MACAU UNIVERSITY OF SCIENCE AND TECHNOLOGY

School of Computer Science and Engineering

Faculty of Innovation Engineering

<<Software Project for Course Software Engineering>>

Homework ID : Task3-Structured Requirements Analysis

Report Title : Tarturus Structured Requirements Analysis

Student Name : Yu Siyuan, Liu Benhuang

Student No. : 1220004131, 1220004875

Date : 12/11/2024

Abstract

The "Tarturus Structured Requirements Analysis" details the comprehensive design and functionality of a gaming system through Entity-Relationship (ER), Data Flow (DFD), and State Transition Diagrams (STD). The ER diagram highlights the interconnected roles of game entities like Players, Enemies, and Maps. The DFD focuses on data processing and system interactions, ensuring dynamic gameplay responsiveness. The STD illustrates state changes in response to player actions, emphasizing real-time engagement. Together, these diagrams demonstrate a sophisticated framework designed to enhance player experience and system efficiency within the game.

Table of Contents

List of Figures

Chapter 1 Entity-Relationship Diagram

1.1 Overview

The ER diagram models the data structure for the game, showing the relationships between entities such as Player, Enemy, Map, Achievement, Music, and Game Setting. Each entity has attributes that define its role in gameplay, contributing to the interactive and immersive nature of the game.

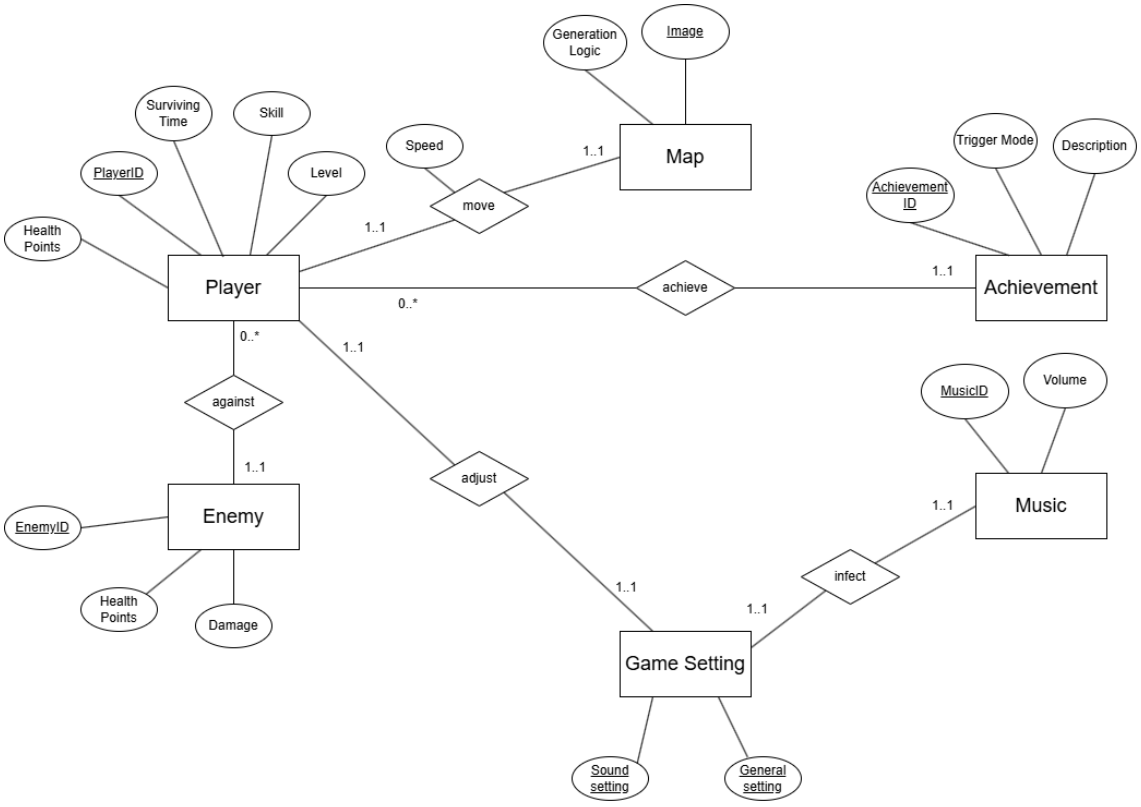

Fig 1-1 The ER diagram

1.2 Key Entities and Their Attributes

Player: Attributes include PlayerID, Health Points, Surviving Time, Skill, and Level.

Enemy: Attributes include EnemyID, Health Points, and Damage.

Map: Attributes include Generation Logic, Speed, and Image.

Achievement: Attributes include AchievementID, Trigger Mode, and Description.

Music: Attributes include MusicID and Volume.

Game Setting: Attributes include Sound setting and General setting.

1.3 Relationships Between Entities

Player and Enemy (against): Defines interactions between player and enemy during combat.

Player and Map (move): Player navigates across different maps.

Player and Achievement (achieve): Achievements unlocked based on actions.

Game Setting and Music (infect): Game settings can adjust music settings.

1.4 Diagram Representation and Analysis

The ER diagram visualizes the data structure and relationships. This clear layout aids in understanding how each entity supports gameplay and contributes to player engagement.

Chapter 2 Data Flow Diagram

2.1 Overview of the Data Flow Diagram (DFD)

This DFD represents the flow of data and processes in our team game system. The system involves several key processes, including receiving user input, updating game states, and integrating the information to the game server. The diagram showcases the high-level interactions and data exchanges between the various components of the system.

2.2 Level-1 DFD Presentation

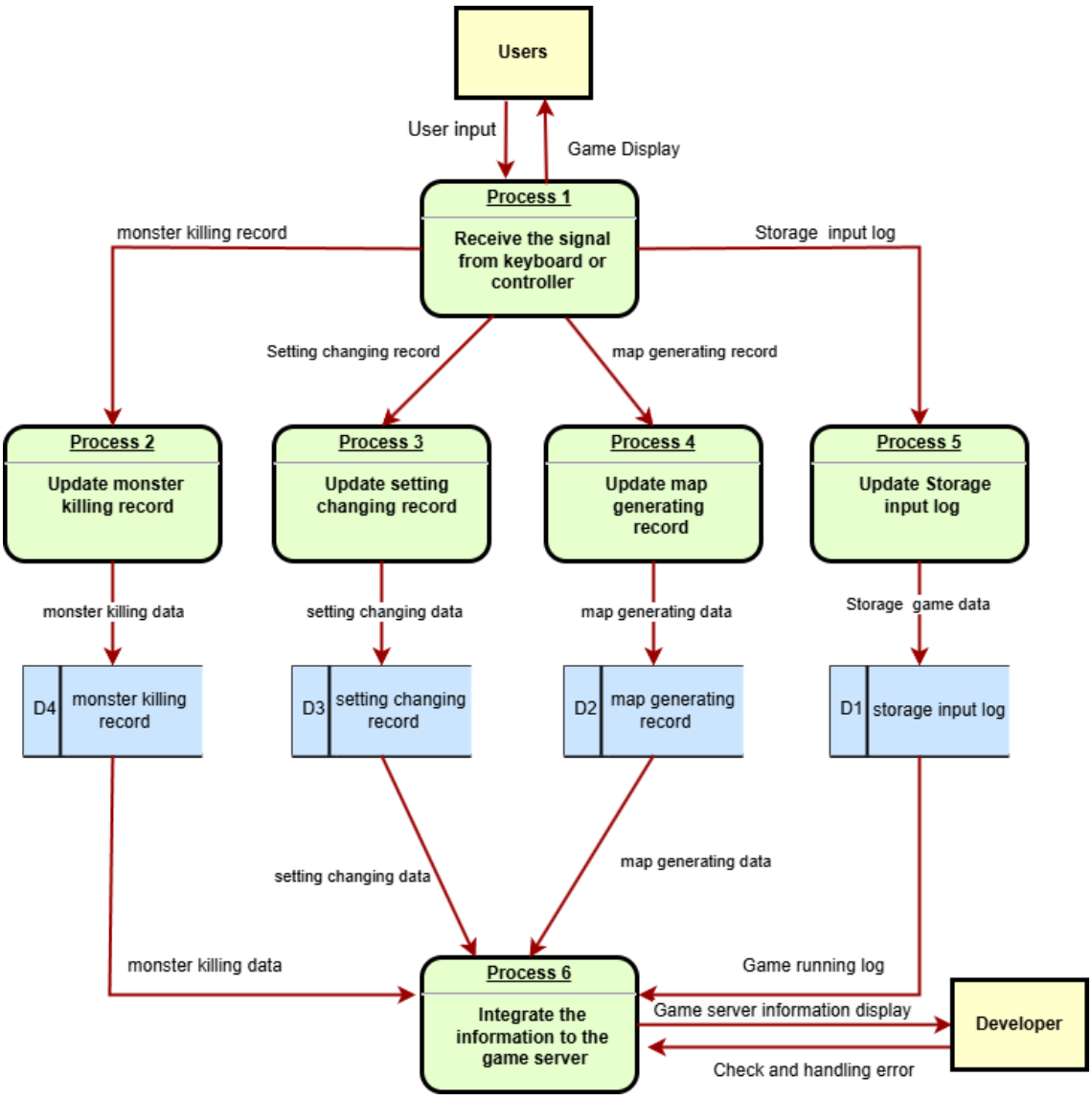

Fig 2-2-1 Level-1 Data Flow Diagram

The data dictionary for this DFD would include the following entries:

- **User input:** Keyboard strokes or controller actions from the user.
- **Game display:** Output shown to the user during the game.
- **Monster killing data:** Information on monsters killed by the player.
- **Map generating data:** Data related to the generated game map.
- **Storage game data:** Data stored during the session.
- **Game running log:** Log of the game's runtime information.

2.3 Analysis of the Level-1 Data Flow Diagram (DFD)

- **The Level-1 DFD provides an overarching view of the system.**
  It highlights the central role of Process 1, "Receive the signal from keyboard or controller," which acts as the gateway for all user interactions. The diagram demonstrates the interconnected nature of the processes, as the output of one process often becomes the input for another.
- **The Level-1 DFD shows the relationship between the processes and data.**
  where the data generated by the processes is stored for later use or retrieval. This includes datastores such as the "monster killing record," "map generating record," "setting changing record," and "storage input log."

2.4 Level-2 DFD Presentation

The Level-2 DFD offers a more detailed view of the system's processes and their internal workings. It helps to further understand the specific responsibilities and functionalities of each process.

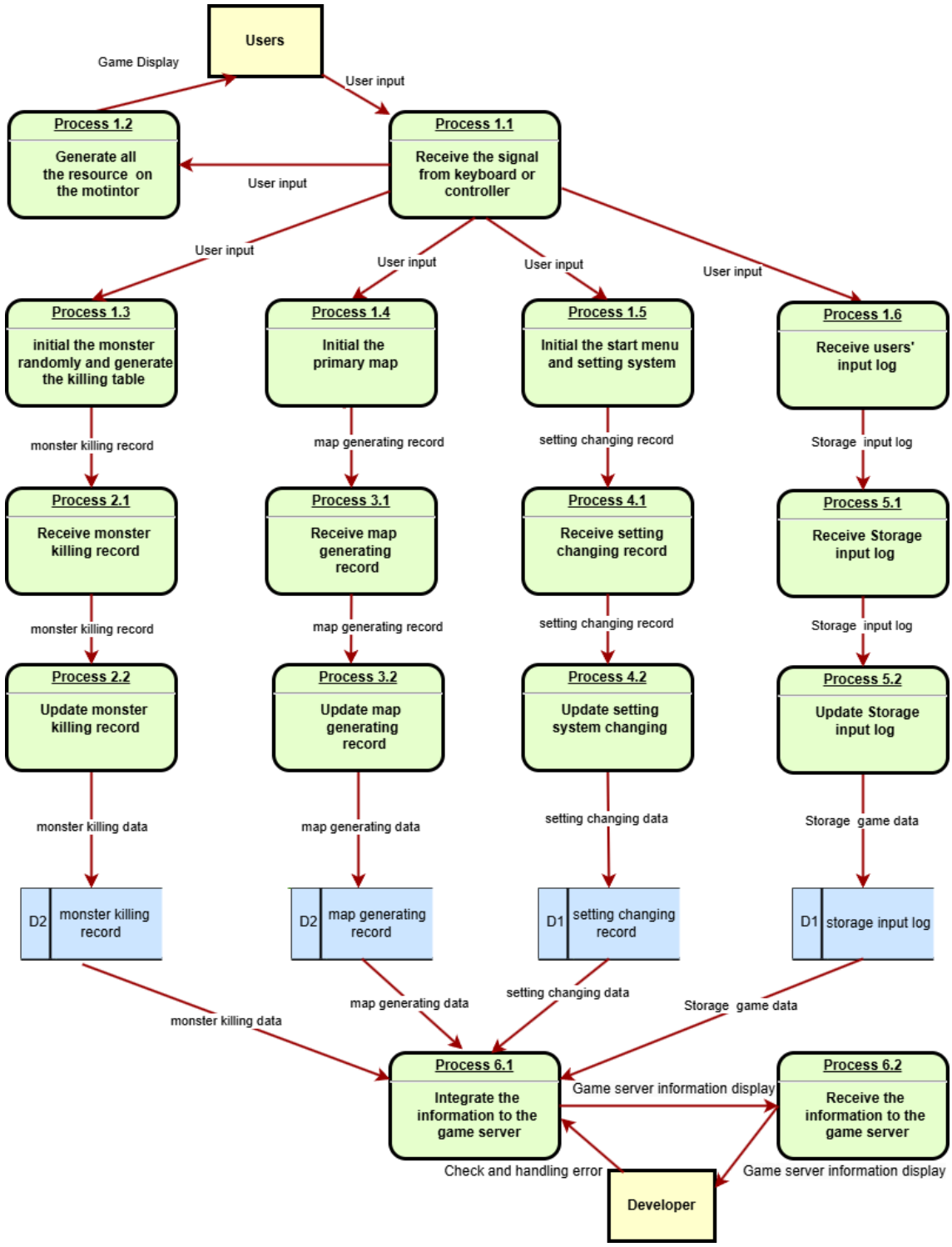

Fig 2-4-1 level-2 Data Flow Diagram

2.5 Analysis of the Level-2 Data Flow Diagram (DFD)

The Level-2 DFD offers a detailed examination of the processes and data flows within the game system, providing insights into the specific functionalities of each component.

1. User Input Handling:

- **Process 1.1 - Receive the signal from keyboard or controller:** This initial process captures user inputs, acting as the gateway for all interactions. It ensures that all actions are appropriately directed to subsequent processes.

2. Resource Generation:

- **Process 1.2 - Generate all the resource on the monitor:** This process is crucial for creating a dynamic game display, ensuring that users can see the necessary visual elements based on their input.
- **Process 1.3 - Initial the monster randomly and generate the killing table:** This initializes the game environment by randomly generating monsters and setting up a record for monster kills, enhancing gameplay engagement.
- **Process 1.4 - Initial the primary map:** This establishes the foundational map for the game, which is essential for navigation and gameplay strategy.

3. User Interface and Settings Management:

- **Process 1.5 - Initial the start menu and setting system:** This process manages the start menu and game settings, allowing users to customize their experience before gameplay begins.
- **Process 1.6 - Receive users' input log:** This logs user interactions throughout the session, providing valuable data for monitoring user behavior and improving the game experience.

4. Data Recording and Updating:

- **Process 2.1 - Receive monster killing record:** This process updates the datastore with information about monsters killed, contributing to the game's progression and player achievements.

- **Process 3.1 - Receive map generating record:** This ensures that the map generation data is accurately recorded, which is vital for maintaining the game state.
- **Process 4.1 - Receive setting changing record:** This captures any alterations made to game settings, allowing for a tailored gaming experience based on user preferences.
- **Process 5.1 - Receive Storage input log:** This updates the log of user inputs, further enriching the data available for analysis.

5. Integration and Server Communication:

- **Process 6.1 - Integrate the information to the game server:** This critical process aggregates data from various sources, preparing it for transmission to the game server. It ensures that all gameplay information is centralized.
- **Process 6.2 - Receive the information to the game server:** This final step involves sending the integrated data to the server, ensuring real-time updates and maintaining the integrity of the game environment.

Chapter 3 State Transition Diagram

3.1 Overview

The State Transition Diagram (STD) models the game's behavior by illustrating how the system transitions between various states in response to player actions.

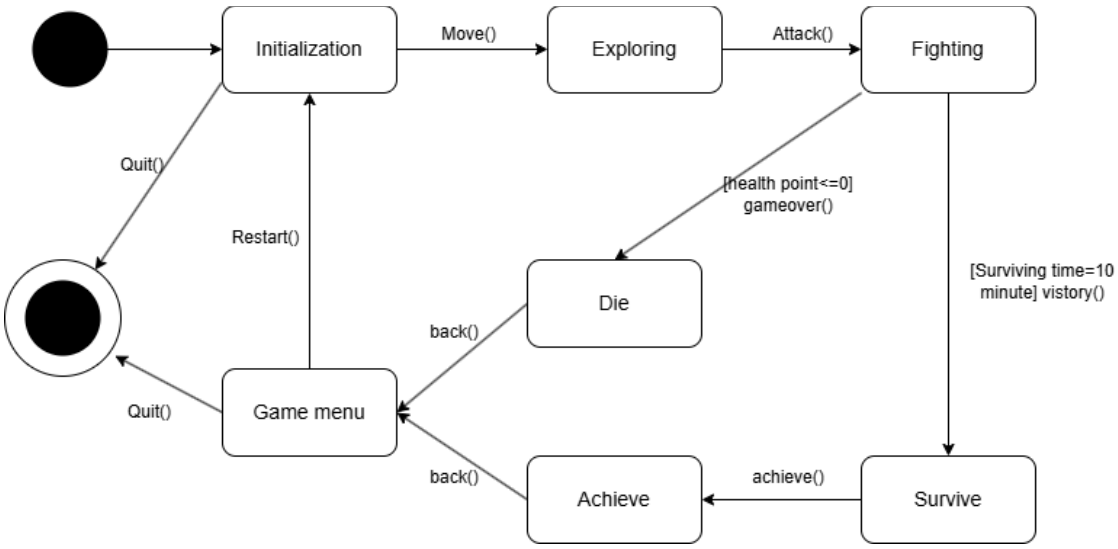

Fig 3-1 The STD diagram

3.2 Key States and Transitions

Initialization: Game setup with player attributes.

Exploring: Player moves across maps, potentially encountering enemies.

Fighting: Combat state triggered by encounters with enemies.

Die: Triggered if health points reach zero.

Achieve: Represents an achievement unlocked by the player.

Survive: Indicates survival for a specific duration.

3.3 Behavior Mapping in Gameplay

The STD shows gameplay flow, from exploring maps to encountering enemies, achieving milestones, or facing defeat. Each transition reflects player actions and game responses, creating an engaging experience.

3.4 Diagram Representation and Analysis

The STD visually represents the states and transitions, supporting the system’s functionality and player interaction model.

References

1. https://github.com/steinsgo/Tarturus/blob/main/Project/ERdiagram.drawio
2. https://github.com/steinsgo/Tarturus/blob/main/Project/level1.drawio
3. https://github.com/steinsgo/Tarturus/blob/main/Project/level2.drawio
4. https://github.com/steinsgo/Tarturus/blob/main/Project/SFD.drawio
5. https://github.com/steinsgo/Tarturus/blob/main/Project/SE%20context%20diagram.drawio