



IHIYO

CSE

Adam Lei, Edwin Sotres, Terrence Akinola

Sara Davis, Vinh Le, David Feil-Seifer

External Advisors: June S. Saraceno

Department of Computer Science and Engineering, University of Nevada, Reno

ABSTRACT

Ihiyo is a Top-Down 2D puzzle game developed in the Unity Engine Software that delivers a story-rich gaming experience for all players. As the protagonist, *Ihiyo Nali*, the player possesses the ability to turn into a ghost on a moment's notice to complete a wide variety of unique puzzles. Phase through walls, possess different objects—anything is possible. *Ihiyo* is critical for players looking to engage with a narrative journey that leaves them more profoundly insightful about the limited time they have in this dwindling world.

CONCLUSION

Ihiyo at the current state has core features including character movement, NPC interactions, dialogue systems, and quest frameworks. While the gameplay loop and puzzles remain in development, there is an established, solid foundation for the narrative experience. This project demonstrates how cultural storytelling can be meaningfully integrated with puzzle mechanics to create experiences for all players.

DESCRIPTION

Main Goals & Features

Players guide Ihiyo Nali, a boy who can shift between physical and ghostly forms, using unique abilities like wall-phasing and object possession to solve puzzles. *Ihiyo* blends logic-based challenges with emotional storytelling, encouraging reflection on mortality and spiritual legacy.

Core Gameplay Systems

- Dialogue & Non-Player Character (NPC) Interaction: Branching dialogue with interactive NPCs driving narrative progression (custom node-based system using Unity's ScriptableObjects).
- Inventory & Pickup: Item collection tied to puzzle-solving and exploration, managed via custom C# components.
- Quest System: Tracks objectives and guides player progression, implemented with event-driven scripts.

Persistence Features

- Save/Load: Three save slots with independent progress tracking, handled via PlayerPrefs and custom JavaScript Object Notation (JSON) save files.
- Key Bindings: Customizable controls persist across sessions using Unity's input system and save/load functionality.

UI & Menus

- Functional main/pause menus with integrated settings and key remapping, created using Unity's UI Toolkit and custom scripts.

FUTURE WORK

- **Expansion** into a full title with an immersive overarching narrative. Aim for Steam release
- **Transform** into a tech demo to experiment with different mechanics and systems
- **Expand** existing concepts and mechanics to fully realize their potential within the scope of a small project

SNAPSHOT



Figure: A gameplay screenshot showing a quest initiation. The player speaks to "Dad," triggering a "NEW QUEST" popup and choosing between two dialogue options to retrieve his lost wallet.