**Group 11: Treasuring Land**

**Project Summary**

Our game, **Treasuring Land**, is an adventure game where the player explores an unfamiliar world to obtain the ultimate treasure. The game features various systems such as weather, rewards, and penalties, each affecting the player uniquely. For example, rainy days can create swamps, resurrection stones can increase the player's life, and a decreasing satiety level may invite attacks from ghosts and monsters.

**Propositions**

**- Xij:** Represents the player’s position. This proposition is True if the player is located at coordinates (i, j).

**- Tij:** Represents the treasure’s location on the map. True if there is treasure at coordinates (i, j).

**- Gij:** Represents a ghost’s presence on the map. True only if there is rainy weather and the player’s satiety level is low.

**- Exhaustion:** Indicates the player’s energy state. True if the player is alive and exhaustion is between 1 and 100; False if exhaustion is 0.

**- HP:** Represents the player’s health. True if the player is alive.

**- Mij:** Represents the map coordinates at (i, j). True only if the player is at the same location (i, j).

**- Keyij:** Represents the presence of a key at location (i, j). True if the player is at the same location as the key.

**- Weaponij: I**ndicates the presence of a weapon at location (i, j). True if the player is at the same location as the weapon.

**Constraints**

**- Treasure Location Uniqueness:** The treasure can only exist in one location at a time, preventing duplicate treasures. For example, if the treasure is at T₁₂, then ¬T₁₃ holds for all other locations.

**- Winning Condition:** The game is won when the player’s position coincides with the treasure’s location, i.e., Xij = Tij.

**Additional Constraints:**

**- Marsh Regions:** Multiple marsh regions, such as M23 ∧ M24 ∧ M25, prevent players from passing through certain areas on the map, requiring detours.

**- Weather Constraints:** Ghosts (Gij) are contingent on rainy weather and low satiety.

**- Exhaustion and Health:** If the player’s exhaustion reaches zero and health (HP) is false, it implies the player has died.

**- Item Dependency:** Some game outcomes require specific item combinations, e.g., having both Keyi` and Mij to activate certain conditions.

**Model Exploration**

**- Basic Positioning:** Implemented player and treasure positioning.

**- Weather and Effects:** Added weather system and corresponding effects on game elements.

**- Health and Exhaustion:** Incorporated health and exhaustion tracking to dynamically adjust game difficulty.

**Jape Proof Ideas**

**- Uniqueness of Treasure Location:**

Tij​→¬Tkl​ for all i≠k or j≠l

Ensures that the treasure exists at only one position on the map.

**- Winning Condition:**

Xij ∧ Tij → Win

If the player’s position (Xij) matches the treasure’s position (Tij), it implies a win.

**- Health and Exhaustion Constraints:**

If HP (health) is false or Exhaustion = 0, it implies the player’s death:  
 HP ∧ (Exhaustion=0) → Death.

**- Weather Constraints:**

The weather (e.g., rainy or sunny) affects ghost presence on the map:

(Rain ∧ ¬Exhaustion) → Gij​

**Logical Consistency Sequence:**

To show the player can win if preconditions are met, a logical sequence of conditions is used:

A1 ​∨ A 2​∨ A3 ​∨ A4​ ∨ A5​ ⇒ (Xmn ​= Tmn​)

A1, A2, A3, A4, and A5 represent each consistent game state required for a winning outcome.

**First-Order Extension**

To extend the model to a predicate logic setting, propositions and constraints would be modified to include universal quantifiers for more complex conditions, such as environmental or item-based interactions. For instance, constraints would specify that all reachable cells must satisfy certain conditions based on player position, item possession, or weather states.

**Requested Feedback**

To enhance our project, we would appreciate feedback on the following:

**- Model Constraints:** Are there any additional constraints you think would improve the consistency and challenge of the game, particularly around item interactions or environmental effects?

**- Proof Structure:** Does the current setup of our Jape proofs effectively demonstrate logical consistency, or are there improvements in the proof sequence that could strengthen our validation?

**- First-Order Extension:** Are there specific predicate logic extensions or universal constraints you suggest to make our model more comprehensive in a larger or more complex game environment?

**- Jape-Prove:** We are struggling with the jape proofs, we didn’t think of anything to prove the logical stuff.

**Useful Notation**

Feel free to copy/paste the symbols here and remove this section before submitting.

∧

∨

¬

→

∀

∃