**Bansilal Ramnath Agarwal Charitable Trust’s**

**Vishwakarma Institute of Technology, Pune-37  *(An autonomous institute of Savitribai Phule Pune University)***

******

**Title : To prepare Class diagram for a defined problem with relationships, associations, hierarchies, interfaces, roles and multiplicity indicators.**

|  |  |
| --- | --- |
| **Year** | **Third** |
| **Branch** | **AI & DS** |
| **Division** | **AI-A** |
| **Batch** | **3** |
| **PRN** | **12320092** |
| **Roll no.** | **81** |
| **Name** | **Pandit Manasi Dilip** |
| **Subject** | **Software Design and Methodologies** |

1. **Introduction**

The Treasure Hunt Game System is designed as an interactive experience where players participate in an augmented reality (AR)-based treasure hunt. The system consists of multiple classes representing different components of the game, including players, clues, leaderboards, and game environments.

**2. System Components and Class Descriptions**

**2.1 Player**

* Attributes:
  + name: String
  + score: int
  + progress: int
* Methods:
  + login(): Allows the player to log into the game.
  + startGame(): Starts the treasure hunt.
  + scanClue(): Scans the AR-based clue.
  + interactWithAR(): Facilitates interaction with AR elements.
* Relationships:
  + Participates in the Treasure Hunt.
  + Belongs to the Game Environment.
  + Updates the Leaderboard.

**2.2 TreasureHunt**

* Attributes:
  + totalClues: int
  + currentClue: int
* Methods:
  + startHunt(): Begins the treasure hunt.
  + checkProgress(): Checks the player’s progress.
  + displayFinalScore(): Displays the final score at the end of the game.
* Relationships:
  + Contains multiple ARClues.
  + Players participate in the Treasure Hunt.

**2.3 GameEnvironment**

* Attributes:
  + difficultyLevel: String
  + leaderboard: List
* Methods:
  + loadGame(): Loads the game settings and environment.
  + updateLeaderboard(): Updates the leaderboard with player scores.
* Relationships:
  + Players belong to a Game Environment.

**2.4 ARClue**

* Attributes:
  + clueID: int
  + hint: String
  + location: String
* Methods:
  + displayHint(): Displays a hint for the clue.
  + validateClue(): Validates whether the clue is correctly solved.
* Relationships:
  + Contains different types of clues: ImageClue, AudioClue, and TextClue.
  + Implements the Scannable interface.

**2.5 Scannable (Interface)**

* Methods:
  + scan(): Initiates scanning of the clue.
  + validateScan(): Validates the scanned clue.
* Relationships:
  + Implemented by ARClue.

**2.6 Clue Types**

* ImageClue: Represents a visual clue.
* AudioClue: Represents an auditory clue.
* TextClue: Represents a textual clue.

**2.7 Leaderboard**

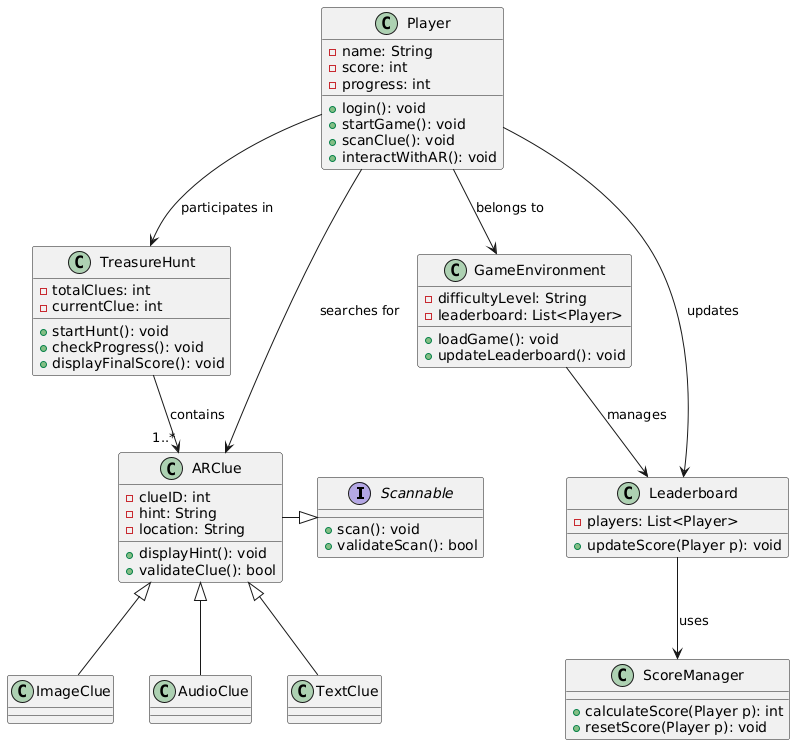
* Attributes:
  + players: List
* Methods:
  + updateScore(Player p): Updates the score of a player.
* Relationships:
  + Managed by ScoreManager.

**2.8 ScoreManager**

* Methods:
  + calculateScore(Player p): Calculates the score for a player.
  + resetScore(Player p): Resets the player’s score.
* Relationships:
  + Used by the Leaderboard.

**3. System Functionality and Flow**

1. A player logs in and starts the game within a specific Game Environment.
2. The Treasure Hunt consists of multiple ARClues that the player must solve.
3. The player interacts with and scans AR clues, which can be text, image, or audio-based.
4. The Treasure Hunt system checks the progress and validates the clues.
5. The player's score is updated in the Leaderboard through ScoreManager.
6. The game environment updates the leaderboard as the game progresses.
7. The final score is displayed once the hunt concludes.



**Conclusion :**

The Treasure Hunt Game System is structured using object-oriented principles, ensuring modularity and scalability. The integration of ARClues and different clue types enhances the gaming experience, while the leaderboard and score management provide a competitive aspect. Future enhancements could include multiplayer support and AI-driven clue suggestions.