

# Use Case Overview for System TwitterNetHack

Assignment in the course PA1435 Objektorienterad Design

2017-04-05

Robin Westerlund, Max Nilsson, Rikard Magnom, Michael Lindroth

Author Name	Social Security Number	Thinking	Writing
Robin Westerlund	9510090195	25%	0%
Max Nilsson	9509052875	25%	100%
Rikard Magnom	9701014699	25%	0%
Michael Lindroth	9012283033	25%	0%

It is written in the Book of Camaxtli:

After the Creation, the cruel god Moloch rebelled against the authority of Marduk the Creator. Moloch stole from Marduk the most powerful of all the artifacts of the gods, the Amulet of Yendor, and he hid it in the dark cavities of Gehennom, the Under World, where he now lurks, and bides his time.

Your god Camaxtli seeks to possess the Amulet, and with it to gain deserved ascendance over the other gods.

You, a newly trained Digger, have been heralded from birth as the instrument of Camaxtli. You are destined to recover the Amulet for your deity, or die in the attempt. Your hour of destiny has come. For the sake of us all: Go bravely with Camaxtli!

## System Description

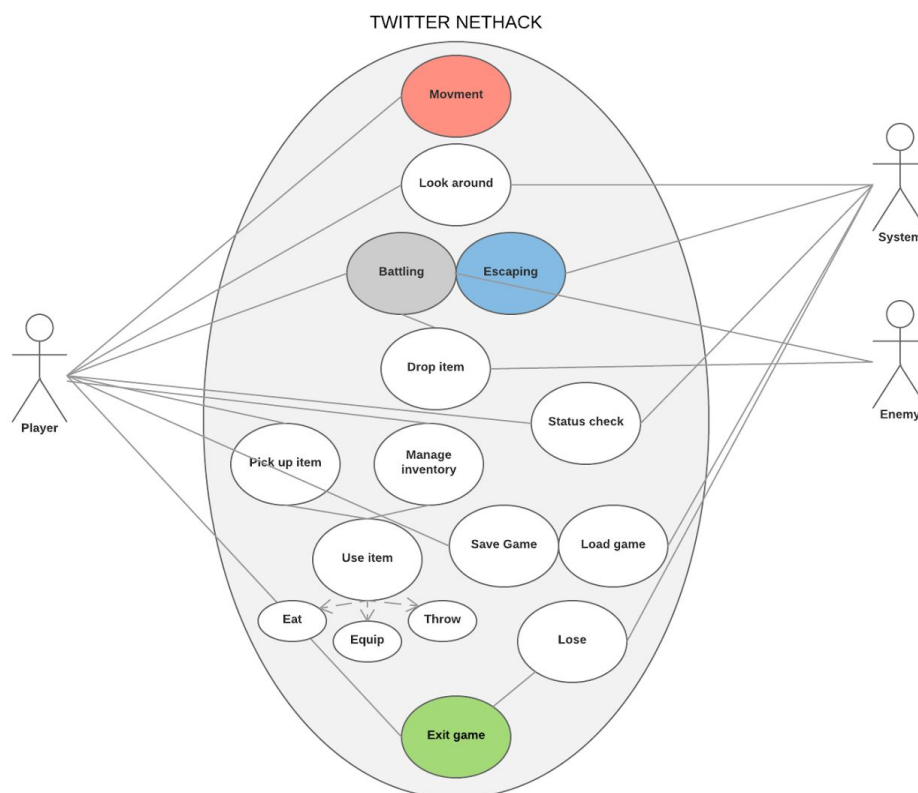
NetHack is a game that involves traversing a maze-like dungeon called “Mazes of Menace”, in which the goal is to obtain the Amulet of Yendor and escape the dungeon alive. Along with the amulet you are encouraged to loot as much treasure as possible, since you want the most points possible. This will place you on the list of high scores.

The goal of the system is to modify the classic game NetHack by using Twitter to generate dungeons and implementing multiplayer support. It will also be expanded in many ways to achieve more depth in the game.

## Use Case Diagram

### TWITTER NETHACK

Max Nilsson | April 10, 2017



## Description of Actors

1. Player: The player is the only human actor in the game, the player will do all the requests and decide on the outcome of the game.
2. System: The system will handle the requests of the player and calculate the results of the player's actions, i.e. random drops from the enemy.
3. Enemy: The enemy that the player will fight when battling or escaping.

## High-Level Use Cases

1. Name: Movement  
Actors: Player  
Type: Primary  
Description: The player indicates that they want to move and in which direction, they move accordingly.
2. Name: Look around  
Actors: Player, System  
Type: Primary  
Description: The player indicates that they want to look and in which direction, then sees what is there.
3. Name: Battling  
Actors: Player, Enemy  
Type: Primary  
Description: The player encounters an enemy and chooses to fight. The player and the enemy trade blows until a victory or defeat is declared.
4. Name: Escaping  
Actors: Player, System  
Type: Primary  
Description: The player encounters an enemy and chooses to escape. Upon a successful escape the battle is avoided entirely.
5. Name: Drop item  
Actors: Enemy.  
Type: Primary  
Description: Enemy can drop items after being defeated.
6. Name: Status check  
Actors: Player, System  
Type: Primary  
Description: The player wants to check their current status, upon choosing to do so they are presented with their current status.
7. Name: Pick up item  
Actors: Player  
Type: Primary  
Description: The player finds an item in the dungeon, and then chooses to pick it up.
8. Name: Manage inventory  
Actors: Player  
Type: Primary  
Description: The player opens their inventory and is presented with their items, then manages the items in some way.
9. Name: Use item  
Actors: Player

Type: Primary

Description: The player enters their inventory, then chooses an item to use and uses the item.

10. Name: Eat

Actors: Player

Type: Secondary

Description: The player can try to eat the item.

11. Name: Equip

Actors: Player

Type: Secondary

Description: The player places the item on themselves

12. Name: Throw

Actors: Player

Type: Secondary

Description: The player throws the item at an enemy.

13. Name: Save game

Actors: Player, System

Type: Primary

Description: The player can save the game to continue later.

14. Name: Load game

Actors: Player, System

Type: Primary

Description: The player can load the game to continue.

15. Name: Lose

Actors: System

Type: Primary

Description: The system tells the player when he/she has lost.

16. Name: Exit

Actors: Player

Type: Primary

Description: The player has finished the game and exits the game.