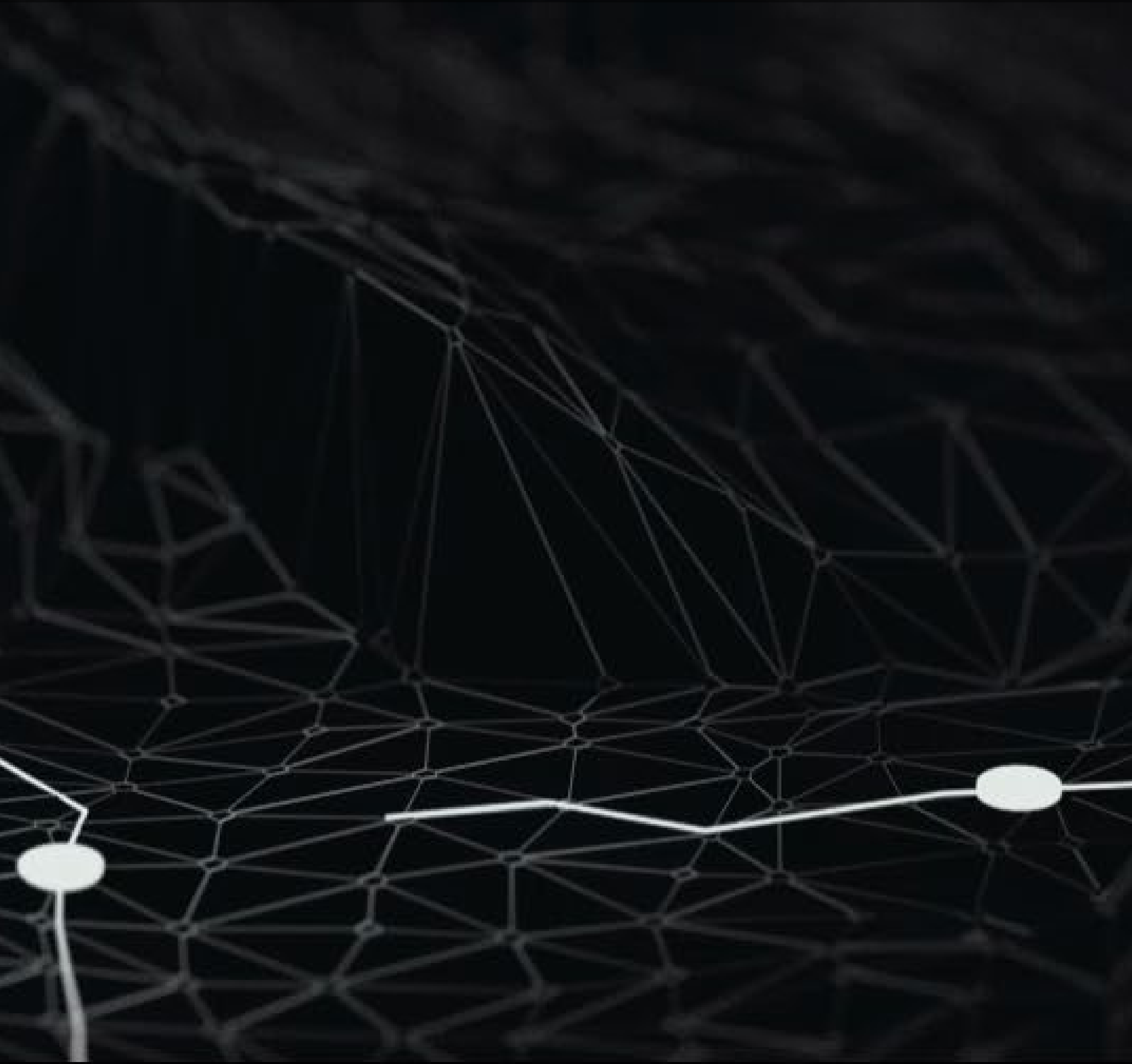
HAVOK Navigation



Havok Navigation

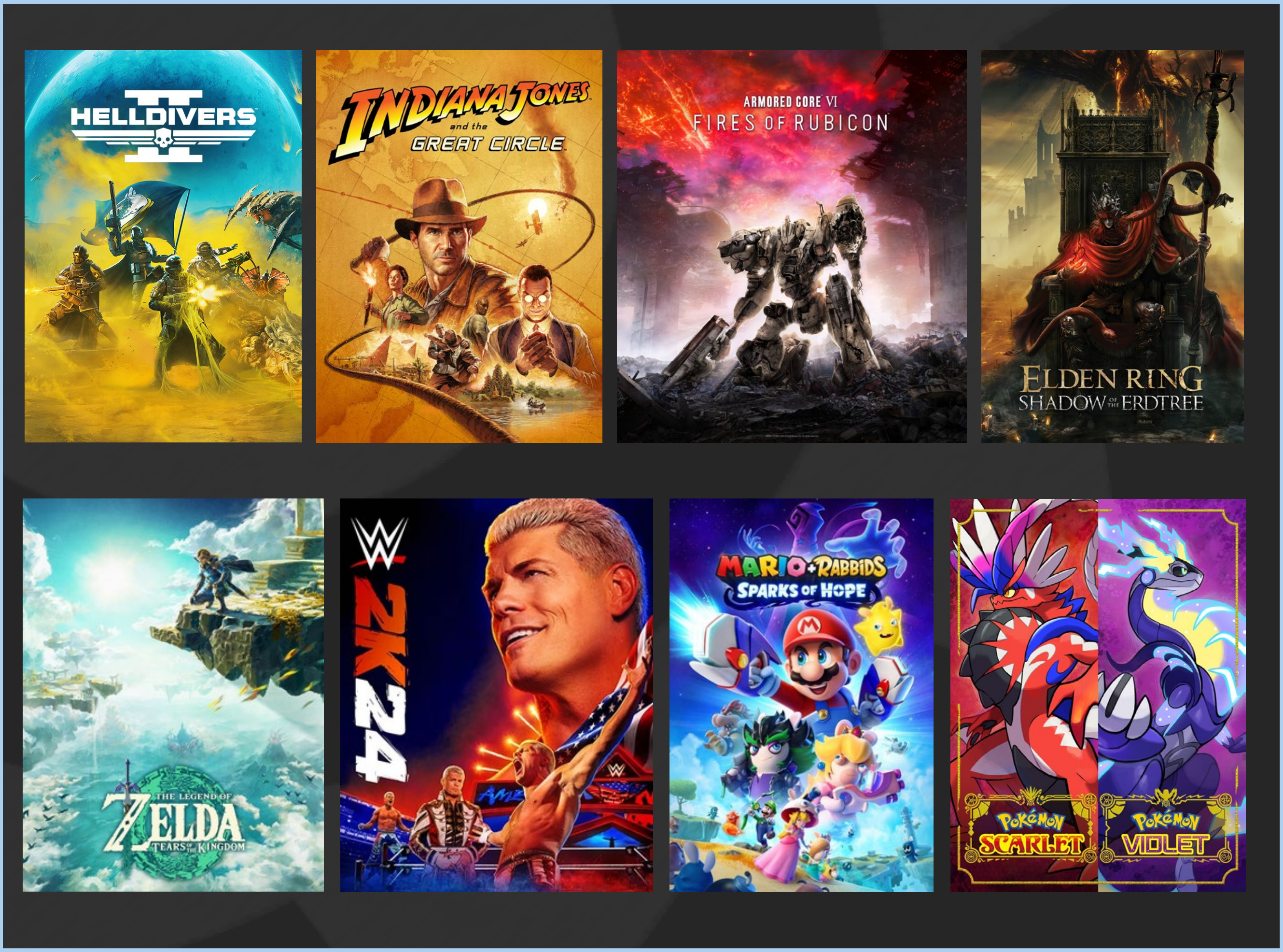
Product Overview

Havok Navigation equips your NPCs to explore rich dynamically changing worlds. Havok Navigation is a C++ SDK providing building blocks to integrate into your character systems in creative ways. This allows you to build custom navigation behaviors that react to dynamically changing environments on a robust, battle-tested foundation.

Used by the biggest game franchises in the world like The Legend of Zelda, Doom, Elden Ring, Pokémon and many more, Havok Navigation is a proven solution across a wide range of game genres.

Create more immersive and dynamic player experiences, by leveraging Havok Navigation in your game.

Examples of games powered by Havok Navigation


HELLDIVERS II
INDIANA JONES and the GREAT CIRCLE
ARMORED CORE VI FIRES OF RUBICON
ELDEN RING SHADOW OF THE ERDTREE
THE LEGEND OF ZELDA TEARS OF THE KINGDOM
W 2K24
MARIO + RABBIDS SPARKS OF HOPE
Pokémon SCARLET
Pokémon VIOLET

Havok Navigation

Navigation Data Generation

Fast, high-quality Navigation Meshes: Havok's navigation mesh generator uses a custom rational arithmetic-based number representation to create watertight nav meshes that perfectly align with your level boundaries.

Traversal Analysis: Automatically place edges connecting nav mesh faces based on character specific traversal movement sets. For example, automatically create links connecting platforms your characters can jump between.

Customizable generation: Use callbacks during nav mesh generation to inject custom information into the nav mesh, like marking individual faces with custom materials based on query results.

3d Navigation: Create navigation volumes to represent traversable areas in 3d. This can enable flying or swimming characters for example.

Cross-Platform Determinism: Our navigation data generation algorithms are fully cross-platform deterministic allowing you to create identical nav meshes and volumes independent of the hardware of your build farm. If your game uses procedurally generated content, this can ensure identical nav meshes regardless of the platform the game runs on.

Pathfinding and Queries

Customizable, efficient pathfinding: Full control over search parameters such as cost control and edge traversal. Multithreaded distribution of pathfind requests.

Multi-Radius navigation: Our clearance system allows you to navigate characters with varying radii on a single navigation mesh.

Hierarchical pathfinding: Cluster graphs add a layer to the nav mesh, allowing accelerated pathfinding to sections not yet loaded.

Collision avoidance: Havok's collision avoidance system supports large crowds avoiding moving obstacles and other characters with emergent effects like lane formations and swirling. Designed and tuned to work well with animation-driven locomotion systems.

Dynamic Environments

Runtime stitching: Load and unload sections of the nav mesh at runtime to only keep relevant parts of the world in memory. Our runtime stitching feature allows connecting neighboring cells even if they don't perfectly align.

Silhouette cutting: Remove parts of the nav mesh at runtime based on an overlapping query object. For example, remove an area from the nav mesh based on an overlapping physics object blocking traversal.

Silhouette painting: Alter the nav mesh based on dynamic, player-controlled events. For example, increase the cost of traversal of cells overlapping with the impact area of a water balloon creating mud.

Runtime addition and removal of custom edges: Add user edges to the nav mesh to enable new traversal dynamically at runtime. Represent moving platforms by enabling/disabling these edges based on the state of the platform.

Edge and face filtering: Dynamically adjust the cost and traversability of individual navigation mesh faces and edges. Create one-way doors or nav mesh faces with varying traversal cost.