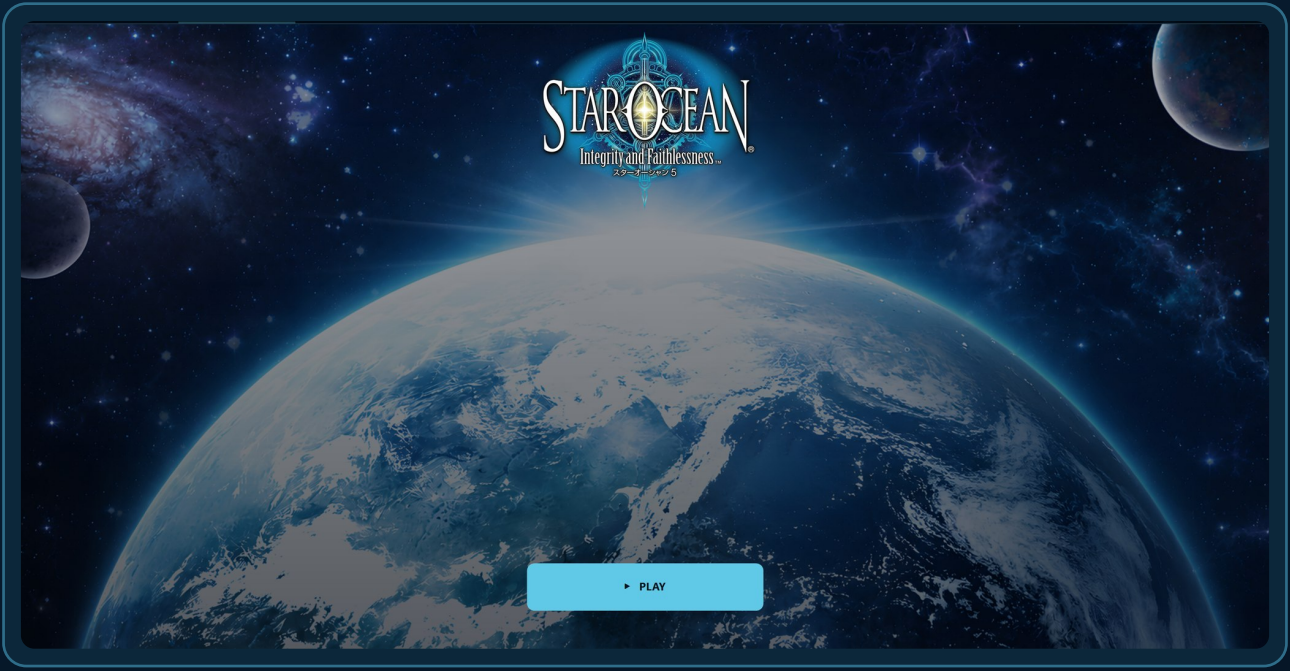
THE COCKPIT LAUNCHER

Set up your game, choose the right options and solve problems — with real screenshots and clear explanations.



THE 60-SECOND OVERVIEW

Your game. One control centre.

The launcher prepares your own game files and remembers your settings. Most players only need the **Game**, **Video** and **Controls** tabs to get started.

Just want to play?

Extract package → Start root launcher → Game → Prepare → Home → Play.
Choose your graphics and controller settings before playing.

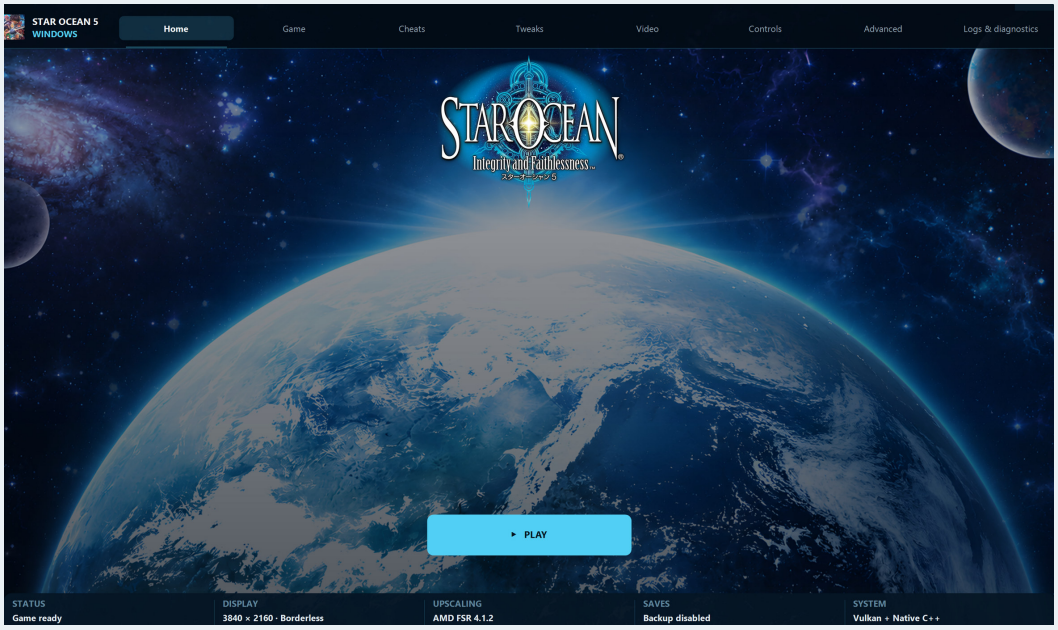
Want to change something?

Video for image quality. **Tweaks** for camera/HUD. **Cheats** for permission. **Logs** when diagnosing a problem.

> **Quick reality check:** This guide describes Windows player **1.1.3** The source screenshots show the **1.1.0 interface**; labels and selected values in them are examples. **Selected ≠ active:** verify the actual applied setting in the game or performance overlay.

BEGINNER FRIENDLY · REAL SCREENSHOTS · CLICKABLE CONTENTS · ADVANCED DETAILS INCLUDED

Three visual cues: cyan = selected tab or action; green = readiness; amber = caution or untested edition. A greyed-out option is unavailable for the current configuration.



Game content is not included in the Windows package. Use only game files you own. For controller shortcuts while playing, see the companion **In-Game Menu Guide**.

YOUR READING MAP

Find it fast.

Use the clickable chapter titles or page numbers to jump to a topic. You don't need to read every page in order.

SKIP TO THE ESSENTIALS

New player? Start with First launch. **Low FPS?** Video, Image and Performance. **Something broken?** Logs and Troubleshooting.

01 **First launch** 4
Go from download to Play in five steps

02 **Home** 5
What the dashboard and status badges mean

03 **Game** 6
Game folders, PKG import, saves and backups

04 **Cheats** 9
What is available and what actually turns on

05 **Tweaks** 11
Leader cycling, camera comfort and HUD layouts

06 **Video** 13
Display choices and image quality explained

07 **Controls** 16
Controller selection and keyboard shortcuts

08 **Advanced** 17
Updates, cache, overlay and optional tools

09 **Logs & diagnostics** 20
Capture useful evidence when something fails

10 **Troubleshooting** 23
Quick remedies and known limits

11 **AMD FSR SDK appendix** 24
Advanced: provider verification, packaging and limits

CHAPTER /

First launch

Before you start	What you need
Operating system	Windows 10 or 11, x64.
Graphics	A working Vulkan 1.3 driver and compatible GPU. Reconstruction has additional requirements.
Game	Your own supported PS4 game files or a supported local PKG. Game content is not bundled.
Application	The complete extracted Windows player package in a writable folder.

Five steps to play

1. Extract the complete player package into a writable folder.
2. Start **Star Ocean 5 Launcher.exe** at the package root. It opens the real launcher in `bin`.
3. Open **Game**, then select your extracted game folder or import a supported `.pkg`.
4. Wait for game validation and preparation. Check the detected Title ID and status.
5. Review **Video** and **Controls**, return to **Home**, and select **Play**.

> IMPORTANT
>
> Start the**root launcher** and keep the complete package together. Moving only the executable breaks access to the applications, libraries and assets in the package.

Keep `bin`, the asset folders, `VERSION` and the documentation together. Do not move just the root executable to another folder. Players do not need Python, MSYS2, PowerShell or .NET; building the source requires the toolchain described in the Windows build instructions. A compatible x86-64 CPU and a working Vulkan 1.3 graphics driver are required. Game files and firmware are not included.

CHAPTER /

Home

Page	Use it for
Home	Start the game and read the configuration dock.
Game	Prepare game files, choose saves and manage backups/ shortcuts.
Cheats	Review the native catalogue and allow access to it in game.
Tweaks	Choose leader, camera-follow and HUD-layout settings.
Video	Select GPU, monitor, output and rendering options.
Controls	Choose a detected input device and customize keyboard bindings.
Advanced	Language, updates, cache, overlay, capture tools and fast-forward.
Logs and diagnostics	Read, select, copy and export diagnostic output.

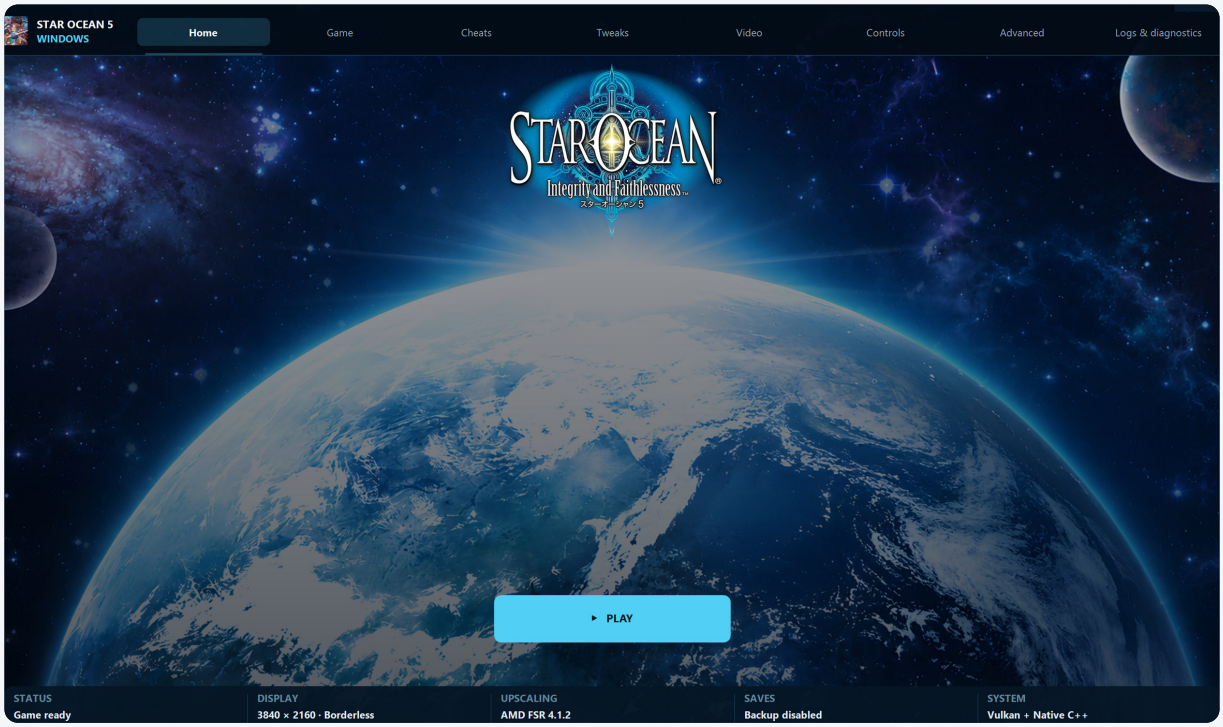


Figure 1 — Home: game logo, Play and the bottom configuration dock.

The main page presents the game logo, **Play** and a bottom status dock. Select and prepare a valid game before playing. The dock summarizes the current configuration; it does not guarantee compatibility or a particular frame rate. The launcher minimizes when starting the game. Use its caption buttons to minimize or close; closing asks for confirmation. Drag the top header's free area to move the window. Resize the window for your screen; the interface scales with available space. Scroll longer settings pages to reach all controls.

CHAPTER /

Game

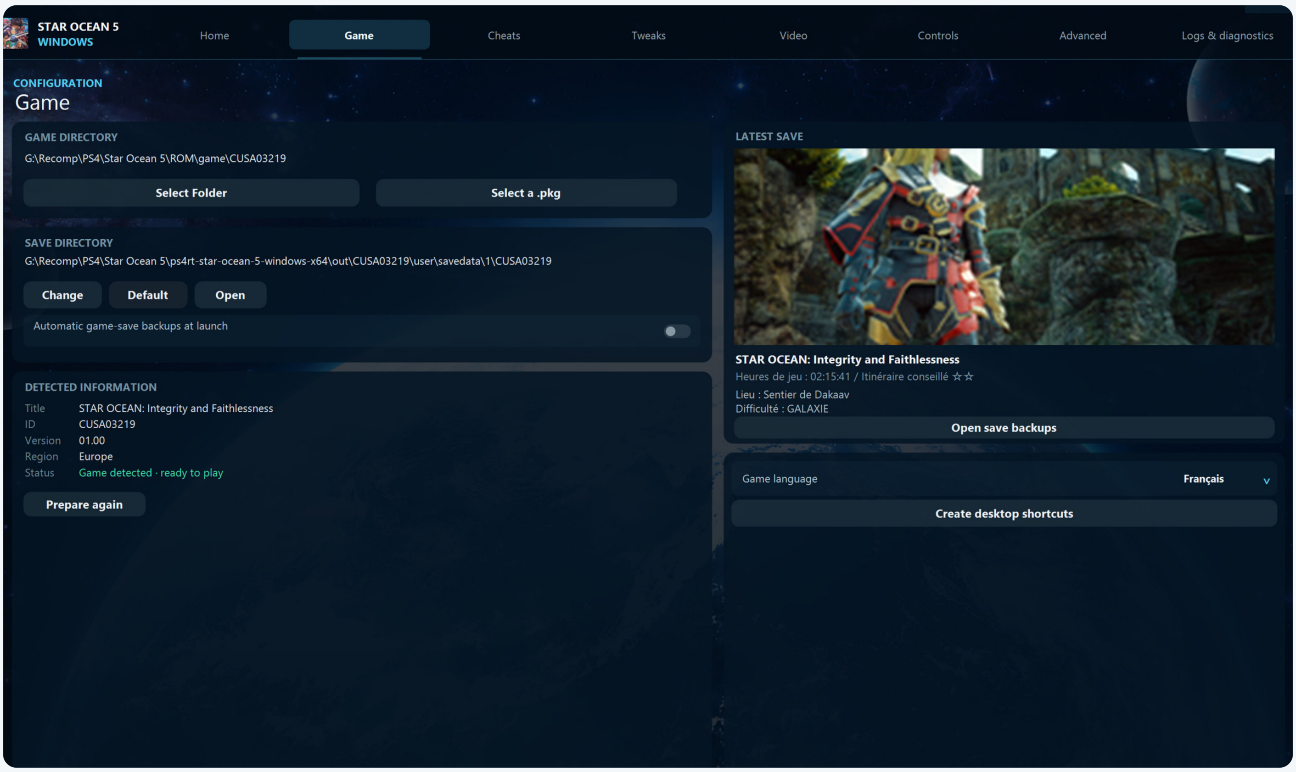


Figure 2 — Game: detected content, current save path, latest-save metadata and desktop shortcuts.

Select an extracted folder

Select **Select Folder**, then choose the actual game directory containing `eboot.bin`, `sce_sys` and `sce_module`, rather than a parent directory or an individual executable. The launcher remembers the selection. **Prepare game** creates the local working files; **Prepare again** repeats preparation when a game is already configured. Preparation does not modify the original game extraction.

Detected information shows the title, Title ID, game version, region and readiness after validation. The game version here is the PS4 content version, not the Windows launcher version.

Indicator	Meaning
Green — `CUSA03219`	Tested European PS4 edition.
Orange — `CUSA03246`, `CUSA01722` or `CUSA03567`	Recognized PS4 edition; accepted with an untested-edition warning.
Red — unrelated or unsupported ID	Launch is refused. PS3 editions are not supported.

Acceptance of another region does not establish compatibility with every update, save or language for that edition. The ID badge is shown only after game validation.

> GOOD TO KNOW
>
> The**game version** is the PS4 `APP_VER`. The **Windows version** is the application version shown at the bottom of the launcher. These numbers serve different purposes.

PKG import

1. Select **Select a .pkg** and choose your locally owned PS4 package.
2. Review the import dialog and choose the parent extraction directory. The launcher remembers it.
3. Confirm extraction and watch the expanded-byte progress. Detection alone does not mean extraction is finished.
4. Wait for extraction, content validation and preparation to complete before selecting **Play**.

A completed extraction is stored in a folder named after the detected Title ID, for example `<chosen parent>\CUSA03219`. An existing destination is not silently replaced. Canceling an import cleans up the partial import created by that operation.

Import uses the independent native `bin/PkgExtractor.exe`. Its supported PS4 PFS/PFSC path needs the package creation passcode; the launcher uses the standard all-zero passcode. Unknown passcodes, unsupported structures and invalid packages are reported in diagnostics. Retail-license/key recovery and downloading PSN content are not provided. Selecting a PKG does not make the runtime execute the compressed archive directly.

Save directory and backups

Control	Action
Change	Select the live-save location.
Default	Restore the application's standard live-save path.
Open	Open the live-save folder shown on the page.
Automatic game-save backups at launch	Enable or disable the shared backup policy.
Open save backups	Open backup snapshots for the detected Title ID.

Save directory displays the current Windows path. **Change** selects a custom live-save location, **Default** restores the standard location, and **Open** opens the current one. Changing the setting selects a location; it is not a promise to convert or migrate another edition's saves. Stop the game before replacing live save files. The launcher's preview releases its image files so you can replace saves without closing the launcher.

For a standard installation, paths below are relative to the player folder:

Data	Location
Launcher preferences	`out/launcher.ini`
Prepared game and runtime settings	`out/CUSA03219/`
Live saves	`out/CUSA03219/user/savedata/1/<Title ID>/`
Backup snapshots	`out/CUSA03219/user/savedata-backups/1/<Title ID>/<slot>/backup-.../`

The current working-root name remains `CUSA03219` even for another accepted edition; the live-save and backup subfolders use the detected Title ID. A custom live-save location does not relocate the backup root.

Automatic game-save backups at launch enables the shared native backup engine for pre-launch snapshots and supported backup requests made by the game after saving. Disabling it preserves normal save/load and existing backups. **Open save backups** opens the detected edition's backup directory. Completed snapshots contain `data` and, when available, `sce_sys` metadata; they are not ordinary files to drop directly into a live-save slot.

Completed backups are held read-only against replacement/deletion while the launcher is running. Close the launcher before manually replacing a backup. This protection does not lock live saves or replace a separate copy kept outside the application folder.

> WATCH OUT
>
> **Live saves and backups are different folders.** Exit the game before replacing live saves. Close the launcher as well before replacing protected backup snapshots. Keep an independent copy before restoring a save; selecting a path does not migrate its contents.

Latest save, language and shortcuts

The latest-save card displays the available image and details from save metadata, such as play time, location and difficulty. It is a preview, not a save-slot selector or a command to load that slot in the game. Returning to **Game** refreshes the metadata.

Game language offers the languages configured for the detected edition. It remains independent of the launcher language. **Create desktop shortcuts** creates two desktop links:

- **Star Ocean 5 Launcher** opens the launcher normally.
- **Star Ocean Integrity and faithlessness** starts the selected game with the last saved settings, without showing the launcher interface. Preparation, GPU/display checks, controller/keyboard mappings and the selected save-backup policy still apply.

A hidden launcher process supervises quick play and exits after a successful game session. If setup is incomplete or startup fails, the launcher opens for setup or diagnostics. Quick play does not interrupt gameplay with automatic update checks. Both links use the application icon and the stable application entry point. Keep the player folder in place; after moving it, recreate the links from its launcher. Older `Star Ocean 5` links can be removed manually once the new links have been created.

CHAPTER /

Cheats

USEFUL TIP

The launcher is the catalogue and permission switch. Individual cheats are activated in the running game's**Cheats** tab; enabling access here does not turn every code on.

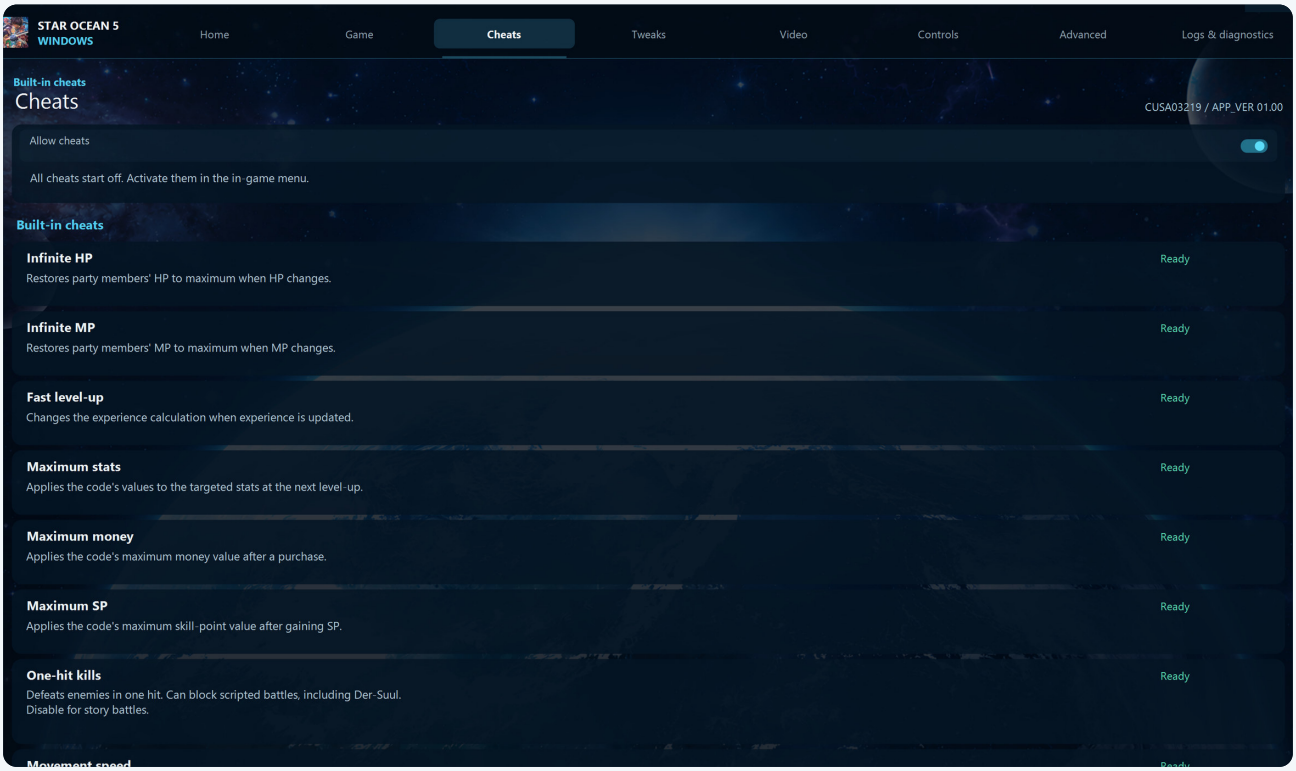


Figure 3 — Native cheat catalogue. Ready marks availability; scroll to view the full list.

The native catalogue is checked against Title ID, APP_VER and original executable bytes. Enable access here, then apply individual codes in the in-game Cheats tab. All begin off. The catalogue contains eight native codes:

Cheat	Trigger / effect
Infinite HP	Restore party members' HP to maximum when HP changes.
Infinite MP	Restore party members' MP to maximum when MP changes; independent of HP.
Fast level-up	Change the experience calculation when experience is updated.
Maximum stats	Apply the code's targeted stat values at the next level-up.
Maximum money	Apply the code's maximum money value after a purchase.

Cheat	Trigger / effect
Maximum SP	Apply the code's maximum skill-point value after gaining SP.
One-hit kills	Defeat enemies in one hit; disable for scripted story battles.
Movement speed	Set party movement to ×1, ×1.5 or ×2; enemies and NPCs retain their normal speed.

Movement speed has three positions: `x1` (off), `x1.5` and `x2`. Money/SP/stat codes take effect at the event named in their description. Disabling a code does not remove values already obtained. One-hit kills can block scripted battles; disable it for story battles. External cheat-profile import and the unsafe mass-inventory code are no longer offered.

> **WATCH OUT**
>
> Turning a code off stops its ongoing effect. Levels, stats and currency already obtained are not rolled back. Use a separate save before experimenting.

CHAPTER /

Tweaks

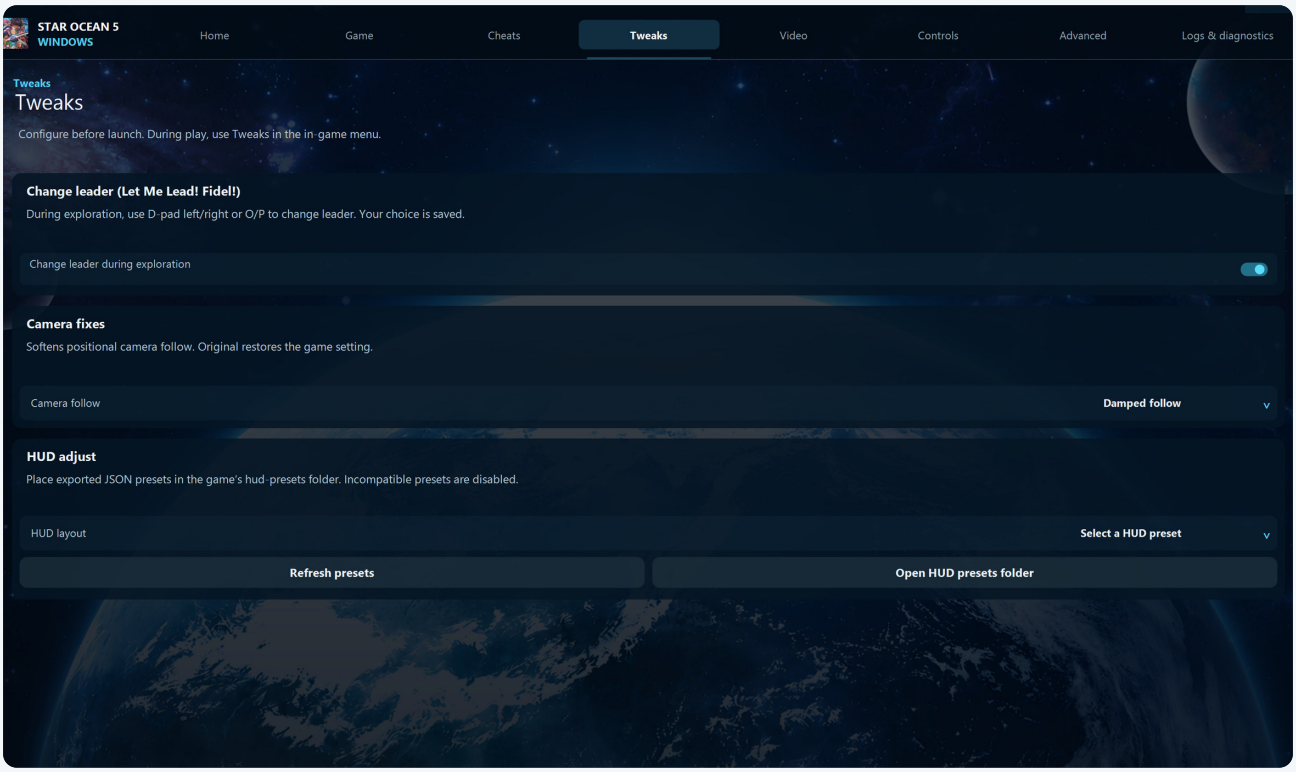


Figure 4 — Tweaks: exploration leader, camera follow and HUD preset library.

Configure startup choices here. During play, use the matching **Tweaks** tab in the in-game menu.

Adjustment	Control	Result
Change leader (Let Me Lead! Fidel!)	Enable, then use **O / P** or D-pad **left / right** during exploration.	Cycle eligible party members. The saved choice is restored when the game state permits it.
Camera fixes	Original / Light damping / Damped follow.	Adjust the game's positional follow response.
HUD adjust	Choose a compatible JSON preset.	Apply mapped HUD placement, scale and visibility.

Leader cycling is blocked while **L2 or R2** is held. After switching, move clear of a zone entrance before using it; transition guards reduce repeated entrance triggers.

> WATCH OUT
>
> **Known issue — Fidel's follow behaviour:** Fidel can still fail to follow the party in some locations when another character leads. This is not considered fixed in 1.1.3. Restore Fidel as leader if this occurs. Save normally before testing a different leader.

Camera fixes preserves manual rotation and collision handling. Damped follow softens abrupt positional movement; it does not lock camera height, fully remove terrain-induced vertical motion, or provide a free camera/FOV slider. Compare modes along the same route.

Use a HUD preset

1. Select **Open HUD presets folder**.
2. Copy the compatible `.json` file into that folder.
3. Select **Refresh presets**, then choose the layout in **HUD layout**.
4. Start the game, or select the layout from **Tweaks > HUD adjust** during play.

Use **Original** to restore original HUD placement. Without an active custom preset, the HUD keeps its original layout. The loading blackout still covers the whole output. Presets support `4:3`, `16:10`, `16:9`, `21:9` and `32:9`; only mapped groups can be adjusted. The game/version compatibility checks still apply, and source JSON files are preserved.

Preset background setting	Behaviour
Original	Retain that supported background's original sizing.
Extended	Fill the output in width and height, including narrow aspect ratios.
No explicit override	Use the runtime's background policy.

Explicit schema-4 background settings take priority over automatic sizing. Older schema-1, 2 and 3 presets remain readable. Background sizing is separate from foreground text, icons and videos; extending a backdrop does not stretch those elements. Pause controls, including the Options/Back prompt, have their own mapped group. Subtitle adjustment remains incomplete and should not be relied on.

CHAPTER /

Video

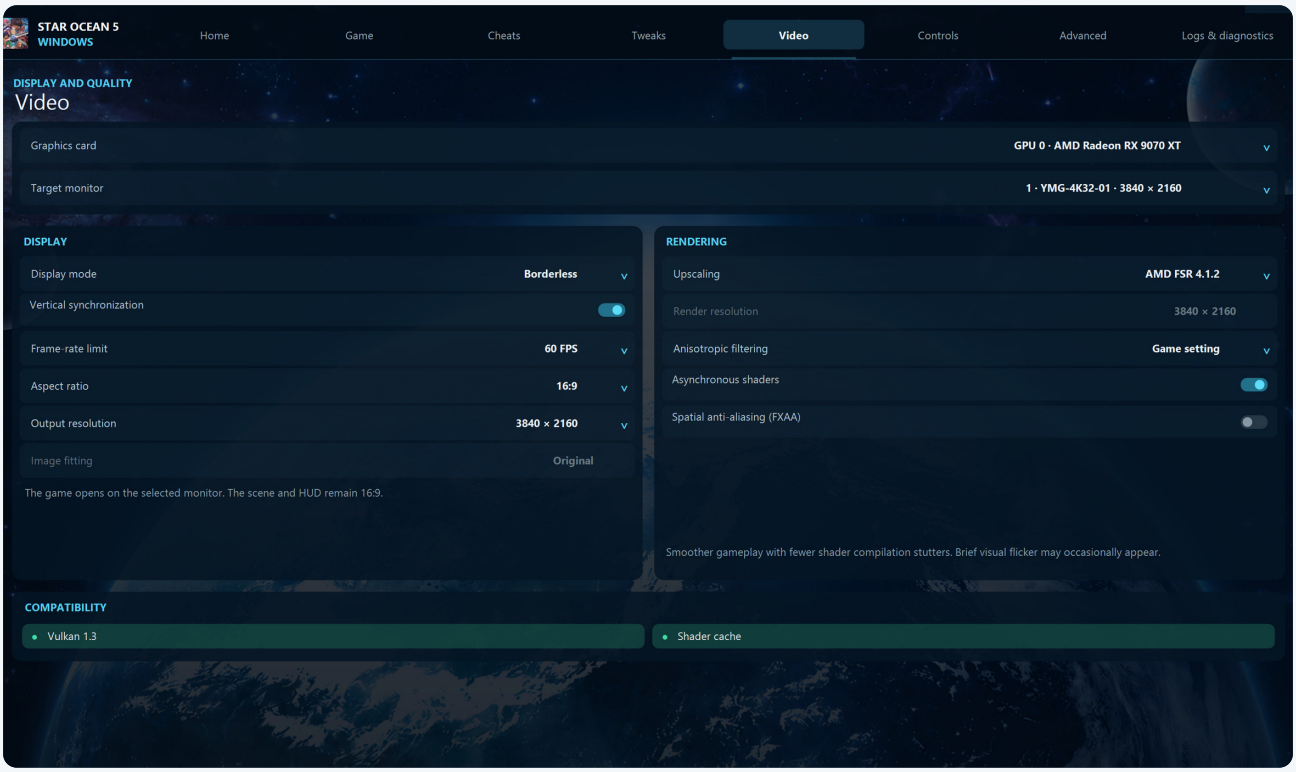


Figure 5 — Video: GPU and monitor selection above separate Display and Rendering panels.

Choose the GPU and target monitor before starting a game. Output sizes/aspect filters are restricted to the selected display's supported catalogue/modes. Fullscreen requires a verified display mode; Borderless retains the desktop mode. A selected Windowed size sets the initial client dimensions, and manual mouse resizing remains possible.

Display controls

Control	What it changes
Display mode	Windowed, Borderless or Fullscreen presentation.
Vertical synchronization	Requests synchronized presentation to reduce tearing.
Frame-rate limit	Sets the presentation cap, normally `30` or `60` FPS in the launcher. A cap does not guarantee that frame rate.
Aspect ratio	Filters output choices: `All`, `4:3`, `16:9`, `16:10`, `21:9` or `32:9`, when verified choices exist.
Output resolution	Selects the presentation dimensions from the project's supported catalog and the verified monitor modes.
Image fitting	In Windowed mode, Original preserves proportions; Stretch to window fills the window and can distort the scene and HUD.

Rendering controls

The source supports adjusted 3D projections for `4:3`, `16:10`, `21:9` and `32:9` on the guarded `CUSA03219` / `APP_VER 01.00` path. Choose the output before launch; changing the projection ratio requires restarting the game. The `4:3` and `16:10` paths and their HUD/loading-mask corrections have been accepted in local testing. Unsupported executables retain the original `16:9` projection. Monitor-mode validation still applies.

This does not automatically reposition every HUD element: use a compatible HUD layout. Prerendered video retains its original proportions. Foreground game menus remain centered; supported backgrounds can fill the output according to the HUD preset. Image fitting controls final window presentation, separately from those layout rules.

With reconstruction **Disabled**, manual internal scene sizes include `720p`, `1080p`, `1440p` and `2160p`, subject to runtime/device checks. With FSR/DLSS enabled, its presets own the input size. Output resolution, internal scene resolution and original UI texture quality are different things. A higher internal size does not replace source textures.

Control	What to expect
Upscaling	Disabled, **AMD FSR 3.1.5**, **AMD FSR 4 v7 (INT8)**, **AMD FSR 4.1.2** or **DLSS**, according to GPU, component and runtime checks.
Render resolution	Manual internal scene size with reconstruction disabled. With reconstruction enabled, the in-game Image preset controls the input.
Anisotropic filtering	Game setting, `x2`, `x4`, `x8` or `x16`. Improves eligible textures viewed at an angle; requires a game restart.
Asynchronous shaders	Reduces first-use compilation stalls, with a possible tradeoff of temporary visual omissions or flicker.
Spatial anti-aliasing (FXAA)	Optional SDR edge smoothing in the final image, including the HUD. Off by default; can soften fine detail.

Choose a reconstruction provider

Selector	Implementation	Availability
Disabled	Render without temporal reconstruction; choose the internal size manually.	Subject to render-size/device limits.
AMD FSR 3.1.5	Integrated FSR 3.1.5 reconstruction.	GPU policy and required Vulkan features.
AMD FSR 4 v7 (INT8)	Supplied v7 INT8 model/shader bundle.	Compatible GPU, model assets and output dimensions.
AMD FSR 4.1.2	Official AMD provider integrated through SDK 2.4.	Installed provider components, driver and runtime feature validation.
DLSS	NVIDIA reconstruction through the optional bridge/runtime.	Compatible NVIDIA GPU and required components.

Choosing an NVIDIA GPU does not automatically choose DLSS or disable compatible FSR paths. The selected provider owns its preset. DLSS Native (100%) is reported as **DLAA** when active; other DLSS presets use Super Resolution. XeSS and frame generation are not exposed.

> **IMPORTANT**
>
> **Selected is not the same as active.** Runtime checks can reject a request or select a safe fallback. Check the in-game applied resolution, performance overlay and diagnostic log for the provider actually used. A DLL or a menu label alone does not prove support.

The official AMD provider and the v7 INT8 path are separate implementations. Their assets are not interchangeable. See SDK integration for component details and the in-game preset reference for input/output examples. FSR can cost more than it saves on older hardware. The lower-tier FSR 3.1 override under Advanced cannot bypass missing hardware capabilities.

Anisotropic filtering affects eligible textures viewed at an angle and requires restart. Asynchronous shaders reduce compilation stalls but can briefly omit/flicker some rendering. VSync and frame caps control presentation; they do not guarantee a sustained frame rate. See Image for live presets and sharpening.

CHAPTER /

Controls

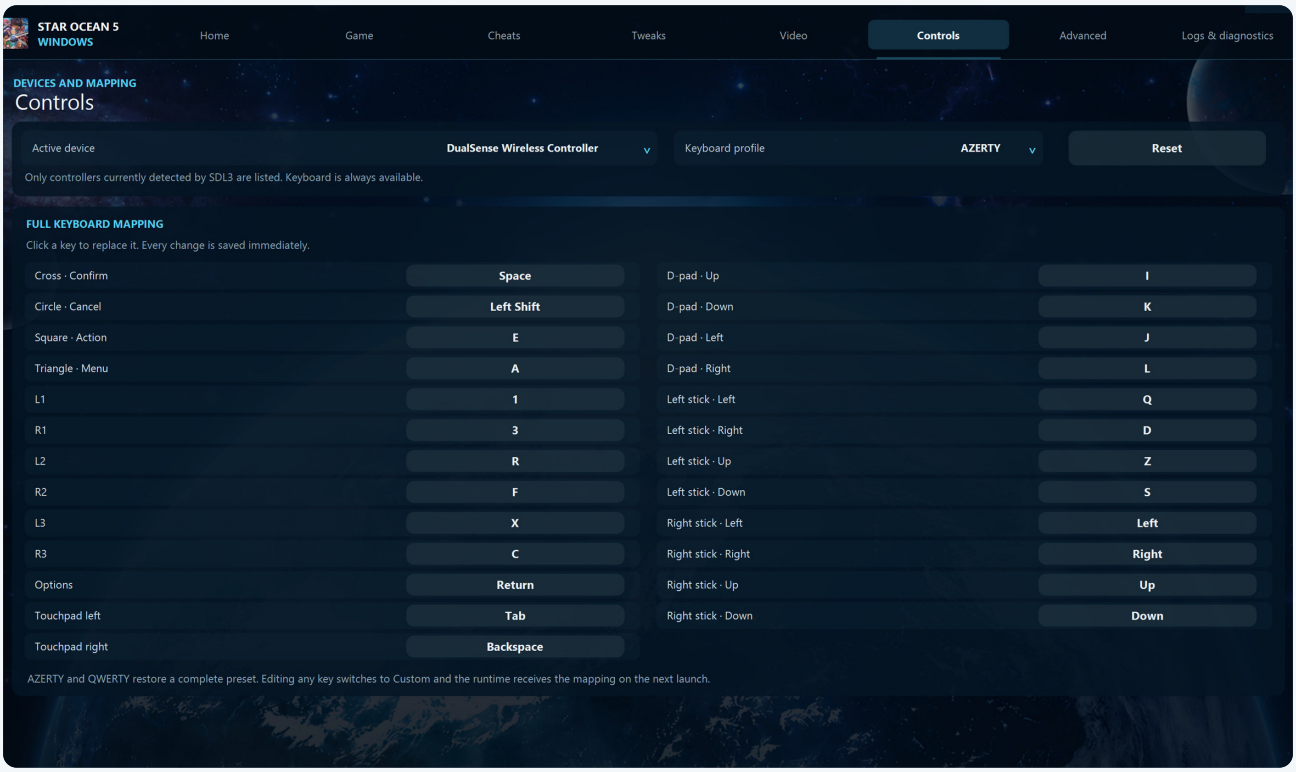


Figure 6 — Controls: a detected DualSense and the editable AZERTY keyboard profile.

Active device lists SDL-detected controllers and **Keyboard**. DS4, DualSense and Xbox-style devices use SDL's standardized controls. The runtime automatically handles controller activity; the displayed name is not an exclusive lock to a particular hardware ID when several controllers are connected. Choosing Keyboard prevents gamepad input.

Keyboard profile provides `AZERTY` and `QWERTY` defaults. To customize a binding:

1. Click its current key button.
2. Press the replacement key, or press **Escape** to cancel capture.
3. Repeat for the other controls and start the next game session.

Changes are saved immediately, switch the profile to **Custom**, and are passed to the runtime on the next launch. Bindings cover the face buttons, shoulders/triggers, stick clicks, Options, both touchpad-click sources, D-pad and both stick directions. **Reset** restores the selected default layout; selecting `AZERTY` or `QWERTY` also replaces the custom set.

The screenshot shows an `AZERTY` example. Read the displayed bindings for your chosen profile instead of assuming those letters are the `QWERTY` defaults. This page provides keyboard remapping; it does not expose a custom gaming-mouse button mapper or arbitrary gamepad rebinding editor. Controller vibration is adjusted in the in-game Controller tab.

CHAPTER /

Advanced

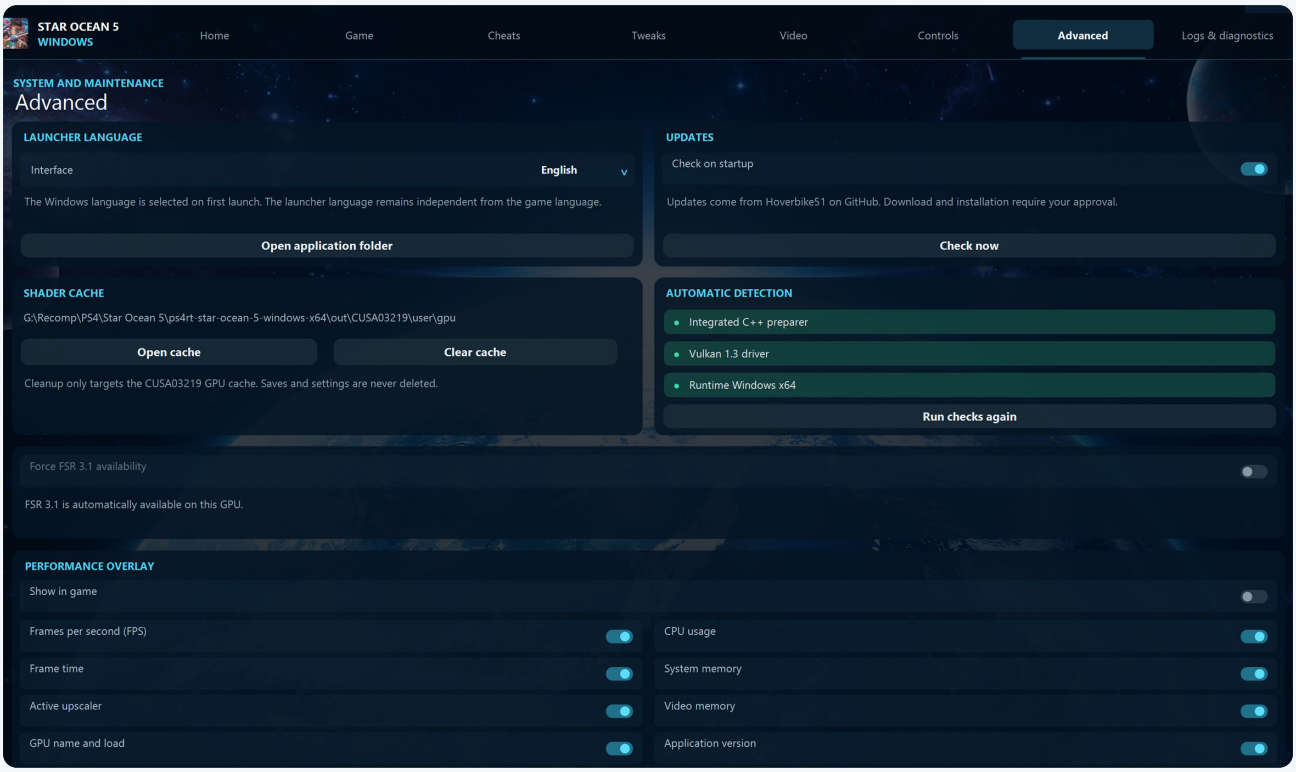


Figure 7 — Advanced: language, updates, cache, detection and overlay visibility.

Language and application folder

Interface selects the launcher's language. Windows language is used on first launch; your saved choice is used afterward. The twelve choices are English, French, Spanish, Portuguese, Brazilian Portuguese, Italian, German, Russian, Simplified Chinese, Traditional Chinese, Korean and Japanese. Game language is independent. Start a new game session to load a changed interface language into the custom in-game menu.

Open application folder opens the installation directory.

Updates

Check on startup enables release checking; **Check now** starts a manual check against Hoverbike51's GitHub releases. It checks Windows player packages, not the original project's Linux release version. A numeric Git tag or GitHub source-code ZIP alone is not an installable update.

When an eligible newer package is available, the dialog offers:

- **Download** — download, verify and extract the package, then hand it to the native installer.
- **Open GitHub** — open the release page without installing.
- **No** — dismiss the offer for this session; startup checks may offer it next time.

> **IMPORTANT**
>
> The installer must wait for the running launcher to exit before replacing its files. Follow the handoff dialog and close the game first. Do not start a second launcher during installation. If replacement fails, keep the update log and recovery folder.

Close the game before installation. Updates verify the archive size, GitHub SHA-256 digest, paths, versioned executables and staged startup before replacing program files. Missing checksums or network errors prevent automatic installation. Source checkouts are not replaced by the player updater. Equal or older versions are not installed over the current version.

The updater preserves `out`, game files, saves, settings, caches and external mod folders. Replaced program files and a recovery journal are kept under `out/updates/.../previous`. A failure attempts rollback and reports the outcome; keep the recovery files until the new installation has been checked. This does not provide a power-loss recovery guarantee.

Shader cache and system checks

Open cache opens the GPU cache location. **Clear cache** requests confirmation and removes shader/cache data, preserving saves and settings. Stop the game first. Clearing the cache is a troubleshooting operation; the next visit to a scene may require shader compilation again. Keep an existing cache for ordinary play.

Run checks again refreshes detection of the native preparer, Vulkan driver and Windows runtime. These checks cannot certify a complete playthrough or every graphics option.

Force FSR 3.1 availability

The project uses a conservative default policy based on recognized desktop GPU classes, with `GTX 1080`, `RX 5700` and `Arc A750` among its reference points. It is not a universal performance ranking for all newer, mobile or unidentified cards.

If a technically capable GPU is outside that default policy, **Force FSR 3.1 availability** can expose it after a confirmation about possible performance loss or instability. The choice is off by default and tied to that GPU. It is unavailable when FSR 3.1 is already admitted normally, or when required Vulkan features are absent. It cannot force `FSR 4` or bypass hardware feature checks. Launch the game again after changing it.

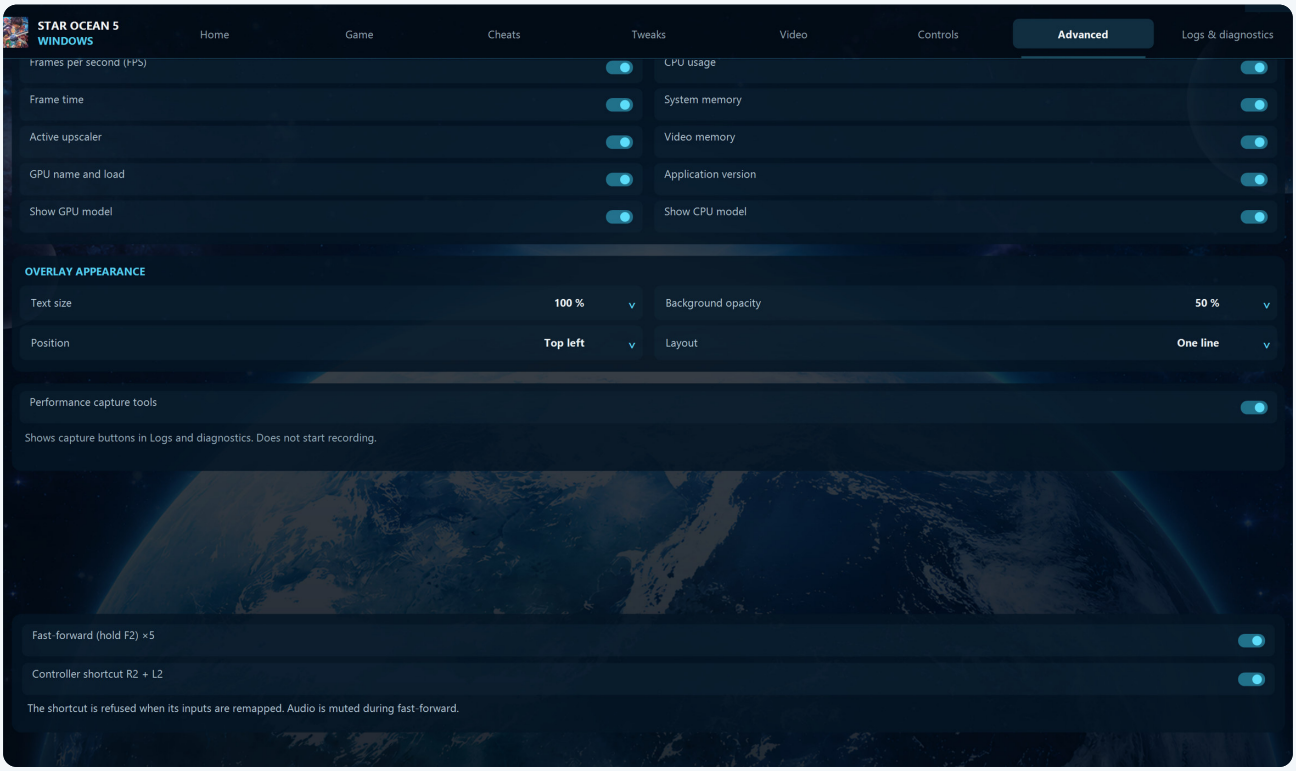


Figure 8 — Advanced, scrolled: model-name switches, overlay appearance, capture tools and ×5 fast-forward.

Overlay and capture tools

Performance overlay controls the on-screen FPS, frame time, active upscaler, GPU, CPU, RAM, VRAM and application-version readouts. **Show GPU model** and **Show CPU model** independently replace the model names with GPU / CPU when disabled; readings remain visible.

Appearance	Available choices
Position	Top left, top center, top right, bottom left, bottom center, bottom right.
Layout	One line, one item per line, or Technical.
Text size	50%–300%.
Background opacity	0%–100%.

The in-game Performance tab exposes the same controls. Its active-upscaler reading includes the applied FSR preset, **DLAA** for DLSS Native (100%), or **NO PRESET** without reconstruction. **Technical** adds frame-time history, the busiest thread, logical-processor count and sampling metadata. A thread at 100% means approximately one fully occupied logical processor, while process CPU is averaged across the available logical processors. The overlay does not record a performance report.

Performance capture tools is off by default. Enabling it only reveals **Capture next session** and **Open captures** in Logs and diagnostics; it does not start a recording. Scroll the Advanced page if the lower controls are outside the window.

Acceleration is off by default. When enabled, hold **F2** or **R2 + L2** for **x5** game-time acceleration; releasing the shortcut restores normal speed. Sound is muted during acceleration. The actual speed depends on available processing capacity. Remapped shortcuts and inactive game windows do not trigger acceleration.

CHAPTER /

Logs and diagnostics

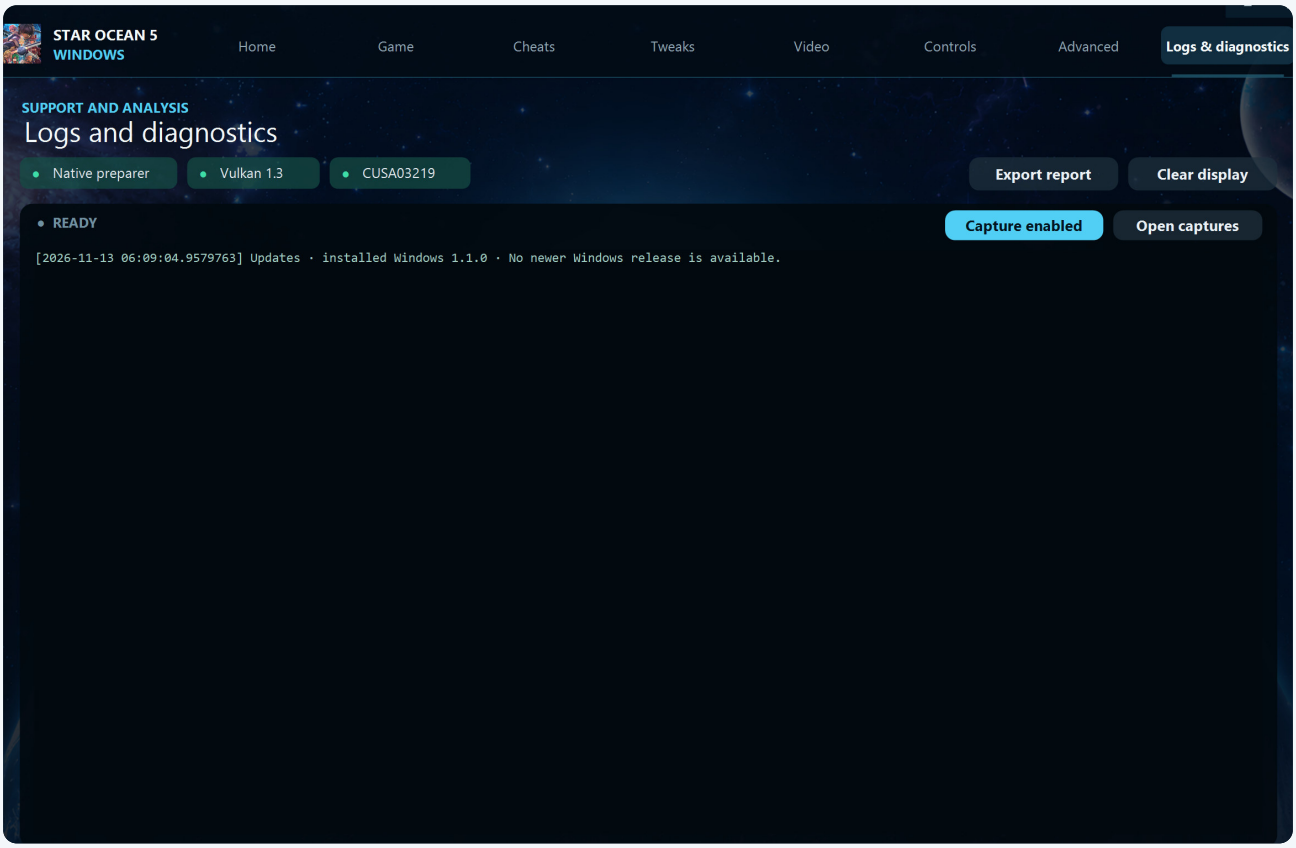


Figure 9 — Logs: selectable text and capture buttons exposed through Advanced.

The badges summarize preparation, Vulkan and the validated game ID. Console messages remain **English**, regardless of the interface language, so reports can be compared consistently. Each completed record receives a local-time timestamp in the format `[YYYY-MM-DD HH:mm:ss.fffffff]`, for example `[2026-11-13 02:33:39.1234567]`. Fractional digits describe the clock representation, not a guarantee of measurement accuracy. Copying, scrolling and exporting preserve the original timestamp; partial runtime output is assembled into complete lines before stamping.

- Drag to select console text; use **Ctrl+A** to select all and **Ctrl+C** to copy.
- Use vertical scrolling to revisit earlier output and horizontal scrolling for long lines.
- **Export report** writes a diagnostic text file under `out/logs/`.
- **Clear display** clears the visible retained log; it does not clear saves or shader caches.

To capture a performance investigation:

1. Enable **Advanced > Performance capture tools**.
2. Open **Logs and diagnostics** and enable **Capture next session**.
3. Start the game, reproduce the problem and exit through the in-game **Exit game** command.
4. Use **Open captures** to find the local CSV/JSON reports under `out/CUSA03219/user/gpu/performance/`.
5. Turn capture off when you no longer need a diagnostic session.

The screenshot shows the capture controls because **Performance capture tools** is enabled. With that Advanced option off, both capture buttons are hidden. **Capture enabled** means the next-session switch has been armed; **Open captures** opens existing local reports. Reports remain local; nothing is uploaded automatically. A normal menu exit allows the runtime to finalize its report; an external forced termination can leave partial data.

Performance probe 1.1.3

> USEFUL TIP
>
> **Capture on the affected computer.** Use the same save, route and settings that produce the slowdown. A successful run on another PC does not exclude a scheduling, driver or hardware-dependent issue.

1. Follow the capture steps above. For a short report, wait for loading to finish, reproduce the problem for 30–60 seconds and leave the game through **Exit game**.
2. Export the launcher diagnostic from that same session. Keep the performance `.csv` and matching `.json` together; they share a filename timestamp.
3. Note the game edition, build version, GPU driver, save/route and whether VSync, an FPS limit, acceleration or an external overlay was enabled. Describe where the slowdown starts.
4. Attach the diagnostic and both performance files to the issue. Keep an original copy. Captures remain local until you choose to share them.

Report evidence	What it tells you	Interpretation limit
`frame_ms` and stall counts	Frame intervals and counts over 100/250/500 ms.	Loading/menu activity may also produce long intervals.
`acquire_resize_ms`, `compose_ms`, `submit_ms`, `swapchain_present_ms`, `retire_ms`	CPU wall time in successive presentation stages.	These are not GPU execution timestamps; acquire includes resizing, and present includes the submit-lock wait.
`outside_present_ms`	Frame interval minus the current Present call.	Includes uninstrumented work, scheduling and pacing; does not prove a guest-code bottleneck.
`runtime_mutex_ms`, `runtime_condition_ms`, `runtime_semaphore_ms`, `runtime_join_ms`	Instrumented calls completed during that interval, summed across threads.	A call can begin in an earlier frame. Concurrent waits overlap; never subtract their sum from frame time. Other wait types are not measured here.
`effective_provider_id`, `effective_preset_id`	Provider/preset carried by the queued frame after reconstruction.	Requested settings can legitimately differ during a fallback or transition.
`requested_provider_id`, `requested_preset_id`	Settings selected at the time of capture.	A request is not proof that a provider ran.
`output_width`, `output_height`	Actual swapchain output size.	Distinct from internal scene resolution.
`render_width_sampled`, `render_height_sampled`	Applied internal-size setting sampled at presentation.	Not a measurement of the queued frame's individual image resources.

Report evidence	What it tells you	Interpretation limit
`usage_interval_ms`, `usage_age_ms`, `usage_collection_ms`	Measurement interval, reading age and sampling cost.	Hardware/thread usage is sampled about every 500 ms, not once per frame. Negative values mean unavailable.
JSON `stall_context.windows`	CSV frame ranges and thread snapshots around severe stalls.	Snapshots are taken after the worst observed stall, not at its blocked instruction.

Stable ID	Provider
0	Reconstruction off.
1	AMD FSR 3.1.5.
2	AMD FSR 4 v7 (INT8).
3	Legacy FSR 4.1.1 path.
4	Temporal AA path.
5	DLSS.
6	Official AMD FSR SDK 2.4 integration; see the effective provider name in the overlay.

Preset IDs are **−1** for the original fixed-resolution scene, **0** Native, **1** Quality, **2** Balanced, **3** Performance and **4** Ultra Performance. Provider 0 means no active reconstruction preset regardless of the accompanying numeric preset field.

Each context refers to the CSV's `frame` column, with approximately two seconds of history and follow-up. History retains at most 240 preceding frames. Overlapping events merge; up to 32 contexts are retained. Each worst-frame snapshot includes up to eight sampled busy threads and up to 256 lifetime runtime-thread registrations. JSON reports dropped events or registrations explicitly. A runtime slot distinguishes registrations when Windows reuses a thread ID. `runtime_exit_reported` records a runtime exit notification, not an OS liveness check. Object IDs are session-local correlation values; they do not identify the thread owning a lock.

> **WATCH OUT**
>
> **This is evidence collection, not a slowdown fix.** The instrumented wait paths use fixed storage and are disabled outside capture. Capturing still has a cost and the CSV timeline grows with session length: prefer short reproductions. Reports finalize on normal exit; killing the process or a hard crash may prevent final export. Unobserved calls and exact instruction stacks require a separate trace.

If the bottleneck remains outside the measured stages, use the optional capture-windows-cpu-trace.ps1 on the affected PC. It starts the Windows CPU recorder, waits for the game and records a short trace. Windows may ask for administrator access. Reproduce the slowdown during recording, then share the resulting trace through an appropriate private channel if it contains sensitive system details. Keep the matching executable/debug symbols for analysis. Do not infer a specific FSR/DLSS, CPU or driver fault from a low aggregate usage reading alone.

CHAPTER /

Known issues and troubleshooting

Symptom	First action
Missing DLL or executable	Extract the complete player folder again; keep the real applications and DLLs in `bin`.
Wrong or empty game information	Select the extracted game directory itself and wait for validation.
A rendering preset is refused	Return to previous working preset; see the experimental preset limits. Export diagnostics after the refusal.
Pixelation, a bottom-edge line or a black scene	Note the active provider, preset and applied input size. Compare the same route with reconstruction disabled, then export diagnostics. Do not assume FXAA fixes a reconstruction failure.
Fidel stops following another leader	Restore Fidel as leader. Location-dependent follower failures remain a known issue in 1.1.3.
HUD preset has no effect	Refresh the library, check game/ratio compatibility and the apply status, then test a mapped group. Subtitle adjustment remains incomplete.
Title-screen globe remains 16:9	This scene still has a separate aspect-ratio limitation; do not stretch the complete UI/video presentation to compensate.
Temporary 16:9 framing at a zone boundary	Some transition states can still restore the game's original framing briefly. This is separate from the full-window loading blackout.
FSR is slower on older hardware	Compare with Upscaling disabled, keeping the same scene and output resolution.
Saves are not found	Check the displayed save path, Title ID and save-slot folder structure.
Backups cannot be replaced	Close the launcher before replacing completed backup snapshots.
No downloadable update	A tag is insufficient; a newer compatible Windows player Release asset and its digest are required.

- Save progress in the game's own save system before using **Exit game**. Closing the custom settings menu or the launcher is not a command to save game progress.

CHAPTER /

Advanced appendix: AMD FSR SDK 2.4

> GOOD TO KNOW
>
> For advanced readers. You do not need this chapter to play. It explains why**AMD FSR 4.1.2** may appear, why a request can fall back, and which components the Windows player expects.

Selecting the provider

Choose **AMD FSR (SDK 2.4)** under launcher Video or the in-game Image tab. The existing FSR 3.1, FSR 4 v07 and DLSS choices keep their own implementations. The new provider does not replace source-v07 models or require new trained model files.

The launcher checks the selected GPU policy and installed components. The runtime then queries AMD's Vulkan API for a compatible implementation and validates context creation and dispatch. The performance readout reports the actual active provider after successful initialization. SDK version and reconstruction version are different: the SDK 2.4 package used here reports FSR 4.1.2 on the development RX 9070 XT. Other devices/drivers can select a different supported implementation or reject the provider.

Native (100%), Quality, Balanced, Performance and Ultra Performance request scene input sizes through the existing preset system. Original 1080p remains subject to output-size checks. Outputs include 4:3, 16:10, 16:9, 21:9 and 32:9; the selected scene projection and device limits still apply. Reconstruction does not replace the game's original HUD textures.

The existing FSR 3.1 default GPU policy also gates this choice. The lower-tier override can relax that recommendation but cannot bypass missing Vulkan features or the SDK's compatibility checks. A rejected provider falls back to the admitted FSR 3.1 path, or disabled reconstruction, and logs the reason. Check the active-provider readout.

Distribution and build integrity

Provider files live under `gpu/third_party/amd-fsr-sdk24/`. The packaged application loads only `bin/amd-fsr-sdk24/` beside the real game executable, without searching the working directory or PATH. Missing optional provider DLLs do not prevent application startup.

The signed AMD Vulkan loader/upscaler DLLs are distributed unchanged, together with their app-local Microsoft Visual C++ x64 runtime DLLs. No separate Microsoft runtime installation is needed for this provider. Existing system GPU driver requirements still apply.

CMake and `scripts/verify-amd-fsr-sdk.ps1` verify SHA-256 hashes against `binaries.json`. The normal Windows build and packaging scripts copy the provider folder. Packaging includes the AMD and Microsoft redistribution notices and an inventory of the shipped DLL versions and hashes. Keep the provider directory and notices together when distributing the build.

Upstream reference: AMD FSR SDK. This change adds upscaling through Vulkan. It does not add frame generation, XeSS or a DirectX renderer, and it does not change the existing DLSS compatibility gate.

Verification and remaining gameplay checks

The isolated GPU test initializes the real official provider and checks 120 output/preset/input-format combinations (240 dispatches): eight output sizes across five aspect-ratio families (including 3440 x 1440 and 3840 x 2160), with five input presets at each size and three resource configurations. These cover float-color depth, native D32 depth, native D32/stencil depth, reversed depth, HDR scene color and 8-bit SDR color. It reads back all output pixels and rejects non-finite values or a black center sample. Each combination includes a reset frame followed by temporal continuation without a CPU queue wait. Missing DLLs and invalid input dimensions are rejected before valid dispatch. Settings persistence and unavailable-provider fallback are checked separately.

The official SDK creates image views from the supplied Vulkan images. The integration supplies actual resource formats, depth/stencil usage and matching barrier aspects, rather than declaring every input as a color image. HDR and SDR paths use separate context flags; changing between them recreates the context. Unsupported formats are rejected before dispatch and use the existing fallback. Input formats and color space are logged when a context is created.