Milestone 5: Tech Demo

Project Hidden Threads by Team Midnight Umbrella

https://itsraindi.itch.io/hidden-threads

Milestone 5: Tech Demo

Art

---

i. **What kind of art style did you intend to create, and is it consistent and thematically appropriate?**

The intended art style for Hidden Threads is a semi-realistic 2D narrative mystery style with a dark, grounded atmosphere. The environments are designed to feel believable and realistic, such as apartments, bars, alleyways, and abandoned buildings, to support the investigative and suspense-focused narrative.

The color palette uses muted and darker tones, such as greys, blues, and shadows, to reinforce feelings of tension, uncertainty, and isolation. Brighter colors are used sparingly to highlight important objects, such as Vera's phone or key clues, drawing the player's attention to narrative-relevant elements.

The character designs are more realistic and simple rather than exaggerated, helping players emotionally connect with Harper and making the story feel more immersive. This art style is consistent and thematically appropriate because the game focuses on mystery, disappearance, and uncovering hidden truths.

ii **What technologies must be tested to ensure your art fits within the game properly? Are they in the tech demo?**

The following technologies must be tested to ensure the art integrates correctly:

- Unity's Sprite Renderer for displaying characters, objects, and environments
- Unity's Canvas UI system for dialogue boxes and interactive UI elements
- Proper sprite layering and sorting order to ensure objects render correctly in front of or behind others
- Scene layout and positioning using Unity's Transform and Scene Editor
- Resolution scaling to ensure art displays properly on different screen sizes

Some of these technologies are already present in the tech demo, including:

- Dialogue UI panel and text box
- Scene environment setup
- UI canvas integration

Additional art assets such as final characters, environments, and animations will be added in later stages

iii. **Will your game include cutscenes? Have any of these been produced? What technology will be needed for these?**

Yes, the game will include simple narrative cutscenes, especially at key moments such as:

- The introduction scene where Harper realizes Vera is missing
- Important discoveries, such as unlocking Vera's phone
- The confrontation with Mr. Crocker
- The game endings

These cutscenes will be implemented using:

- Unity's Canvas UI system for dialogue and narrative text
- Character sprites appearing and disappearing on screen
- Scene transitions
- Dialogue progression using the DialogueUI system already implemented

The tech demo currently includes the dialogue system, which is a core technology needed for cutscenes

---

Sound

---

i **Does your game include a soundtrack and/or sound effects?**

Our game features basic sound effects at this stage and framework is currently implemented for background music during gameplay events and menu sequences.

ii **What technologies must be tested to ensure your sound works properly within the game? Are they in the tech demo?**

The following Unity technologies must be tested:

- Unity's AudioSource component for playing sounds
- Unity's AudioListener
- Triggering sounds through scripts when events occur
- Triggering sounds/sound effects for when player clicks/interacts with objects throughout the game
- Volume balancing and looping settings

Sound integration is not fully implemented yet. The tech demo currently focuses on dialogue and UI systems key to gameplay. Sound Design later will focus on creating tense, atmospheric ambience for the player while they solve mysteries.

Writing/Narration

i **What is the narrative and how is it evoked (ex. through writing, atmosphere, characters, setting, etc.)?**

The narrative follows Harper, a journalism student investigating the disappearance of her friend Vera. The story unfolds as Harper explores different locations, examines clues, and uncovers evidence suggesting Vera was involved in investigating a dangerous corporate conspiracy.

The narrative is evoked through:

- Dialogue and internal monologue
- Environmental storytelling, such as objects, notes, and Vera's phone
- Character interactions, including the bartender, Louise, and Mr. Crocker
- Atmosphere created through lighting, art, and sound
- Player exploration and discovery of clues

The narrative creates suspense and emotional investment by gradually revealing information and increasing tension

ii **What technologies are needed for the writing/narration? Are they in the tech demo?**

The following technologies are needed:

- Unity Canvas UI system for dialogue display
- Custom DialogueUI script for showing dialogue text
- Dialogue data structures (string arrays) to store dialogue lines?? (Check this)
- Script logic to advance dialogue when the player presses a key

These technologies are already implemented in the tech demo, including:

- Dialogue UI panel
- Dialogue text display

This allows the game to present narrative content and supports future cutscenes and character interactions.

Programming

---

i **What core mechanics must be implemented (ex. player movement, puzzle logic, etc.) in the game?**

A. Player momevent + Camera

Movement (WASD / controller), turning the camera, and collision with the environment

B. Interation System

A consistent way to interact with objects/NPCs (raycast + “Interact” key, or trigger zones)

Interaction targets: doors, clues/items, phone, puzzle objects, NPC talk trigger (TBD)

C. World/Scene Flow

Title Screen → Game scene (already), which include scene transitions (Real World ↔ other scene/world, or sub-areas) with clean return paths

Spawn points / player positioning when you load scenes

D. Complex system of interactable UI screens for phone app management

Phone / Social-media UI loop (open apps, navigate screens, back/home logic)

HUD elements (prompts, objective text, maybe inventory/journal button)

Settings UI (volume already has code patterns in your repo; needs integration + persistence)

E. Puzzle / progression logic

Rules that define “what counts as progress” (collect X clue, complete Y action, trigger Z state) and Gating: unlocking areas, new dialogue, new phone posts, “transition allowed now” checks.

F. Inventory/journal

Track collected clues/materials/notes

Display them in a UI panel (list/grid)

Save/load (at least minimal persistence)

G.Cutscene / intro delivery

Opening cutscene that reliably plays in your target platform (WebGL/itch.io)

ii **Does your game include NPCs? If so, what is their intended behavior?**

Yes, our game includes NPCs that serve as key narrative elements.

The intended NPC behaviors include:

- Idle presence in the world: NPCs remain in fixed positions within the environment to create a believable and interactive game world.
- Player interaction: When the player approaches and presses the interaction key, the NPC triggers a dialogue sequence.
- Dialogue delivery: NPCs present dialogue stored in an external CSV file planned
- Narrative progression: NPC dialogue helps communicate the story, world background, and emotional tone of the game. It also helps the player make choices that determine the outcome and ending of the game.

For the current demo, NPCs do not move or branch dialogue, but they successfully demonstrate interaction and narrative functionality.

iii **What technologies are needed to ensure your core mechanics and NPCs work properly? Are they in the tech demo?**

Technologies needed (Unity)

Core mechanics

- **Input handling** (Unity old Input Manager or the new Input System)
  → movement, camera look, interact key
- **Physics + collision** (Colliders, Rigidbody/CharacterController, triggers, raycasts)
  → walking into walls, detecting interactable objects, puzzle triggers
- **Scene management** (`UnityEngine.SceneManagement.SceneManager`)
  → Title → Game flow, world/area transitions, respawn points

- **UI system** (Canvas, **TextMeshPro**, Buttons, Panels, EventSystem)
  → phone/social-media UI, dialogue UI, HUD, settings menus

- **Game state/progression tracking** (C# manager + flags/enums, ScriptableObjects, or event system)
  → “puzzle solved?”, “clue collected?”, unlock next step, change NPC dialogue

- **Audio system** (AudioSource/AudioListener + mixer optional)
  → SFX, ambience, volume settings persistence

WebGL cutscene

Video playback pipeline (VideoPlayer + RenderTexture) *or* **image-sequence slideshow** fallback
→ because WebGL video can be unreliable depending on codec + browser autoplay rules

In this case, we tried the image-sequences slideshow, 12 frames gonna played in 3 seconds.

Ideally, we will change the cutscene after the tech demo to make it much smoother and better.