

# Godot on the Web

Current status and future plans

# Godot 4 Release

- Godot 4.0 was released on March 2023
- In the process of releasing Godot 4.2
- Huge modernization of the engine
- Heavily relies on threads
- New Vulkan renderer
- GDExtension (binary addons)
- Lots of new features (Animations, Asset pipeline, Multiplayer, etc)

# Challenges - Rendering

- WebGPU is not ready yet
- Shading language is still being worked on
- Our abstraction (Vulkan/D3D12/Metal) doesn't fit WebGPU

# Consequences - Rendering

- People that needs to target mobile devices prefers Godot 3.x due to issues with WebGL2 on iOS.
- Not able to reap the performance benefits of WebGPU.

# Challenges - Threads

- Support is still not great
- Require secure context and COOP/COEP
- WebXR only works inside specific callbacks which can't be easily accessed from a different thread

# Consequences - Threads

- Hard to monetize (ads and IAP won't work with COOP/COEP in most scenarios)
- Most people are stuck with Godot 3.x
- Couldn't run the main loop on a dedicated thread by default because of WebXR

# Challenges - GDExtension

- No standard linking metadata
- Threads + dynamic link only works in Chrome
- Everything must be built with the same tool (emcc) and version

# Consequences - GDExtension

- Most people won't use GDExtension on the web
- Mono version is still not available on the Web



# Solutions - Rendering

- Implement a GLES3/WebGL2 render
- Explore readding a WebGL1 renderer too
- Explore further refactoring the core abstraction to support WebGPU (WIP)

# Solutions - Threads

- Reintroduce the single-threaded version (TODO)
- Keep working on multi-thread (e.g. move loop to thread) and hope for better browsers support (WIP)
- (Hacky) Edit the service worker with the games we export in order to add required COOP/COEP headers when missing (TODO)

# Solutions - GDExtension

- Document which specific tool version to use
- Mention limitations and wait for browser to fix support
- Suggest the single-threaded version once we have it

# More Work

- Explore wasm64 support
- Explore Native File System API on supported browsers
- Explore more internal refactoring to support async APIs
- Working Asset Library for the Editor (requires new asset store)