Sign in

oktailb / **BentoPack**

Code | Issues | Pull requests | Actions | Projects | Security and quality | Insights

BentoPack

Public

Sponsor | Notifications | Fork 0 | Star 0

main | Go to file | Code

oktailb Update README.md 9c6274e · 2 minutes ago

.github	fix (CI): invoke ctest di...	3 weeks ago
BentoPack	Update livewatchman...	last week
addons	feat (unreal): complet...	last week
benchmarks	fix (meshes): fix bug d...	last week
cmake	fix (CI): fix macos buil...	3 weeks ago
docs	chore (totolist): compl...	last week
examples	chore (filters): define fi...	2 weeks ago
palettes	chore (colorpalettes): ...	2 weeks ago
plugins	feat (filters): complete ...	last week
sample	chore (i18n): resynchr...	2 weeks ago
scratch	chore (colorpalettes): ...	2 weeks ago
scripts	chore (licensing): rem...	3 weeks ago
tests	Update test_live_watc...	last week
wrappers	feat (plugins): create s...	last month
.gitattributes	Initial commit	11 months ago
.gitignore	Update .gitignore	last week
ABOUT	Fill About and Licence ...	11 months ago
BentoPack.webp	chore (readme): convr...	last month

About

Little studio to manage sprites and animated textures

Readme

Apache-2.0 license

Activity

0 stars

0 watching

0 forks

Report repository

Releases

5 tags

Sponsor this project

patreon.com/Oktail

https://en.tipeee.com/lec...

Contributors 1

oktailb LECOQ Vincent

Languages

C++ 74.3% **C** 12.7%
C# 5.7% **GDScript** 3.6%
CMake 2.3% **Python** 1.4%

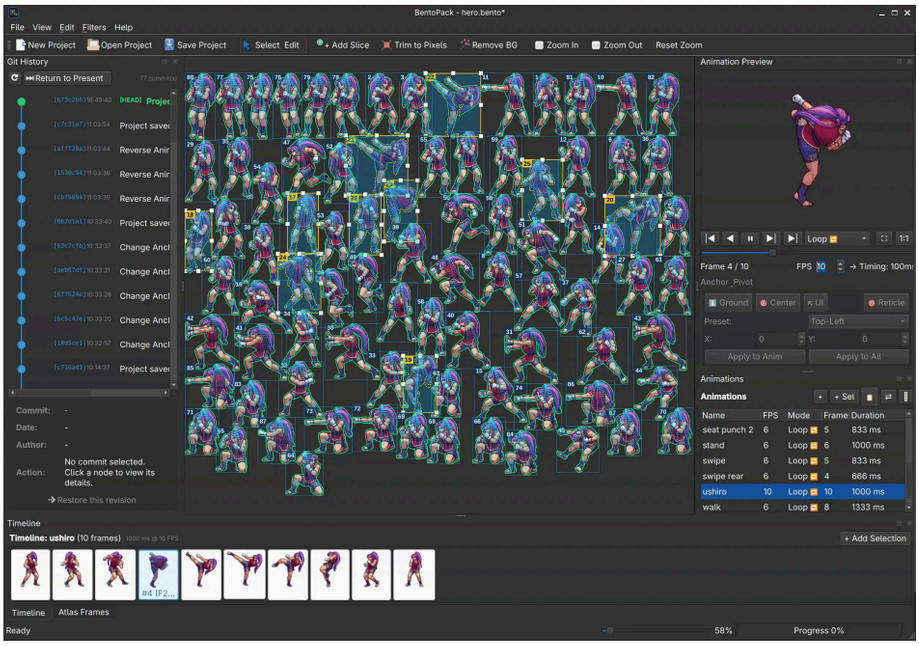
CMakeLists.txt	fix (CI): fix macos buil...	3 weeks ago
LICENSE	chore (license): clarify ...	3 weeks ago
README.md	Update README.md	2 minutes ago
ROADMAP.md	feat (filters): complete ...	last week
RemoveBackground....	chore (readme): convr...	last month
TODO.md	chore (totolist): compl...	last week
build.cmake	fix (ci): add qt6-tools-d...	last month
install.cmake	chore (colorpalettes): ...	2 weeks ago
package.cmake	fix (nsis): windows inst...	3 weeks ago
version.cmake	chore (cmake): cleanu...	last month

README Apache-2.0 license

BentoPack

CI passing | License Apache 2.0 | C++ 17 | Qt 6.5+ | CMake 3.20+

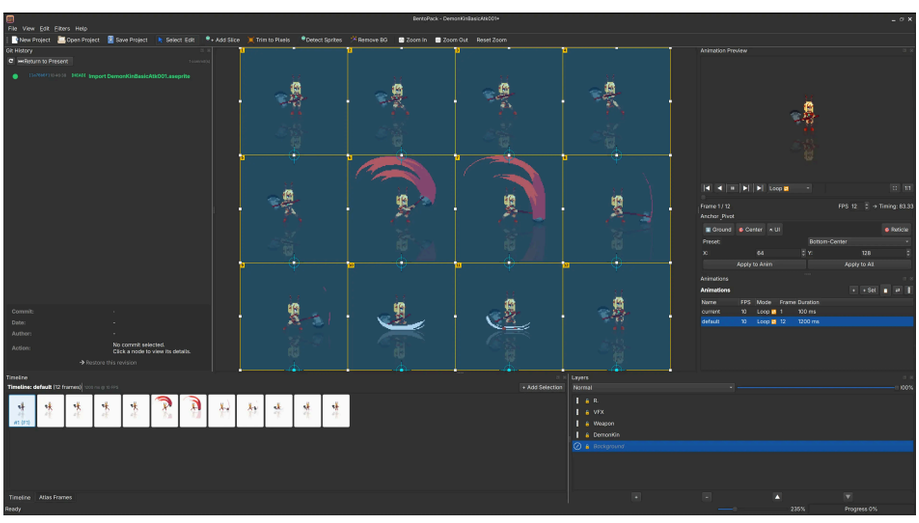
> **BentoPack** is a fast, modular, and modern desktop application tailored for game developers, pixel artists, and 2D animators to pack, slice, arrange, edit, and export high-density 2D sprite sheets, tight polygonal meshes, and animated textures. The entire project (UI, CLI, core engine, plugins, filters, and engine exporters) is 100% Free and Open Source under the **Apache License 2.0**. Official pre-compiled, auto-updating binaries are available for purchase on Steam and Itch.io for convenient 1-click desktop use.



Key Features

Intelligent Extraction & Codec Plugins

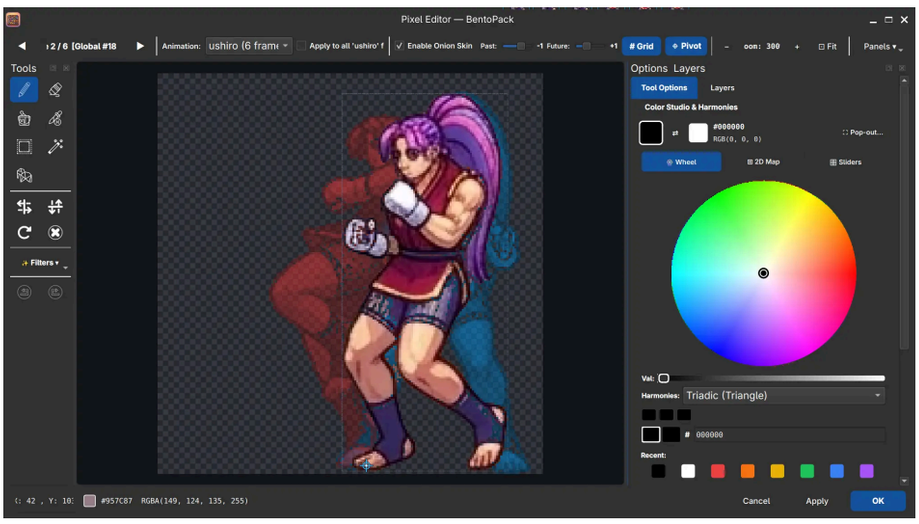
- **Native Aseprite (`.ase` / `.aseprite`) File Extraction:** Direct extraction of multi-layer, cel, frame, and animation tag definitions without requiring Aseprite installed. Automatically preserves blend modes and opacity.
- **Animated GIF Decompilation:** Instant extraction of multi-frame animated GIFs into ordered frame sequences.
- **Smart Sprite Sheet Slicing:** Automatic sprite detection and extraction with configurable alpha and color tolerance ($O(N)$ spatial grid).
- **Format Parsers & Importers:** Import existing sprite atlases with associated **JSON** (TexturePacker / Aseprite / Spine 2D / LibGDX) or **Godot 4** `.tres` metadata.



Surgical Pixel-by-Pixel Editor & Layer Studio

- **High-Precision Canvas (`Ctrl+E`):** Fast, non-interpolated pixel art editor (`SmoothPixmapTransform = false`) with configurable pixel grid (≥ 400) for surgical touch-ups without switching to GIMP or Photoshop.
- **Complete Multi-Layer Architecture (`LayerStackWidget`):** Non-destructive layer stack with layer creation, duplication, deletion, drag-and-drop reordering, visibility toggle, opacity slider (0–100%), and 12 blend modes (*Normal, Multiply, Screen, Overlay, Darken, Lighten, Color Dodge, Color Burn, Hard Light, Soft Light, Difference, Exclusion*), plus *Merge Down* and *Flatten Image*.
- **Onion Skinning:** Configurable forward and backward frame ghosting with custom opacity and tint for fluid animation inbetweening and sub-pixel alignment.
- **Continuous Bresenham Drawing:** 1px pencil and 1px eraser guarantee continuous unbroken lines even during fast mouse sweeps.
- **Instant Sampling & Flood Fill:** Eyedropper (`Alt+Click` or `I`) and 4-way flood fill bucket bounded by sprite limits and active selection.
- **Flexible Selections & Floating Stamp:** Rectangular marquee and magic wand color selection; Cut (`Ctrl+X`), Copy (`Ctrl+C`), and Paste (`Ctrl+V`) with draggable floating stamp preview.
- **Modular 16 Retro & Arcade Palettes (`.gpl`):** Standard GIMP/Aseprite palette system dynamically loaded from disk:
  - **Nintendo:** NES / Famicom (54), SNES / 16-bit (32), Game Boy DMG (4), Game Boy Pocket (4)
  - **Sega:** Master System 8-bit (64), Mega Drive / Genesis 16-bit (64), Saturn / 32-bit (64)
  - **SNK:** Neo Geo AES / MVS Arcade (64)
  - **Computers & Fantasy:** Commodore 64 (16), Amiga OCS (32), NEC PC-Engine (32), IBM CGA Modes 1 & 2 (4), PICO-8 (16), Endesga EDG 32 (32), and Bento Studio Standard (36).
  - **Custom Palettes:** Auto-scanning of user palettes from `%APPDATA%/BentoPack/palettes/` and local `palettes/` folder (`.gpl` , `.hex` , `.pal` , or image sampling).
- **Integrated CDT Mesh & Laser Knife:** Direct polygon vertex editing and interactive edge-cutting knife tool inside the pixel editor.

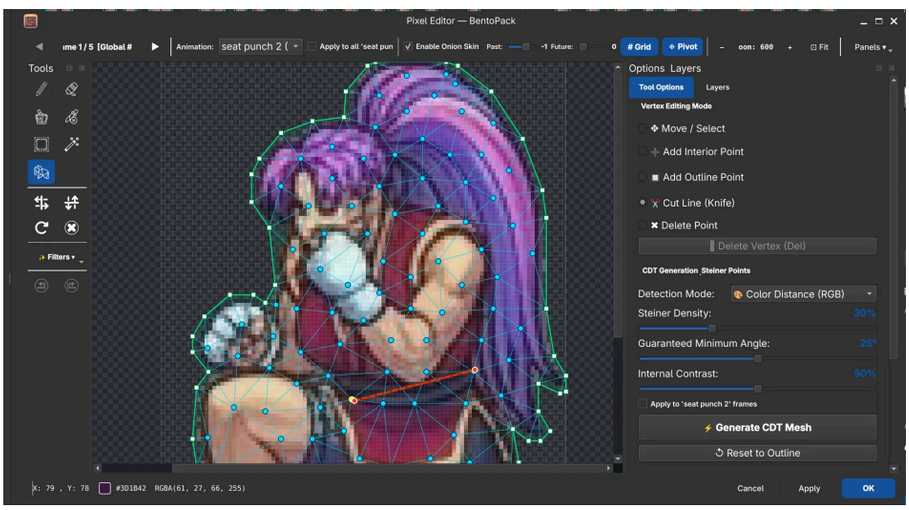
- **Tabbed Ergonomics:** Right column split into distinct tabs separating tool options from the layer stack for clean workflow.
- **Inter-Frame Navigation:** `[◀ Previous]` and `[Next ▶]` buttons (`Page Up` / `Page Down`) allow touching up entire animation cycles frame by frame in a single session.
- **Two-Tier Undo/Redo:** Local stack for fine brush strokes and compound `EditSpritePixelsCommand` synchronizing frame and atlas texture seamlessly with `CompositionMode_Source` .



2D Polygon & Intelligent CDT Mesh Packing

- **Alpha Contour Extraction:** Automatic watertight Marching Squares contouring on sprite alpha silhouettes.
- **Intelligent Simplification:** Ramer-Douglas-Peucker (RDP) boundary reduction with outward normal dilation (0–8px padding) to prevent edge pixel clipping, with customizable vertex budget (3–48 vertices).
- **Constrained Delaunay Triangulation (CDT):** Feature-preserving mesh generation supporting Color Distance Mode and Sobel Edge Filter Mode.
- **Interactive Knife / Cut Tool (`mesh_knife`):** Draw laser cut lines across polygons to manually slice geometry and instantly re-triangulate concave or complex sections.
- **Direct Canvas Vertex Editing:** Click-drag individual vertices, multi-select vertices with `Shift` or `Ctrl` , nudge with arrow keys, double-click edge to insert a vertex, and hit `Delete` to remove vertices with automatic re-triangulation.

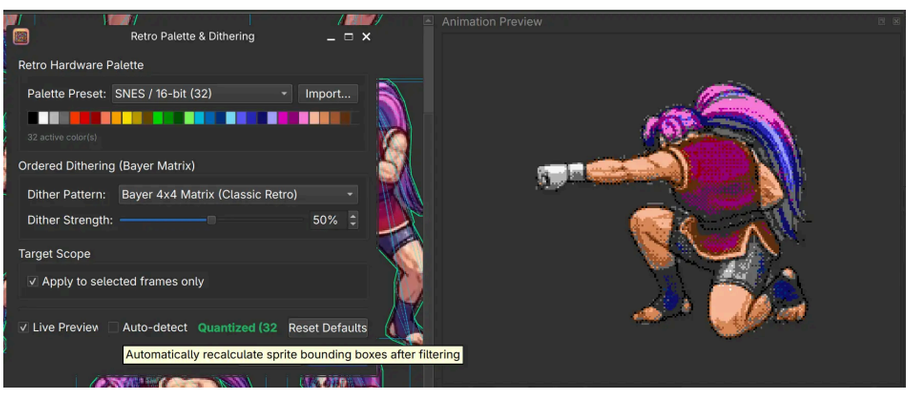
- **Batch Animation Propagation:** One-click option to apply the tuned polygon mesh across all frames of an animation sequence.
- **High-Density Tight Polygon Packing (Nesting):** Compaction allowing bounding boxes to overlap without pixel collisions. Multi-threaded candidate evaluation with configurable thread count (up to hardware cores) and sub-20ms instant calculation.
- **GPU Overdraw Reduction:** Slashes up to 60–80% of transparent pixel fillrate on mobile and desktop GPUs.
- **Multi-Engine Exports:** Godot 4 `_mesh.tres` (`ArrayMesh`), Unity `.unity.json` (`SpriteMeshType.Tight`), and Unreal Engine Paper2D `.paper2d.json` .



Non-Destructive Filter & Cleanup Suite

- **Frame Context Menu & Targeted Scope:** Right-click directly on any sprite in the atlas or timeline to apply filters to the current selection, the whole frame, or batch-apply to all frames in the animation.
- **Live Debounced Preview & Universal Undo:** Real-time feedback with non-destructive rollback via `QUndoStack` (`ApplyFilterCommand`) across 9 specialized filters:
  - **Chroma Background Removal (`Ctrl+Shift+B`):** Dominant color auto-detection, alpha thresholding, and smart crop.
  - **Despill / Anti-Halo:** Clean up 1-pixel colored fringes left by antialiased edges using *Color Clamping* or strict removal.
  - **Outline & Silhouette Generator:** 1-4px customizable stroke (4-connected or 8-connected) and solid hit-flash silhouette masks.
  - **Color Swap (Alt-Skins):** Instant palette replacement preserving pixel art shading gradients (HSV).

- **Color Adjustment:** Real-time Hue, Saturation, Value, Brightness, and Contrast grading.
- **Retro Palette & Bayer Dithering:** Color quantization to 16 historical hardware palettes with Bayer dithering (2x2, 4x4, 8x8) or Floyd-Steinberg, with adjustable strength slider.
- **Pixel Art Rescale:** Pixel-perfect integer rescaling without bilinear blurring.
- **Atlas Bin-Packing (MaxRects) (`Ctrl+Shift+P`):** Multi-heuristic 2D box packing (Best Short Side Fit, Best Area Fit, Best Long Side Fit, Bottom Left, Contact Point).
- **Tight Polygon Packing (Nesting) (`Ctrl+Shift+T`):** Multi-threaded high-density concave/convex polygon packing allowing bounding boxes to overlap.



Timeline & Animation Filmstrip

- **Filmstrip Dock:** Intuitive bottom timeline with thumbnail filmstrip, scrub bar, and drag-and-drop frame reordering.
- **Frame Merging & Compositing:** Drag and drop one frame onto another to fuse them into a single layered sprite.
- **Batch Editing:** Multi-selection operations including group deletion, invert selection, and reverse frame ordering (ideal for ping-pong loops).
- **Real-time Playback:** High-precision preview engine with configurable FPS (1 to 60 FPS), loop controls (Loop, Once, Ping-Pong), and aspect-ratio stabilization.

Interactive Anchor Points & Pivots

- **Zero-Jittering Animation Stabilization:** Shared animation envelope alignment guarantees rock-solid anchored characters during motion, attacks, or jumps without visual shaking or sliding.

- **Dual-Surface Direct Editing:** Move the high-contrast reticle directly within the atlas slices or right inside the live animation preview with real-time coordinate updates and Shift+Click snapping.
- **Instant Cardinal Presets & Batch Actions:** One-click shortcuts for Ground/Bottom-Center, Center, and Top-Left (UI), with batch propagation to entire animations or the full project.
- **Intelligent Preview Navigation:** Automatic optimal framing (*Fit In View*), 5000% razor-sharp pixel art wheel zoom, and pan controls.

Native `.bento` Project Architecture

- **All-in-One Compressed Format:** `.bento` project files package project metadata, frame descriptors, and source assets into a structured ZIP archive.
- **Atomic Transactions & Crash Recovery:** Atomic file writing with journaled crash-recovery safeguard prevents project corruption.
- **Embedded Git Time-Travel:** Integrated non-destructive versioning engine powered by LibGit2. Browse commit history, inspect visual diffs, and revert to previous states without leaving the app.

Multi-Engine Export

- **PNG Sprite Atlas:** Optimized packing of extracted frames into a consolidated texture sheet.
- **Non-Blocking Background Export:** Heavy compression tasks (KTX2 UASTC, high Zstd levels) run asynchronously via `QtConcurrent` with animated progress status, keeping the GUI perfectly responsive.
- **Godot 4 Engine Exporter:** Generates ready-to-use Godot 4 `SpriteFrames` (`.tres`) resources with embedded `AtlasTexture` definitions, animations, and companion `_mesh.tres` 2D `ArrayMesh` resources.
- **Unity 2D SpriteSheet Exporter:** Generates `.unity.json` metadata compatible with Unity's `SpriteMeshType.Tight` , including normalized UVs, inverted Y-axis coordinates, and face indices.
- **Unreal Engine Paper2D Exporter:** Generates `.paper2d.json` descriptors containing both render and collision polygon geometries.

- **TexturePacker / Aseprite JSON:** Universal JSON metadata mapping frame bounds `(x, y, w, h)`, animation tags, and optional mesh geometry (`vertices` , `verticesUV` , `triangles`).

GPU VRAM Hardware Texture Compression

- **Direct GPU Memory Upload (Zero CPU Decompression Overhead):** Bypasses CPU software raster decompression at game runtime. Uploads blocks straight into VRAM, slashing memory bus bandwidth and reducing game asset load times to near-zero.
- **Universal KTX2 Containers & Khronos Basis Universal:** Powered by Khronos `basis_universal` (v2.50) with multi-threaded C++ encoding.
- **UASTC 4x4 Mode (Recommended for Pixel Art):** High-fidelity 8 bpp block compression delivering **75.0% VRAM memory savings** while preserving crisp sprite pixel contours and delicate alpha antialiasing. Transcodes on-the-fly to native GPU formats: **ASTC** (iOS, Android, Switch, Apple Silicon), **BC7** (PC DirectX 11/12, Vulkan, PS5, Xbox), or **ETC2**.
- **ETC1S Ultra-Compact Mode:** Global codebook vector quantization yielding up to **87.5% VRAM footprint reduction** for lightweight UI and mass sprite textures.
- **Lossless Zstandard Supercompression:** Integrated Zstd compression (levels 1 to 22) shrinking file size on disk while retaining GPU block layout.
- **Live VRAM Telemetry in ExportDialog:** Real-time feedback comparing raw RGBA8888 vs UASTC/ETC1S memory footprint with instant percentage savings.
- **Integrated Game Engine Companions:** Emits `.ktx2` companion atlas files natively referenced by Godot 4 (`res://atlas.ktx2`), Unity (`KtxUnity`), Unreal Engine, and WebGL/WebGPU.

Comprehensive Settings & Preferences Center

- **Modular 6-Category Configuration:** Clean searchable preferences interface with responsive filtering.
- **Multilingual UI (i18n):** Complete localization in English, French (Français), and Japanese (日本語) with hot-switch in Preferences.
- **Startup & Automation:** Auto-reopen last active `.bento` project and optional background check for newer

GitHub releases.

- **Export & VRAM Defaults:** Pre-configure favorite target engines, texture format, packing algorithm, and Zstd level.
- **Plugin Management:** Real-time inspection of loaded filter and extractor plugins with metadata, supported extensions, and hot-reload.
- **Integrated Update Engine:** Direct GitHub releases API query, markdown release notes preview, and instant version comparison.

Tech Stack & Architecture

- **Language:** C++17
- **Framework:** Qt 6 (Core, Gui, Widgets, Multimedia, Concurrent, Network, Test, LinguistTools)
- **Core Engine:** `libBentoPackCore` shared library with clean exported API symbols (`BENTOPACK_CORE_EXPORT`) and `libBentoPackWidgets` (`BENTOPACK_WIDGETS_EXPORT`)
- **Dynamic Plugin System:** Hot-loadable Qt 6 plugins (`QPluginLoader`) for both **Filters** (`plugins/filters/`) and **Codecs / Extractors** (`plugins/extractors/`)
- **Plugin Developer SDK:** Installed public headers, CMake package configuration (`BentoPackConfig.cmake`), and reference examples (`examples/sample_filter_plugin` , `examples/sample_extractor_plugin`)
- **Hardware Texture Compression:** Khronos `basis_universal` (v2.50) integration for direct GPU memory upload (KTX2, UASTC, ETC1S, Zstd)
- **Build System:** CMake 3.20+ with modular architecture (`BentoPackCore` + `BentoPack` GUI + `bentopack-cli` + 15 automated CTest suites)
- **Versioning Engine:** LibGit2 (optional, enabled when detected)

Building and Running

Prerequisites

- **C++17 compliant compiler:** GCC 11+, Clang 13+, or MSVC 2019/2022
- **CMake:** Version 3.20 or newer

- **Qt 6:** Version 6.5 or newer (Core, Gui, Widgets, Multimedia, MultimediaWidgets, Concurrent, Test, LinguistTools)
- **Ninja** or **Make** (recommended build generators)
- *(Optional)* **LibGit2** (for embedded `.bento` time-travel versioning)

Build Instructions

Linux (Ubuntu / Debian)

```
# 1. Install prerequisites
sudo apt-get update
sudo apt-get install -y build-essential cmake ni
  qt6-base-dev qt6-multimedia-dev qt6-tools-dev

# 2. Clone repository
git clone https://github.com/oktailb/BentoPack.g
cd BentoPack

# 3. Configure and build
cmake -B build -G Ninja -DCMAKE_BUILD_TYPE=Relea
cmake --build build

# 4. Run tests
ctest --test-dir build --output-on-failure

# 5. Launch
./build/bin/BentoPack
```

Windows (MinGW 64-bit or MSVC)

```
# 1. Clone repository
git clone https://github.com/oktailb/BentoPack.g
cd BentoPack

# 2. Configure with CMake
cmake -B build -DCMAKE_BUILD_TYPE=Release

# 3. Build project
cmake --build build --config Release --parallel

# 4. Run automated test suite
ctest --test-dir build --output-on-failure -C Re

# 5. Launch
.\build\bin\BentoPack.exe
```

Automated Testing

BentoPack includes an exhaustive test suite powered by `QtTest` and `CTest`, validating core models, codecs, controllers, plugins, filters, and multithreaded pipelines across 15 dedicated suites (**100% CTest pass rate**):

```
ctest --test-dir build --output-on-failure --verl
```

Test Suite	Description
`test_core`	Frame data structures, color detection, and algorithm utilities
`test_project`	`.bento` serialization, atomic saves, journal crash recovery, and LibGit2 versioning
`test_extractors`	Native Aseprite (`.ase` / `.aseprite`), GIF, JSON, Godot 4 `.tres`, and Sprite Sheet detectors using sample assets
`test_controller_project`	Project lifecycle, document state management, and file operations
`test_controller_animation`	Timeline controller, FPS pacing, loop modes, and frame reordering
`test_controller_atlas`	Atlas slicing, box editing, pivot alignment, and nudge operations
`test_filters`	Image processing pipeline: Despill, Outline, Color Swap, Color Adjust, Pixel Rescale, Retro Palette & Bayer dithering (13 tests)

`test_mesh`	Watertight Marching Squares, RDP boundary reduction, Constrained Delaunay Triangulation (CDT), knife tool, tight polygon packing, and engine exports (22 tests)
`test_pixel_editor`	1px Bresenham drawing, multi-layer stack, onion skinning, 16 dynamic retro `.gpl` palettes, CDT mesh synchronization (47 tests)
`test_vram_compression`	Khronos KTX2 containers, UASTC 4x4 & ETC1S compression, Zstd supercompression, and GPU transcoding roundtrips (12 tests)
`test_cli`	Headless CLI parser, TexturePacker & Aseprite batch emulation, Godot 4 generator, and POSIX exit codes
`test_app_config`	Settings manager, persistence, preferences serialization, and theme/i18n options

`test_robustness`	Corrupted inputs, zero-size images, memory limit guards, and stress resilience
`test_gui_integration`	Dialog workflows, UI state synchronization, and widget interaction validation
`test_concurrency_and_security`	Thread safety, race-condition mitigation, and safe path traversal checks

Command-Line Interface (`bentopack-cli`)

BentoPack includes an autonomous, 100% headless console binary **`bentopack-cli`** (symlinked to `bentopack-cli` for backward compatibility) designed for game studio build pipelines and CI/CD runners (zero GUI/display required):

TexturePacker Drop-In Mode (with KTX2 VRAM Compression)

Replaces `TexturePacker` directly in existing build scripts without modifying Makefile or CMake commands:

```
# Standard PNG packing
bentopack-cli --sheet atlas.png --data atlas.jso
  --format json-array --algorithm MaxRects --max
  --padding 2 --extrude 1 --trim-mode Trim --siz
  --enable-auto-alias assets/sprites/*.png

# Direct GPU Hardware Texture Compression (KTX2
bentopack-cli --sheet atlas.ktx2 --data atlas.js
  --texture-format ktx2 --opt ASTC_4x4 --zstd-le
```

Aseprite Batch Mode

```
bentopack-cli -b sprites/*.png --sheet atlas.png
```

```
  --sheet-type packed --list-tags
```

Godot 4 Native Resource & Scene Generation

Generates complete `.tres` `SpriteFrames` with jitter-free margins, UID stability across builds, companion `.ktx2` texture, and an instantiable `.tscn` scene:

```
bentopack-cli pack --format godot4 \
  --sheet res/player_atlas.ktx2 --data res/playe
  --godot-scene res/player.tscn assets/player/*.
```

Native Headless Slicing, Filters & Bento Projects

```
# Auto-slice raw sheet into individual sprites w
bentopack-cli slice --remove-bg --tolerance 15 -

# Apply procedural outline headless
bentopack-cli filter --outline 2 red --output-di
```

```
# Inspect or export .bento project metadata
bentopack-cli bento --info project.bento
```

Reactive Watch & Multi-Instance Daemon (`--watch`)

Live background surveillance with intelligent burst debouncing and directory-isolated locking (`QLockFile`):

```
# Interactive watch with 300ms burst debouncing
bentopack-cli pack --sheet atlas.png --data atla

# Multi-instance daemon mode (isolated per direc
bentopack-cli pack --sheet char_atlas.png --data
bentopack-cli pack --sheet ui_atlas.png --data u

# Stop active daemon guarding a specific folder
bentopack-cli --stop-watch assets/characters/
```

Transparent Drop-In Wrappers

Ready-to-use scripts located in `wrappers/` allow legacy makefiles to call `TexturePacker` or `aseprite` seamlessly:

```
# Windows
wrappers\TexturePacker.bat --sheet atlas.png --d

# Linux / macOS
wrappers/TexturePacker --sheet atlas.png --data
```

Benchmarks & Regression Tracking

Execute the autonomous 24-scenario benchmark suite with performance diffing and KPI reporting:

```
python scripts/benchmark_cli.py
```

Reports are automatically written to `benchmarks/REPORT.md` with historical tracking snapshots in `benchmarks/history/` .

Documentation

Comprehensive documentation is available in the `docs/` directory:

- **User Guide**: Complete, step-by-step illustrated manual covering atlas slicing, timeline animations, anchor pivots, filters, MaxRects & tight mesh packing, surgical pixel editing, and multi-engine exports.

- **Developer & Extension Guide**: Architectural deep-dive, step-by-step tutorial for writing custom I/O codecs (`Extractor`) and image filter plugins (`FilterPlugin`), memory scanline best practices, and headless unit testing.
- **API Reference (Doxygen):** Build the complete C++ API reference with interactive inheritance graphs by running `cmake --build build --target doxygen` (outputs to `build/docs/html/index.html`).
- **UNIX Manual Pages:** Traditional troff/groff manpages for `bentopack(1)` and `bentopack-cli(1)` .

Comparison & Market Positioning

BentoPack bridges the gap between raw asset extraction/cleanup (historically handled by tools like *ShoeBox*), sprite atlas packing (*TexturePacker*), and animation sequencing (*Aseprite / Pixelorama / Godot SpriteFrames*).

Feature / Criterion	BentoPack	TexturePacker
License & Pricing	**Open-Source (Apache 2.0)**	Commercial (~40€)
Core Technology	C++17 / Qt 6	C++ / Qt
Smart Atlas Slicing & Import	🟢 **Advanced (O(N) SpatialGrid, Native Aseprite `.ase` , GIF, JSON)**	🔴 None (requires loose images)
Live Filters & Background Cleanup	🟢 **Yes (Plugin Registry, Despill, Outline, Color Swap, Bayer Dither)**	🔴 None
Timeline & Filmstrip	🟢 **Yes (Filmstrip, Ping-Pong)**	🔴 None (Static preview)

Packing Algorithms	🟢 **MaxRects (5 heuristics), Tight Polygon Nesting (Multithreaded), Shelf, Grid, Auto-Alias, Extrude**	🟢 **Industry Leader (MaxRects, Polygon)**
2D Mesh & Tight Polygon Slicing	🟢 **Yes (Marching Squares, Ear-Clipping, CDT, Knife Tool, Godot/Unity/Unreal)**	🟢 Commercial Feature
Surgical Pixel Editor	🟢 **Yes (Bresenham 1px, Multi-Layers, Onion Skin, 16 Retro Palettes, CDT Mesh)**	🔴 None
Anchor Points / Pivots	🟢 **Yes (Interactive Reticle, Zero-Jittering, Godot 4 / JSON)**	🟢 Yes (All presets)
Embedded Time-Travel	🟢 **Unique (Git / LibGit2 dock)**	🔴 None
Godot 4 Integration	🟢 **Native (`.tres` SpriteFrames & `.tscn`)**	🟢 Supported
Headless CLI for CI/CD	🟢 **Yes (`bentopack-cli` , TexturePacker, Aseprite, Godot 4)**	🟢 **Industry standard**
VRAM Texture Compression	🟢 **Native KTX2, UASTC, ETC1S, Zstd, Direct GPU Zero-Decompress, CLI**	🟢 **ASTC, ETC2, KTX2, Basis (Commercial)**

> 💡 Tip
>
> **Why BentoPack?** While *TexturePacker* excels at packing clean loose PNGs for AAA pipelines and *Aseprite* / *Pixelorama* are dedicated pixel art authoring tools, **BentoPack** is uniquely built to **rescue, decompile, clean, organize, and bridge existing 2D sheets** into production-ready game engine resources without external dependencies.

Licensing & Distribution Model

BentoPack is distributed as **100% Free and Open Source Software (FOSS)**:

- **Entire Codebase (Apache License 2.0):** The entire application, CLI, algorithmic engine, filters, extractors, and engine addons are licensed under the permissive **Apache License 2.0**. You are free to compile, modify, inspect, and use BentoPack in any personal or commercial game project without royalty fees, revenue restrictions, or gatekeeper locks. See LICENSE.
- **Official Binaries (Store Convenience):** For creators who prefer an out-of-the-box experience without building from source, official pre-compiled binaries featuring 1-click installers, code signing, and automatic updates are available on **Steam** and **Itch.io**. Purchases directly support ongoing open-source development.

Developer: Vincent LECOQ

Terms Privacy Security Status Community Docs Contact Manage cookies
Do not share my personal information
© 2026 GitHub, Inc.