NexusView: A File Graph Visualization Tool

Harshit Prashant Dhanwalkar

July 11, 2025

Introduction

NexusView is a desktop application developed in Rust using the `eframe` and `egui` crates. Its primary purpose is to visualize the relationships between files within a specified directory, presenting them as interactive graphs. This tool helps users understand the structure and connections within their file systems, particularly focusing on linked files and tagged content.

Key Features

- **File Scanning:** Recursively scans directories to identify various file types, including markdown, code, images, and PDFs.
- **Link Graph Visualization:** Creates a graph where nodes represent files and edges represent explicit links found within the file content (e.g., `[link text](path/to/file)` or `[[wiki link]]` in markdown files).
- **Tag Graph Visualization:** Extracts hashtags (e.g., `#tagname`) from text-based files and builds a separate graph showing relationships between tags and the files they appear in.
- **Interactive Physics Simulation:** Nodes in the graph are arranged using a force-directed layout algorithm, allowing for dynamic and intuitive exploration. Users can drag nodes, adjust physics parameters (spring constant, repulsion, damping), and reset the layout.
- **Content Preview:** When a file node is selected, its content is displayed in a side panel. This includes:
  - Syntax highlighting for code files.
  - Markdown rendering for `.md` files.
  - Basic information (page count, title, author) for PDF files.
  - Image display for common image formats.
- **Directory Navigation:** Provides a tree-view panel to easily navigate and select directories for scanning.
- **Search and Filtering:** Allows users to search for specific files or tags within the graph and filter tags.
- **Cross-Platform Compatibility:** Built with Rust and `eframe`, NexusView is designed to run on Linux, macOS, and Windows.

Technical Overview

The application leverages several Rust crates for its functionality:

- `eframe / egui`**:** For the graphical user interface.
- `petgraph`**:** For efficient graph data structures and algorithms.
- `regex`**:** For parsing links and tags from file content.

- `rayon`: For parallelizing computationally intensive tasks like physics simulation.
- `syntect`: For robust syntax highlighting of code files.
- `pdf`: For reading and extracting metadata from PDF documents.
- `image`: For loading and displaying image files.
- `egui-commonmark`: For rendering markdown content.

Usage

To run NexusView, compile the Rust project and execute the binary, optionally providing a directory path as a command-line argument:

```
cargo run -- <path_to_directory_to_scan>
```

If no path is provided, it will scan the current directory.