**Software Design Document:**

**Table Of Contents:**

* **1) Header…………………………………………………………1**
* **2) System Overview………………………………………….2**
* **3)System Context………………………………………………2**
* **4)Key Features…………………………………………………..3**
* **5)Assumption and dependincies………………………..3**
* **6) Architectural Design………………………………………3**
* **7) Component Desing…………………………………………4**
* **8)Data Design ……………………………………………………5**
* **9)Design Patterns……………………………………………….5**
* **10) Implementation Notes…………………………………5**
* **11) User Interface Design……………………………………6**
* **12)External Interfaces………………………………………6**
* **13) Task Matrix………………………………………………..6**
* **System Overview:**
  + The *Cleaners* system is designed to enhance the longevity and balance of the game world by controlling pollution levels. It introduces dynamic "Cleaner" agents that automatically respond to increased pollution levels and initiate actions to reduce it.
* **System Context:**
  + Cleaner (main actor)
  + PollutionManager (central pollution tracking system)
  + Environment/World (source of pollution increase)
  + The PollutionManager observes pollution sources.
  + Cleaners observe the PollutionManager.
  + When pollution crosses a threshold, Cleaners are notified to act.
  + metin, diyagram, paralel, çizgi içeren bir resim

    Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.
* **Key Features and Functionality:**
  + Automatic pollution tracking.
  + Automatic notification to Cleaners.
  + Dynamic pollution reduction behavior.
  + Enhances sustainability within the game world.
* **Assumptions and Dependencies:**
  + Pollution values increase due to external in-game events.
  + The game engine supports observer behavior
  + Cleaners are registered with PollutionManager.
  + PollutionManager must be updated regularly
* **Architectural Design:**

**diyagram, metin, ekran görüntüsü, çizgi içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.**

* **Architectural Pattern: Observer Pattern**
  + Using the Observer Pattern allows us to decouple the PollutionManager (subject) from the Cleaner components (observers). This allows flexibility in adding/removing cleaners dynamically without modifying the manager.

**Component Design:**

**metin, diyagram, ekran görüntüsü, çizgi içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.**

* + **a) Subsystems and Modules**

**PollutionManager: Tracks pollution level, notifies observers.**

**Cleaner: Reacts when notified to clean up pollution.**

**World: Triggers events that increase pollution.**

**b) Responsibilities**

**PollutionManager: Maintain pollution state and notify observers.**

**Cleaner: Reduce pollution upon notification.**

**World: Cause pollution growth.**

**Data Design:**

* diyagram, metin, çizgi, ekran görüntüsü içeren bir resim

  Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.

Data Validation Rules

* + Pollution must be a non-negative integer.
  + Cleaner must have a valid cleaningPower > 0.
  + Notification occurs only if pollution exceeds a threshold.
* **Design Patterns:**

**i)**Name: Observer Pattern

ii) Why:

Because Cleaners must react automatically to pollution changes tracked by a central system. The Observer Pattern is ideal for reactive systems like this.

iii) Describe how:

A central object called PollutionManager keeps track of pollution. Cleaners subscribe to changes in this value. When pollution increases, all subscribed Cleaners are notified and act to reduce it

* **Implementation Notes:**
  + Use C# events or Unity's messaging system to implement observer notifications.
  + Register cleaners with the manager in Start() or dynamically.
  + Implement pollution logic in a centralized script.
* **User Interface Design:**
  + Environmental feedback: visual decay
* **External Interfaces:**
  + Unity Game Engine APIs.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | System Context | Architectural Designs and Key Features | Component Design Clarification | Data Design Clarification | Notes and Interfaces |
| Umut Baran Boztaş | X |  |  |  |  |
| Mehmet Efe Palaz |  |  |  | X |  |
| Kaan Behzetoğlu |  |  |  |  | X |
| Mehmet Fatih Akay |  | X |  |  |  |
| Efe Selim Sürekli |  |  | X |  |  |