Witcher Dungeon Game - Use Case Specification

1. Actors Definition

Primary Actor:

Player (Gerald): The main character controlled by the user who navigates the dungeon, collects rewards, and fights enemies.

Secondary Actors:

- **Monster**: Moving enemy that pursues the player; vulnerable to silver sword
- **Human Enemy**: Moving enemy that pursues the player; vulnerable to steel sword
- **Regular Reward**: Static collectible object required to win; adds points to score
- **Bonus Reward**: Temporary collectible that appears randomly; adds bonus points but not required to win
- **Punishment Entity**: Static penalty object placed in cells; reduces score
- **Game System**: Manages game state, time, scoring, and win/lose conditions

2. Use Case Descriptions

Use Case 1: Start New Game

Primary Actor: Player
Goal in Context: To initialize a new dungeon game session with the player character at the starting position, ready to navigate through the dungeon.
Preconditions: Game application is launched; no game session is currently active.
Trigger: Player selects "New Game" from the main menu.
Scenario:

1. Player selects "New Game" option
2. System loads dungeon map and creates GameUI
3. System initializes Board with grid of cells
4. System places Player (Gerald) at starting cell
5. System spawns monsters using MonsterFactory and humans using HumanFactory
6. System places regular rewards throughout the dungeon
7. System equips player with steel sword (default weapon)
8. System registers ScoreDisplay and TimerDisplay as observers
9. System displays game board with initial score (0) and timer (0:00)
10. System begins game tick cycle

Exceptions: Map file cannot be loaded → display error and return to menu; Invalid map configuration → load default map; Memory allocation fails → exit gracefully.
Priority: Essential | **Frequency:** Once per game session
Open Issues: Should the player select difficulty level? Tutorial mode for first-time players? Starting equipment choice?

Use Case 2: Move Player Character

Primary Actor: Player
Goal in Context: To navigate the player character through the dungeon by moving one cell at a time in cardinal directions.
Preconditions: Game is active; player character is alive; game is not paused.
Trigger: Player presses arrow key or WASD key.

Scenario:

1. Player presses directional key
2. System creates MoveCommand with specified direction
3. System validates target cell is not blocked by wall or barrier
4. Player moves to target cell if valid
5. System checks cell contents (reward, punishment, or enemy)
6. System updates player position and triggers game tick
7. System notifies observers and redraws board

Exceptions: Target blocked by wall → no movement, feedback provided; Target contains enemy → combat initiated (UC5); Target contains regular reward → collected, score increases, reward removed; Target contains punishment → penalty applied, if score < 0 player loses; Exit reached without all rewards → move invalid; Exit reached with all rewards → player wins (UC9).
Priority: Essential | **Frequency:** Multiple times per second during gameplay
Open Issues: Diagonal movement allowed? Movement animation or instant? Ticks per second?

Use Case 3: Switch Weapon

Primary Actor: Player
Goal in Context: To switch between silver sword (effective against monsters) and steel sword (effective against humans).
Preconditions: Game is active; player is alive; player has both swords.
Trigger: Player presses weapon switch key (e.g., 'Q' or '1'/'2').
Scenario:

1. Player presses weapon switch key
2. System identifies current equipped sword
3. System uses WeaponFactory to access alternate sword
4. Player equips alternate sword (Steel ↔ Silver)
5. System updates weapon display indicator
6. System continues normal gameplay

Exceptions: Only one sword available → no switch, message displayed; Switch during attack → queued until animation completes.
Priority: High priority | **Frequency:** Several times per session
Open Issues: Should switching cost a game tick? Cooldown period? Auto-select weapon near enemies?

Use Case 4: Enemy Pursues Player

Primary Actor: Monster or Human Enemy
Goal in Context: For enemy to move toward player's position, potentially catching and defeating the player.
Preconditions: Game is active; enemy is alive; player is alive; game tick occurred.
Trigger: Game engine tick cycle executes.
Scenario:

1. System triggers game tick event
2. Enemy queries its position and player's position
3. Enemy uses MovementStrategy (ChaseStrategy) to calculate next move
4. System validates target cell is not blocked
5. Enemy moves one cell closer to player
6. System checks if enemy caught player (same position)
7. System updates board display

Exceptions: Move blocked by wall → use RandomStrategy for alternative; Move blocked by another enemy → wait this tick; Enemy catches player → player loses immediately (UC10); Enemy passes reward/punishment → item remains (enemies don't interact with items).
Priority: Essential | **Frequency:** Every game tick for each enemy
Open Issues: Different movement speeds? Pathfinding around obstacles? Maximum detection range?

Use Case 5: Combat with Enemy

Primary Actor: Player
Goal in Context: To attack and defeat an enemy using the appropriate sword type.
Preconditions: Game is active; player is alive; player adjacent to enemy; player has sword equipped.
Trigger: Player presses attack key (e.g., spacebar) when near enemy.
Scenario:

1. Player presses attack key while adjacent to enemy
2. System identifies enemy type (Monster or Human) and equips required sword
3. System creates and executes attack command
4. System applies damage based on sword-enemy compatibility
5. Enemy health reduced; if health ≤ 0, enemy removed from board
6. System plays combat animation/sound and triggers game tick
7. System updates display

Exceptions: Wrong sword used (Steel vs Monster or Silver vs Human) → reduced damage, enemy survives; Correct sword used → enemy defeated in one hit; No enemy adjacent → attack animation only; Enemy moves to player during attack → player loses.
Priority: Essential | **Frequency:** Multiple times per session
Open Issues: Attack cooldown period? Diagonal attacks? Turn-based or real-time combat?

Use Case 6: Collect Regular Reward

Primary Actor: Player
Goal in Context: To collect a regular reward, increasing score and progressing toward win condition.
Preconditions: Game is active; regular reward exists on cell; player is alive.
Trigger: Player moves to a cell containing regular reward.
Scenario:

1. Player moves to cell with regular reward
2. System detects reward and grants the player the reward
3. System removes reward from cell and board collection
4. System notifies ScoreDisplay observer
5. ScoreDisplay updates score on screen
6. System plays collection sound/animation
7. System checks if all regular rewards collected

Exceptions: Bonus reward encountered → see UC7; Multiple rewards in cell → collect all; Reward collection with enemy collision → enemy collision takes precedence, player loses.
Priority: Essential | **Frequency:** 10-20 times per session
Open Issues: Different point values? Visual distinction from bonus rewards? Trigger enemy alerts?

Use Case 7: Collect Bonus Reward

Primary Actor: Player
Goal in Context: To collect a temporary bonus reward before it disappears, gaining extra points (not required for winning).
Preconditions: Game is active; bonus reward spawned; timer not expired; player is alive.
Trigger: Player moves to cell containing active bonus reward.
Scenario:

1. System randomly spawns bonus reward at empty cell
2. BonusReward timer starts countdown (e.g., 5 ticks)
3. System displays bonus with visual indicator (flashing/glowing)
4. Player moves to cell before timer expires
5. Player score increased by bonus value
6. System removes bonus, notifies ScoreDisplay, plays special sound

Exceptions: Timer expires → reward disappears, no points; Bonus spawns on player cell → auto-collected; Enemy on bonus cell → reward remains, collectible after enemy defeated.
Priority: Medium | **Frequency:** 2-5 times per session
Open Issues: Duration of bonus visibility? Location hints? Spawn rate increases over time?

Use Case 8: Encounter Punishment

Primary Actor: Player
Goal in Context: Player encounters punishment entity, resulting in score penalty and potential game loss.
Preconditions: Game is active; punishment exists on cell; player is alive.
Trigger: Player moves to cell containing punishment.
Scenario:

1. Player moves to punishment cell
2. System detects punishment and applies penalty to player
3. Player score decreased by a penalty value
4. System removes punishment and checks if score < 0
5. System notifies ScoreDisplay observer
6. ScoreDisplay updates score (shows red if negative)
7. System plays penalty sound/animation
8. System continues gameplay if score ≥ 0

Exceptions: Score drops below 0 → player loses immediately (UC10); Score exactly 0 → game continues but player must collect reward before next punishment; Punishment at exit → exit becomes inaccessible.
Priority: Essential | **Frequency:** 3-8 times per session
Open Issues: Varying penalty values? Grace period before first punishment? Visible or hidden? Avoidable/destroyable?

Use Case 9: Win Game

Primary Actor: Player
Goal in Context: To successfully complete the dungeon by collecting all regular rewards and reaching the exit.
Preconditions: Game is active; player is alive; all regular rewards collected; score ≥ 0.
Trigger: Player moves to exit cell after collecting all regular rewards.
Scenario:

1. Player collects final regular reward
2. System confirms all regular rewards collected and activates exit cell
3. Player moves to exit cell

4. System validates exit conditions met and stops game tick cycle
5. System calculates final score and total game time
6. System displays "You Won!" message with final score and time
7. System plays victory sound/animation
8. System offers options: "Play Again" or "Return to Menu"

Exceptions: Exit reached without all rewards → move blocked, message displayed; Exit reached with enemy adjacent → victory still occurs.
Priority: Essential | **Frequency:** Once per successful session
Open Issues: Score leaderboard? Completion time affect score? Multiple difficulty levels? Bonus rewards affect grade (S, A, B, C)?

Use Case 10: Lose Game

Primary Actor: Player
Goal in Context: Game ends unsuccessfully when player is caught by enemy or score becomes negative.
Preconditions: Game is active; player is alive (until loss condition).
Trigger: Enemy moves to player's cell OR player's score drops below 0.
Scenario (Path A - Enemy Collision):

1. Enemy moves to player's cell (or vice versa)
2. System detects collision and stops game tick
3. System displays "Game Over - Caught by Enemy!" with final score/time
4. System offers "Try Again" or "Return to Menu"

Scenario (Path B - Negative Score):

1. Player moves to punishment cell
2. Punishment applies penalty, score becomes negative
3. System stops game tick
4. System displays "Game Over - Score Below Zero!" with final score/time
5. System offers "Try Again" or "Return to Menu"

Exceptions: Simultaneous collision and negative score → enemy collision message; Multiple enemies reach player → single "Game Over" message.
Priority: Essential | **Frequency:** 1-10 times per session
Open Issues: Continue option (extra life)? Checkpoint saves? Difficulty settings affect loss conditions?

Use Case 11: Pause Game

Primary Actor: Player
Goal in Context: To temporarily halt gameplay without losing progress.
Preconditions: Game is active; game is not already paused.
Trigger: Player presses pause key (e.g., 'ESC' or 'P').
Scenario:

1. Player presses pause key
2. System stops game tick cycle and freezes timer
3. System displays pause menu: "Resume", "Restart", "Settings", "Quit"
4. Player selects "Resume"
5. System hides pause menu and resumes game tick and timer

Exceptions: Settings selected → display settings menu; Restart selected → confirm, then UC1; Quit selected → confirm, return to main menu.
Priority: Medium | **Frequency:** 1-5 times per session

Open Issues: Timer continues during pause? Allow pause during animations? Auto-pause when the window loses focus?

3. Design Pattern Integration

Creational Patterns:

- **Factory Method** (UC1, UC4): MonsterFactory and HumanFactory create appropriate enemy types
- **Abstract Factory** (UC3): WeaponFactory creates SilverSword and SteelSword
- **Singleton** (UC1): GameEngine ensures single game instance

Structural Patterns:

- **Composite** (UC4): EnemyGroup manages collections of enemies moving together
- **Adapter** (UC6, UC7): RewardAdapter integrates legacy reward system
- **Bridge** (UC5): Weapon abstraction separates interface from implementation

Behavioral Patterns:

- **Observer** (UC2, UC6-10): ScoreDisplay and TimerDisplay observe game state changes
- **Command** (UC2, UC5): MoveCommand and AttackCommand encapsulate player actions
- **Strategy** (UC4): MovementStrategy (ChaseStrategy, RandomStrategy) enables different AI behaviors
- **Iterator** (UC1): BoardIterator traverses board cells during initialization

4. Use Case Relationships

Triggers Win Condition (UC9): UC2 (Move), UC6 (Collect Regular Reward)
Triggers Loss Condition (UC10): UC2 (Move to enemy), UC4 (Enemy catches player), UC8 (Negative score)
Supporting Use Cases: UC3 (Switch Weapon) supports UC5 (Combat); UC11 (Pause) can interrupt any active use case
Concurrent Use Cases: UC4 (Enemy Pursues) runs parallel to UC2 (Player Movement) during each game tick