

CIPHER GAME - TECHNICAL DESIGN DOCUMENT (TDD)

GitHub Repository: <https://github.com/ShahzainAli23/CyberSecurityGame>

Team Members:

- Fatima Mahmood (24314)
 - Arbaz Asif (25123)
 - Babar Ali (24154)
 - Sarfaraz Ahmed (24520)
-

1. Overview:

Cipher Game is a 2D top-down educational puzzle game developed in Godot Engine. It introduces students to classical cryptographic techniques such as Caesar, Vigenère, and substitution ciphers. The player explores a pixel-art environment, interacts with objects, solves cipher-based challenges, and progresses through levels by unlocking encrypted doors.

2. Game Engine Choice:

Godot Engine 4.3 was selected over Unity due to:

- Native support for 2D game development
 - Lightweight performance (smoother experience on low-end machines)
 - Simpler scene/node-based architecture
 - Built-in scripting (GDScript) and signal system
-

3. High-Level Architecture:

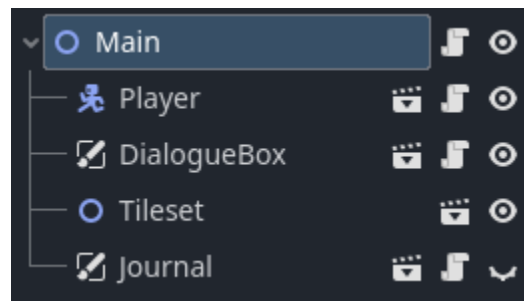
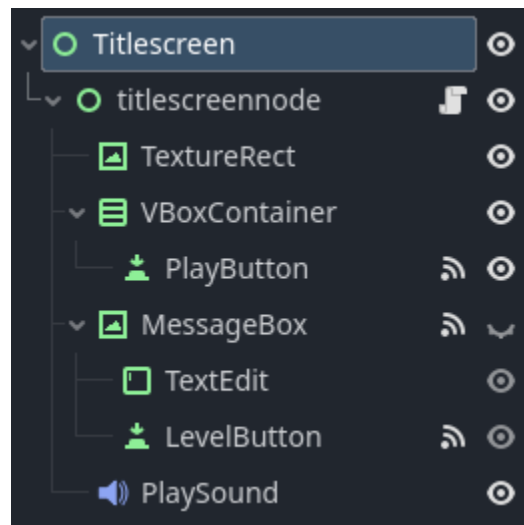
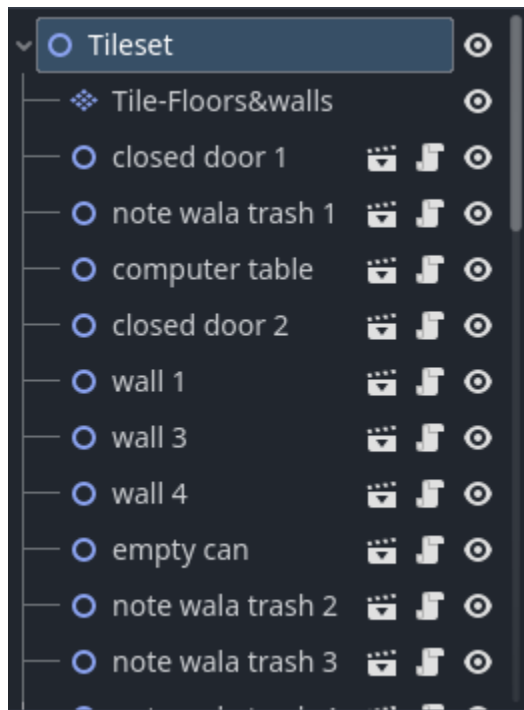
Titlescreen.tscn

↳ Play Button → Loads Main.tscn

Main.tscn

- ↳ Loads Tileset.tscn
 - ↳ Tileset contains all interactable items
- ↳ Player.gd (CharacterBody2D)
- ↳ DialogueBox (CanvasLayer)
- ↳ Journal (CanvasLayer)
- ↳ Background Music (via SoundManager singleton)

Scenes are built using reusable templates (e.g., `item.tscn`) with flexible scripts. Each interactive object (e.g., `red_tv`, `open_door`, `notes`) has its own GDScript file with encapsulated logic.



4. Key Features & Implementation:

a. Top-Down Movement:

- Node: `CharacterBody2D`
- Input via `Input.get_vector()`
- Directional animation switching (Idle, Run, Talk)

b. Interaction System:

- Items use `Area2D` to detect proximity
- Player press `C` to trigger `interact()` function
- Each object script defines its custom interaction behavior (dialogue, password check, destroy, etc.)

c. Dialogue System:

- Implemented with `CanvasLayer`, `Label`, and `NextIcon`
- Dialogue lines are typed out character-by-character (typing effect)
- Password interaction integrated into the same system

d. Journal System:

- `CanvasLayer` with buttons (Next, Prev, Close)
- Pages are stored in an array, cycling logic via button signals

e. Password Puzzle Mechanic:

- Object stores expected password as a string
- If entered correctly, it replaces the current node with a new one (e.g., `closed_door` → `open_door`)

- If wrong, shows feedback dialogue

f. Sound System:

- Step sounds, dialogue beeps, password success/fail, door opening, and background music
- Centralized in `SoundManager.gd` (autoload singleton)

g. Level Progression Logic:

- 4 cipher puzzles (each opens a door)
- Final (5th) puzzle unlocks red TV
- Correct final password spawns blue TV
- Blue TV ends the game on interaction

5. Technical Stack:

- **Engine:** Godot 4.3
- **Language:** GDScript
- **Art Tools:** Pixelorama (custom pixel assets), itch.io (tilesets)
- **Audio:** Pixabay
- **Version Control:** Git (GitHub repo provided)

6. Game Logic & AI:

- Minimal AI used; puzzles are static
- All logic handled through signals and scripted state changes

- No procedural generation or pathfinding
-

7. Audio/Visual Integration:

- Sound effects triggered via helper functions
 - Visual polish includes animations, pixel art environments, and smooth transitions
-

8. Networking:

- Not applicable. Single-player offline game.
-

9. Platform Requirements:

- **OS:** Windows/Linux/macOS
 - **RAM:** 4GB+
 - **Disk:** < 500MB
 - **GPU:** Integrated graphics supported
-

10. Future Improvements:

- Add admin interface for managing puzzles
- Add level selection menu
- Dynamic difficulty scaling or hint toggle
- Procedural puzzle generation for replayability

11. Conclusion:

This TDD outlines the technical core of Cipher Game, including its architecture, logic, engine usage, and game systems. The game was designed with modularity and clarity, enabling each object to encapsulate its own logic and making it easy to expand in the future.