# **Ontology Development Report: Whispers of the Clocktower Game**

This report details the process of developing an OWL (Web Ontology Language) ontology for the "Whispers of the Clocktower" game. My primary objective was to create a formal, machine-readable representation of the game's core entities, mechanics, and relationships, based on the provided game design document. This ontology serves as a structured knowledge base, enabling detailed analysis and querying of the game's intricate systems.

## **1. Game Core and Ontological Mapping**

The "Whispers of the Clocktower" game revolves around a player navigating various locations, collecting items, interacting with NPCs, solving puzzles, and managing game state variables like time and NPC trust to achieve one of several endings.

The process of mapping these game elements to an ontology involved defining:

### **Classes**

These represent the types of entities in the game.

* Location: Places the player can visit (e.g., Plaza, Library).
* Item: Objects the player can acquire or use (e.g., Oil Can, Brass Emblem).
* NPC: Non-Player Characters with whom the player can interact (e.g., Iris, Garrick).
* Puzzle: The challenges or tasks the player must complete (e.g., Book Ordering Puzzle, Forging Sequence Puzzle).
* PuzzleFlag: Boolean states indicating the completion of a puzzle or a specific game condition (e.g., booksSolved, gateUnlocked).
* Player: The entity controlled by the user.
* Ending: Represents the possible outcomes of the game (e.g., sunriseVictory).
* Barrier: A general class for obstacles, with a specialized subclass, TrustBarrier, for obstacles that depend on an NPC's trust level.
* EmotionalState: A class representing the various emotional states an NPC can have.

### **Properties**

The ontology defines object and data properties to establish relationships and attributes between these entities.

#### **Object Properties (Relationships):**

* Item-centric properties:
  + :foundAt (Item → Location): Indicates where an item is located.
  + :isAcquiredBy (Item → Player): Indicates which player acquires an item.
  + :unlocks (Item → Location or PuzzleFlag): Indicates what an item can unlock.
* NPC-centric properties:
  + :associatedWith (NPC → Item): Links an NPC to a specific item.
  + :influencesItemAvailability (NPC → Item): Specifies which items an NPC can influence access to.
  + :providesHintFor (NPC → Puzzle or Location): Indicates what an NPC can provide a hint for.
* Player-centric properties:
  + :hasInventory (Player → Item): Represents the items a player possesses.
  + :currentLocation (Player → Location): Indicates the player's current position.
* Puzzle & Location-centric properties:
  + :isLocatedIn (Puzzle → Location): Specifies where a puzzle takes place.
  + :requiresItem (Puzzle or Location → Item): The item(s) needed for a puzzle or to access a location.
  + :setsPuzzleFlag (Puzzle → PuzzleFlag): The flag set upon puzzle completion.
  + :grantsRewardItem (Puzzle → Item): The item(s) granted as a reward for a puzzle.
  + :hasNextTrigger (Location → PuzzleFlag): The puzzle flag required to proceed from a location.
* Game State & Ending properties:
  + :determinesOutcome (PuzzleFlag → Ending): Indicates how puzzle flags influence the game's ending.

#### **Data Properties (Attributes):**

* :hasName (owl:Thing → xsd:string): A name for any entity.
* :hasUse (Item → xsd:string): A description of an item's function.
* :hasPersona (NPC → xsd:string): The personality of an NPC.
* :hasCoreAttitude (NPC → xsd:string): The core attitude of an NPC.
* :hasNeutralHint, :hasGoodHint, :hasBadHint (NPC → xsd:string): Dialogue options based on trust.
* :hasTriggerCondition (PuzzleFlag or Ending → xsd:string): The logical condition for a flag or ending.

**2. Ontology Development Process and Problem Solving**

The development of this ontology was an iterative process. To be honest, I gave crucial initial information regarding all connections, classes, and properties, which formed the foundation of the ontology. Following this, Gemini performed a review and refinement phase where logical inconsistencies and areas for increased precision were identified and corrected.

Several key issues were detected by Gemini during review:

**Problem 1: Incorrect Domain/Range on owl:topObjectProperty.**

Gemini identified a critical logical error where I had accidentally assigned specific domain and range restrictions (Location and PuzzleFlag) to owl:topObjectProperty — the most general object property in OWL. This caused a misunderstanding of the intended logic in OWL. These restrictions were actually meant for the specific property: hasNextTrigger, and I have since corrected the mistake.

**Problem 2: Refining the Barrier Concept.**

I forgot to include the Barrier class in the ontology, which represents the trust score. If a character trusts the player, the barrier is removed. Once removed, the character can speak freely and reveal all necessary information.

I have now added the Barrier class and a subclass called TrustBarrier. The Barrier class is a general concept for obstacles in the game. The subclass, TrustBarrier, is used to specifically represent an obstacle that is directly tied to an NPC's trust level. The ontology will now use properties like isAffectedByTrustOf and requiresTrustScoreGreaterThanOrEqual on individuals of the TrustBarrier class to link them to a specific NPC and a required trust threshold. This allows the ontology to accurately model the game's social mechanics, where player actions influence an NPC's trust, which in turn unlocks new information or pathways.