

Detailed Use Cases (Iteration 1) for System Finding Trump

Assignment in the course PA1435 Object Oriented Design

2017-04-10

Author Name	Social Security#	Thinking	Writing
Emil Wahl	880729-0351	17%	17%
Fredrik Carlsson	961206-0070	17%	17%
Daniel Cheh	941231-1772	17%	17%
Aksel Grahn	940311-0159	17%	17%
Sebastian Vusak	960406-1110	17%	17%
Gustav Björk	960223-4131	17%	17%

System Description

The system is based on the classic game Nethack. Nethack is a text-based and two-dimensional game. It is a rogue dungeon crawler with a random generated map. In this dungeon you will be able to find treasure and fight monster.

The main difference between our system and Nethack is that our system is connected to Twitter and all the levels, items, and monsters are generated from a seed in the form of a Twitter feed.

Detailed Use Cases

Use Case: Generate level

Actors: Twitter

Description: System reads Twitter Feed and generates level, items and monsters based on keywords in Twitter feed.

Preconditions: No game is running.

Main Course of Events:

Twitter	System
1. Twitter finds seed.	2. System generates level from seed. 3. System displays level.

Alternative Flow of Events:

1. Empty seed is found. Try again until a good seed is found.

Special Requirements: None

Use Case: Move around

Actors: Player, NPC

Description: Players and Monsters tells system their move order. System registers move order and moves the players/monsters to new location.

Preconditions: None.

Main Course of Events:

Player, NPC	System
1. Player / NPC wants to move to a certain position.	2. System register that movement. 3. System moves Player / NPC

Alternative Flow of Events:

3. Can't move around because e.g. a rock is blocking the road. Player remains in same position.

Special Requirements: None.

Use Case: Pick up item

Actors: Player

Description: Player picks up object. System registers and updates level and player's inventory.

Preconditions: Object within reach of the player. Player is not in combat.

Main Course of Events:

Player	System
1. Player picks up item.	2. System adds item to player inventory. 3. System removes item from level.

Alternative Flow of Events:

1. Enemy is next to object. Object remains on ground. Combat starts.

1. Object is trap. Player takes damage.

Special Requirements: None.

Use Case: Winning conditions

Actors: Player

Description: Players find a specific object that ends the game. System display message for victory.

Preconditions: Player is within reach of the winning item. Player is not in combat.

Main Course of Events:

Player	System
1. Player pick up winning item.	2. System register pick up item. 3. System display winning message.

Alternative Flow of Events: None.

Special Requirements: None.

Use Case: Show character information

Actors: Player

Description: Player opens inventory. Systems displays player's character information and objects in inventory.

Preconditions: Player is not in combat. Player is standing still.

Main Course of Events:

Player	System
1. Player presses inventory key.	2. System displays inventory and character information.

Alternative Flow of Events: None.

Special Requirements: None.

Use Case: Fight Monster

Actors: Player, NPC

Description: Players and Monster engage in a fight. System records actions. System determines outcome. Displays result.

Preconditions: Player stands next to monster.

Main Course of Events:

Player, NPC	System
1. Player/NPC decides which action to perform.	2. System registers action. 3. System calculates and updates status of actors. 4. System displays result.

Alternative Flow of Events:

3. Action fails. No status update.

Special Requirements: Player needs a weapon equipped.

Use Case: Losing conditions

Actors: Player

Description: Player dies. System displays message for loss.

Preconditions: Player hit points reaches zero

Main Course of Events:

Player	System
1. Player dies when hit points reaches zero	2. System checks if the condition is true 3. System freezes the game and displays the message "Game Over" and a button "Quit"

Alternative Flow of Events: None

Special Requirements: None