ma… 16 Branches 26 Tags Go to file Go to file Code ···

lethal-guitar Merge pull request #925 from lethal-guitar/add-missing-include ✓

f05996f · 6 months ago 2,638 Commits

.github	Fix Mac OS CI build	7 months ago
3rd_party	Upgrade Catch to version 3.4.0	6 months ago
benchmark	Update Google Benchmark	7 months ago
cmake	Fix Mac OS build	6 months ago
dist	Bump version	2 years ago
docker	Creating Shell script which runs clang-…	2 years ago
src	Fix Mac OS build	6 months ago
test	Upgrade Catch to version 3.4.0	6 months ago
tools	Creating Shell script which runs clang-…	2 years ago
.appveyor.yml	Clean up CI config	3 years ago
.clang-format	Format code with clang-format	3 years ago
.clang-tidy	Update .clang-tidy file	6 years ago
.dockerignore	Extend .dockerignore	3 years ago
.gitignore	Add .vscode to .gitignore	4 years ago
.gitmodules	Integrate Dear ImGui library	5 years ago
AUTHORS.md	Update AUTHORS.md	last year
BUILDING.md	Update BUILDING.md	2 years ago
CMakeLists.txt	Bump version	2 years ago
CONTRIBUTING.md	Update CONTRIBUTING.md	2 years ago
LICENSE.md	Change License to GPL v2	8 years ago
README.md	Clarify state of the project	last year

README GPL-2.0 license

Rigel Engine build passing CI passing Discord 11 online downloads 7k

What is Rigel Engine?

This project is a re-implementation of the game Duke Nukem II, originally released by Apogee Software in 1993 for MS-DOS. RigelEngine works as a drop-in replacement for the original executable: It reads the game's data files and plays just like the original, but runs natively on modern operating systems and is written in modern C++ code.

Compared to running the original in an emulator, RigelEngine offers many modern enhancements and usability improvements, including:

- wide-screen mode
- smooth scrolling & movement mode with increased frame rate
- quick saving
- improved game controller support
- extended modding features

About

A modern re-implementation of the classic DOS game Duke Nukem II

#game #opengl #cross-platform #game-engine #cpp #sdl2 #reverse-engineering #modern-cpp #entity-component-system #game-2d #cpp17 #retrogaming #game-engine-2d

Readme

GPL-2.0 license

Activity

922 stars

32 watching

60 forks

Report repository

Releases 25

v0.9.1 (stable) Latest

on Dec 10, 2022

+ 24 releases

Sponsor this project

https://www.paypal.com/donate?hoste…

Packages

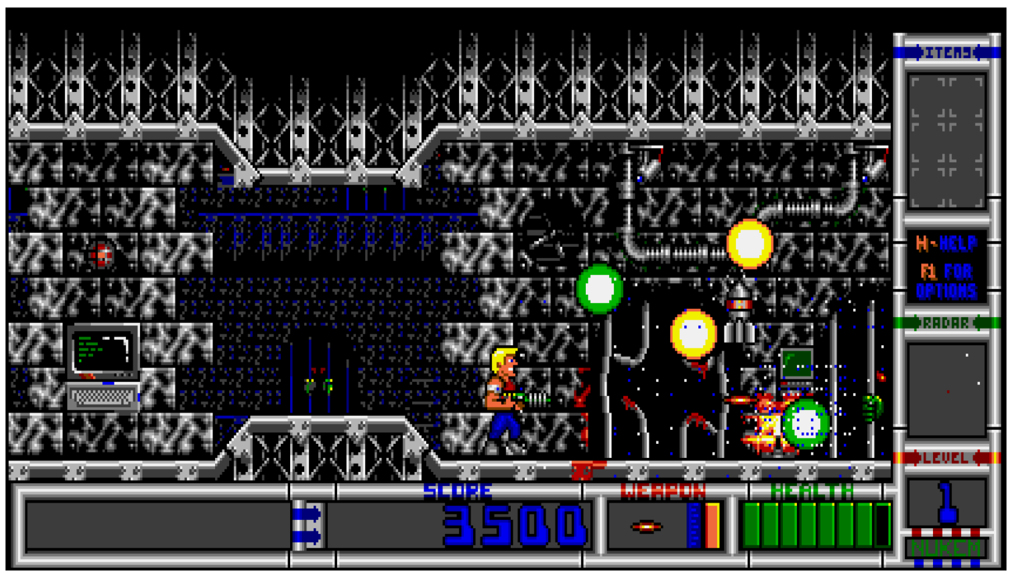
No packages published

Contributors 16

+ 2 contributors

Languages

● **C++** 80.0%	● **C** 17.5%
● **CMake** 2.4%	● **Other** 0.1%



There was never any source code released for the original game, so this project is based on reverse engineering (specifically, disassembly of the original binaries). See Duke2Reconstructed for a complete reconstruction of the original code.

Try the web version! (compiled to wasm via Emscripten)

Read the F.A.Q.

Watch a showcase video (outdated)

Current state

RigelEngine is stable and fairly complete in terms of feature set. All four episodes of the game (shareware and registered version) are fully playable and on par with the original game.

I currently don't have any plans for significant future development - consider the project to be in "maintenance" mode. This means I'll try my best to fix major problems/bugs that come up and keep CI running, but I won't add any new features.

Supported platforms

RigelEngine runs on Windows, Linux, and Mac OS X.

It's fairly easy to install on Steam Deck, too.

The Linux version also runs well on small single-board computers (SBCs) like the Raspberry Pi and Odroid Go Advance. See Running on Raspberry Pi and Odroid Go Advance.

There's also a (3rd party) PS Vita port.

System requirements

RigelEngine is not very demanding, but it does require OpenGL-capable graphics hardware. Either OpenGL 3.0 or OpenGL ES 2.0 can be chosen at compile time.

Any Nvidia or AMD graphics card from 2007 or later should run the game without problems. Intel integrated GPUs only added OpenGL 3 support in 2011, however. On Linux, using GL ES can be an option for those older Intel GPUs.

See Supported Graphics cards for more info.

Aside from the graphics card, you don't need much. The game needs less than 64 MB of RAM, and runs fine on a single-core ARMv6 CPU clocked at 700 MHz (Raspberry Pi 1).

Contributing

Contributions to RigelEngine are very welcome! Please have a look at the contribution guide before making a PR.

There is a growing body of documentation on the Wiki, to help with getting into the code base. A good place to start is Architecture Overview

If you are looking for some easy tasks to get started, take a look at issues labeled good first issue.

Running RigelEngine

In order to run RigelEngine, the game data from the original game is required. Both the shareware version and the registered version work. When launching RigelEngine for the first time, it will show a file browser UI and ask you to select the location of your Duke Nukem II installation. The chosen path will be stored in the game's user profile, so that you don't have to select it again next time.

It's also possible to pass the path to the game files as argument on the command line, which can be handy during development.

Acquiring the game data

You can download the shareware version for free from archive.org. The full version (aka registered version) can be bought on Zoom Platform, a store which still has a license to sell the game (unlike other digital stores like Steam and Gog).

If you already have a copy of the game, you can also point RigelEngine to that existing installation.

The only files actually required for RigelEngine are:

- `NUKEM2.CMP` (the main data file)
- `NUKEM2.F1`, `.F2`, `.F3` etc. up to `.F5` (intro movie files)

The intro movies aren't mandatory, RigelEngine simply skips movie playback if the files aren't found. They are still part of the experience though, so I do recommend including them when copying the game files somewhere.

If there are existing saved games, high score lists, or settings found in the game files, RigelEngine imports them into its own user profile when running for the first time.

Command line options, debugging tools, more info

You can find more info that didn't quite fit in this README over on the Wiki. For example, you'll find info on how to activate the built-in debugging tools, a list of bugs in the original version that have been fixed in Rigel Engine, etc.

Pre-built binaries

Pre-built binaries are provided with each Release. As of version 0.8.0, this includes Windows (x64), Mac OS (x64),