



AN ACTION-ADVENTURE FOR PC | UNREAL ENGINE 5.8 | VERTICAL SLICE PLAYABLE

ONLY HOPE

Mars is dead. You are the last pilot with a working suit - and flight, hover and wingsuit glide are how you cross it, and how you fight it.

The Prodigal Developers

Olongapo, Philippines | theprodigaldevelopers.com

EPIC MEGAGRANTS

Request: \$111,000 | 18 months

A Mars-set action-adventure built in UE5 by a small Filipino team - flight, wingsuit traversal, and boss combat across an open Martian expanse. Proof that world-class Unreal games can be made in the Philippines.

GENRE

Third-person
action-adventure

ENGINE

Unreal Engine 5.8
Mover 2.0, Lumen, Nanite

PHASE

Pre-production
vertical slice playable

PLATFORM

PC first
console post-launch

Self-funded to date. No outside funding secured.

— PREMISE

The colony already failed. You arrive after.

Only Hope is a third-person action-adventure set on a Mars whose colonization effort has already collapsed. You play the last pilot with a working suit, moving across an abandoned planetary surface to find out what happened to the colony - and what is still living in it.

The planet is the antagonist before anything else is.

[SCREENSHOT]

Martian expanse - wide vista

[SCREENSHOT]

Colony ruin

[SCREENSHOT]

Pilot suit

Traversal. Combat. Discovery.



PILLAR 01

TRAVERSAL

Powered flight, stationary hover, and an unpowered wingsuit glide with real lift and drag. Air tunnel rings grant boost and turn open distance into a readable, skill-expressive route.

Movement is not how you reach the content
- it is the content.

PILLAR 02

COMBAT

Encounters are built around large Martian creatures rather than volume. The slice ships the Martian Alpha, a boss that forces the player to use the full traversal kit under pressure.

The flight and glide systems are the combat systems.

PILLAR 03

DISCOVERY

A holographic scanning layer reads the colony's history off the environment - research logs and records recovered from the sites where they were left.

The story is assembled by the player, not delivered in cutscenes.

Most Mars games are walking simulators with a red filter.

Our answer to a vast, empty world is to make crossing it the pleasure, not the tax.

- ◆ **Powered flight.** Full directional control with thrust and fuel response.
- ◆ **Hover.** Stationary hold for combat repositioning and precision approach.
- ◆ **Wingsuit glide.** Models lift and drag, not a fall-speed clamp - trade altitude for speed and hold a line across kilometres.
- ◆ **Air tunnel rings.** Placed boost gates that give an empty planet a shape to move through without clutter.

[SCREENSHOT]

Glide over canyon / air tunnel ring

The Martian Alpha



[SCREENSHOT]

Martian Alpha boss encounter

The vertical slice's headline encounter. A large Martian creature that cannot be fought from the ground - the player has to stay airborne, read attack tells while moving, and convert glide lines into openings.

Three distinct attack phases, a full health state machine, and a death sequence, shipping in the slice.

Design rule: every combat system is a traversal system used under pressure. We do not add a separate combat verb set - we raise the cost of using the one the player already loves.

This already runs.



A playable vertical slice in Unreal Engine 5.8. Implemented and working:

- ◆ **Custom locomotion stack on Mover 2.0** - flight, hover, wingsuit glide, ground locomotion, slide and crouch, with correct transitions between every mode.
- ◆ **Mover-to-GASP animation bridge** - including the chooser-driven pose selection the Game Animation Sample Project depends on.
- ◆ **Large-scale Martian landscape** - produced by our own offline generation pipeline and validated against walkability metrics rather than by eye.
- ◆ **The Martian Alpha boss encounter** - three attacks, health state machine, death sequence.
- ◆ **Air tunnel traversal and the holographic discovery layer** - both playable in the slice.
- ◆ **Lumen GI and reflections, Nanite geometry, Chaos physics** - throughout.

Two problems with almost no public reference implementation



PROBLEM 01

AERIAL LOCOMOTION ON MOVER 2.0

Unreal's Mover framework ships with no flight, hover, or glide movement mode. We implemented all three as first-class Mover modes, with the transition logic that makes them interoperate cleanly with ground locomotion.

To our knowledge there is no other public implementation of aerial movement on Mover.

PROBLEM 02

BRIDGING MOVER TO GASP

GASP's chooser-driven animation stack assumes the Character Movement Component. Running it on a Mover backend means reconstructing the character-state snapshot the AnimGraph reads and matching chooser context classes exactly.

This is undocumented, and it is the exact wall that stops most developers who try to combine the two.

Spec-driven planetary terrain

Our Martian landscape is not hand-sculpted. It is generated offline from a written specification and validated automatically against walkability metrics.

Terrain changes are made by editing the spec and rebuilding - not by manual sculpting passes. That is what makes a planet-scale surface maintainable by a team of three.

01

SPEC

Terrain written as a text specification

02

BUILD

Deterministic offline generation

03

VALIDATE

Automated walkability metrics

04

ITERATE

Edit the spec, rebuild, re-measure

What we release to the community

Released under a permissive license and funded as a deliverable - not as a promise we make room for later.

RELEASE 01

MOVER SAMPLE PROJECT

A standalone drop-in reference containing our flight, hover and glide Mover modes plus the Mover-to-GASP animation bridge.

RELEASE 02

TERRAIN PIPELINE

The spec-driven offline landscape generation and walkability validation system, open sourced.

RELEASE 03

TECHNICAL WRITE-UPS

Published breakdowns of both systems, covering the failure modes we hit so others do not have to rediscover them.

Plus open technical sessions for Philippine university game development programs and local developer meetups.

— WHY THIS MATTERS HERE

The Philippines builds other people's games.

The Philippine games industry is substantial and growing - and overwhelmingly outsourcing and co-development for foreign studios. Very few Filipino teams direct their own original titles. Fewer still attempt one at this level of technical ambition on Unreal Engine.

A funded, shipped, technically serious Filipino-owned Unreal title changes what the next team here believes is available to them.

That is why we build publicly and release our technology openly.

<5

person studio

\$0

outside funding to date

3

systems to open source

18

months to ship

What the grant unlocks

The vertical slice proves the game works. It does not prove we can build eight hours of it. This funds the transition from pre-production to full production.

REQUESTING

\$111,000

Band: \$100,000 - \$150,000

18-month runway, 3 full-time Filipino developers

- ◆ **Deliverable.** A finished, commercially released title.
- ◆ **Plus.** The open-source locomotion and terrain technology, released alongside it.
- ◆ **Specialists.** Score, sound design, voice and creature art we cannot staff internally.

\$111,000 over 18 months



Core team - 3 FTE, Philippines rates

\$70,200 63%

Lead gameplay engineer \$27,000 / Gameplay + technical animation programmer \$21,600 / Environment and 3D artist \$21,600

Specialist contract work

\$21,000 19%

Original score and sound design \$10,000 / Voice + EN-FIL localization \$4,000 / Creature art, two additional bosses \$7,000

Production infrastructure

\$10,300 9%

Workstations and capture hardware \$6,000 / Perforce, build and cloud hosting \$2,700 / DCC and audio licenses \$1,600

Community release and launch

\$9,500 9%

Open-source packaging and write-ups \$3,500 / Festivals and industry events \$3,500 / Store assets and trailer \$2,500

The largest line is people. This converts a self-funded side effort into three salaried full-time Filipino game developers.

18 months to release

MONTH 1-4

MONTH 5-10

MONTH 11-15

MONTH 16-18

TEAM UP AND SYSTEMS

Hire the core three. Harden the locomotion stack. Lock the combat verb set and the enemy framework.

CONTENT PRODUCTION

Full Martian expanse build-out. Two additional bosses. Complete the discovery and narrative layer.

CONTENT COMPLETE

Score, sound design and voice integrated. Localization. Full playthrough tuning.

POLISH AND SHIP

Performance, certification, launch. Open-source release and technical write-ups published.

Milestone gate at each phase boundary: a playable build, not a document.



We are not asking Epic to fund an idea.

We are asking Epic to fund the production of something that already runs.

STUDIO

The Prodigal Developers

WEB

theprodigaldevelopers.com/only-hope

BUILD

the-prodigoal-developers.itch.io/only-hope

LINKEDIN

linkedin.com/company/the-prodigoal-developers