did what

This project was made with AI, and that is no secret. Claude, Anthropic's AI model, running in Claude Code on the author's Windows PC, did the analysis and wrote every line of code and text. The decisions were the author's.

The author

- Picked the game after reading that many players find it too slow, and asked why.
- Decided what to build: only the Viking faster, switchable – not a higher frame rate, not a faster game.
- Added what he knew: the Amiga and Atari ST versions play faster.
- Asked for the comparison video, the forum texts, this summary, the download page and the builder, chose the name LX and checked the results.

Claude (AI)

- Unpacked the cartridge, disassembled the game and measured every game step in a tracing emulator built from Felix.
- Found why the game is slow and where the Viking's movement lives, and proposed how to double it without changing the jump paths.
- Wrote the 6502 code, a packer for the game's compression, the build, the tests, the video, the builder and all texts.
- Found and fixed its own mistakes with the tests, for example a broken crouch picture after moving a file.

> **In numbers:** from 23 to 25 October the author wrote 11 prompts for this project, about 200 words in all.

Why the original is slow

Measured in the emulator on level 1, about 500 game steps while walking and fighting:

Measurement	Value
Display refresh in the levels (the menus run at 50 Hz)	60 Hz
Game steps per second (494 of 498 steps take 2 refreshes, 4 take 3)	29.8
Work per step until the picture is ready (median; maximum 2.24)	1.62 refreshes
– background: 12 tile rows, one sprite chain per row	0.67
– sprites and status display (84 separate sprite jobs for the score text)	0.56
– game logic	≈ 0.4
Viking walking speed (1 pixel per step; 2 with the SPEED item)	30 px/s
Jump: 20 steps up at 1 pixel, then falling at 1 pixel per step	≈ 1.4 s
Energy clock: 8 points every 1,440 steps, walking or standing	48 s

So the game is not badly programmed: the picture is ready in time, and the frame rate hardly ever drops. The pace comes from the movement itself. The walking speed is a variable, and the SPEED item doubles it, so the engine could move the Viking faster. But rising and falling are always 1 pixel per step: the game moves the Viking, takes

the whole move back on a collision, and finds the ground only at an exact height (y mod 8 = 6). Bigger vertical steps would skip that height. The jump has no acceleration – 20 identical entries in a table – which makes it feel floaty. Since the Amiga and Atari ST versions play faster, the slowness is a property of the Lynx version: it runs 30 steps per second and kept the movement of 1 pixel per step.

How LX works

Two original steps in one

In FAST mode the Viking's control routine – input, walking, jumping, attacking, collision and scrolling – runs twice per game step. Each run is exactly one original step with 1-pixel moves, so the ground check, the collision and the jump table work as before, and every jump keeps its height and width. A counter that lets the walking animation advance every other step is flipped between the two runs, so the animation advances once per step. Pushes from outside, such as being knocked back by an enemy, still happen once per step.

Room for new code

The game uses nearly all of the Lynx's 64 KB. The new code – 119 bytes – lives in the upper half of the zero page, which the game never touches (checked in all disassembled code and all recorded runs). A 26-byte loader in a gap behind the game program loads it once at power-on from a free entry of the cartridge's file directory.

A packed game

The game program is stored with byte-pair compression. To change four jumps, LX unpacks the two program files, patches them and packs them again with its own packer, which compresses as well as the original. The files grow by 28 and 3 bytes, so two small neighbouring files move into free space on the cartridge, and the directory is updated.

A table that bypasses the directory

The first try moved a different file – one of the Viking's 23 animation pictures. The game loads those through its own table of fixed cartridge positions, not through the directory, and the crouching Viking turned into a stripe of garbage. The step-by-step comparison with the original found it; now only files that the game loads through the directory are moved.

The switch in the pause

The pause shows VIKING CLASSIC FAST below GAME PAUSED. Erasing text would need a sprite outside the zero page (a sprite description in page 0 stops the Lynx's sprite engine), so both words are always drawn and only their colours change – the new text covers the old one exactly.

The cartridge checksum

The Lynx boot code checks a checksum over the whole cartridge, stored in the encrypted first boot stage. LX computes the new checksum and encrypts the boot stage again with the well-known Lynx boot key, as for any homebrew cartridge.

Testing

- **CLASSIC = original:** the complete game RAM compared after every game step with the original, with the same input: 1,570 steps in level 1 and 1,183 steps in the dragon fight – identical. Only the state of the sound driver is sometimes sampled at another moment, because loading takes a different time; all 69 sound requests come at the same step and frame.

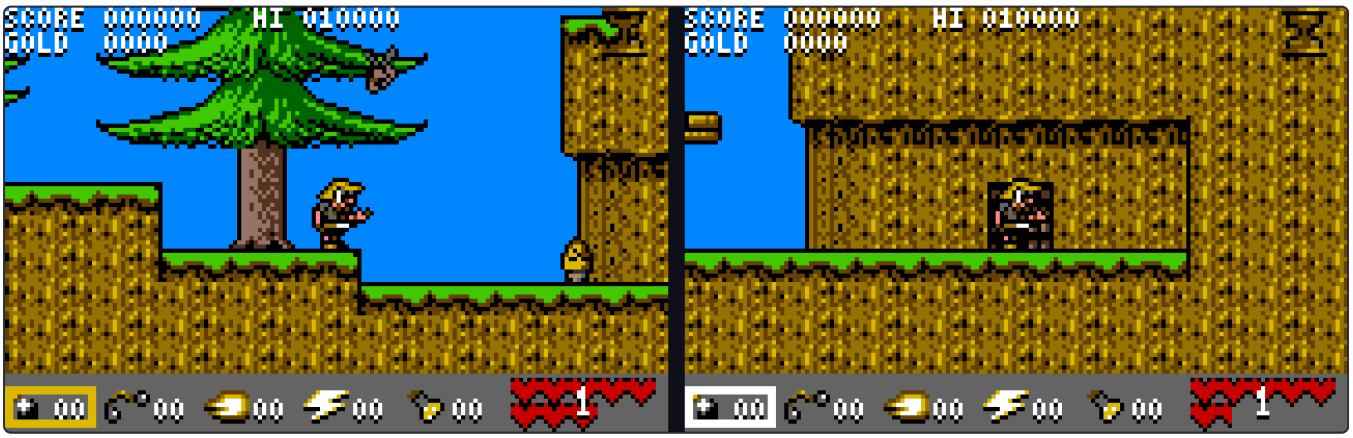
- **FAST = the same path:** the Viking's position after LX step j equals the original after step 2j+1, for walking, jumping and falling, until an enemy pushes him.
- **Endurance:** 20,000 frames with random input, pauses, mode switches, Game Over and a new game; the new code stayed intact.
- **Stack:** the Viking's animation picture is kept in the stack area of the original game; CLASSIC uses exactly the original stack depth, FAST two bytes more, still 20 bytes away.
- **Builder:** the same build in Python, in the Windows program and in the Linux file, identical to the reference build; changed, damaged, already patched and unrelated files are refused.

> Not tested yet: the later levels and real hardware. The changed code is the same in every level, but feedback from players is welcome.

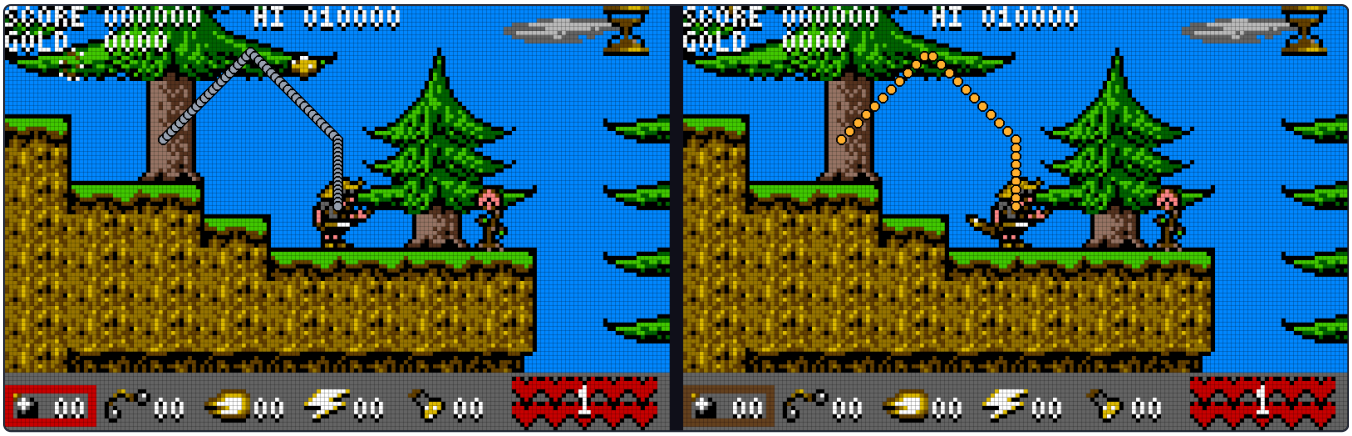
Tools

lynxtrace	A tracing emulator built from the Felix emulator's core: runs without a window, follows every instruction, records RAM accesses, RAM snapshots per game step, sprite jobs, video and sound.
Disassembler	Recursive 65C02 disassembler that uses the instructions the emulator actually executed as entry points.
Cartridge tools	Boot decryption and checksum, the file directory, unpacker and packer for the game's compression.
Measurements	Time per game step, time per routine of the main loop, sprite engine time per job, speed and path of the Viking.
Tests	Step-by-step comparison with the original, path comparison of FAST with the original, endurance run, builder test matrix.
Video	A simple autopilot for the Viking, recordings of both versions and a renderer for the comparison video.
ROM Builder	Windows program and Linux command line, checked for game data before release.

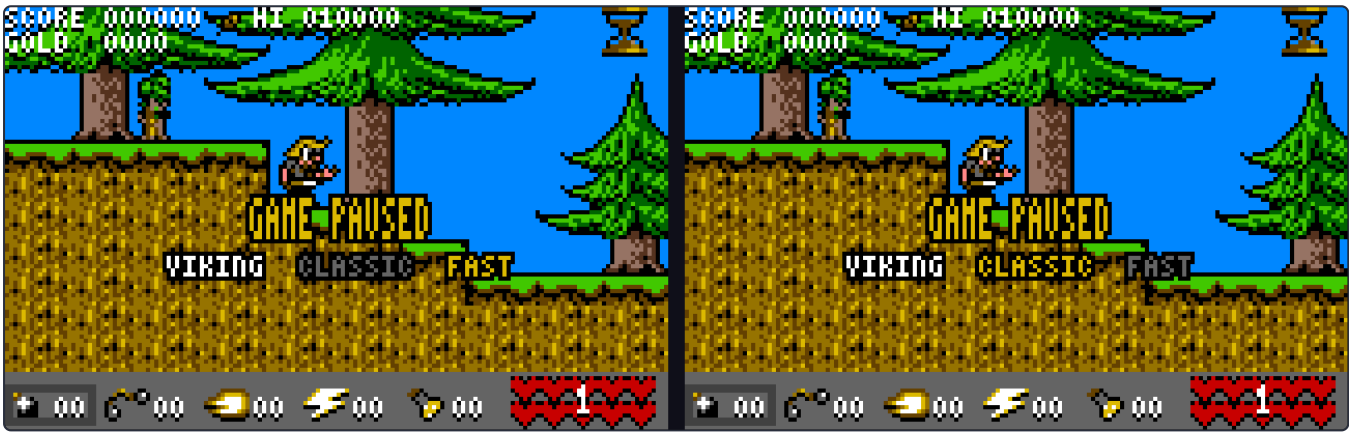
Viking Child LX on the Lynx



The first 20 seconds of level 1 with the same simple autopilot: the original (left) gets 470 pixels far, LX (right) 823 – stops at walls and hits by enemies take the same time in both.



The same jump with one dot for every game step: the original needs 58 steps (1.9 seconds), LX 29 (1.0 second) on exactly the same arc.



The pause: FAST selected (left) and CLASSIC selected (right). A or B switches.



Unchanged: the forest at the start of level 1, the caves and the dragon at the end of the first level.

2×	30	2,753	1.8×	0
walking and jumping speed of the Viking	game steps per second, as in the original	game steps compared: CLASSIC identical to the original	distance after 20 s with the same autopilot	bytes of game data in the ROM Builder

Viking Child © 1991 Imagitec Design, published by Atari Corp. Design: Nigel Cook. Programming: David Chiles. Graphics: P. Goldsmith and Nigel Cook. Music: Barry Leitch. Producer: Juli Wade. Viking Child LX: unofficial, non-commercial fan project, developed with Claude Code; not affiliated with or endorsed by the rights holders. Use the ROM Builder only with your own copy of the game and do not share the built .lnx file. Screenshots: recorded in the Felix-based tracing emulator from the original game and from Viking Child LX.