

Author Name	Social Security Number	Thinking	Writing
Morgan L.	971107-xxxx	25%	25%
Henrik N.	970416-xxxx	25%	25%
Magnus N.	940429-xxxx	25%	25%
Victor O.	970612-xxxx	25%	25%

## System description

The system is going to be based around the classic game NetHack but in 2D graphics. NetHack is a sort of adventure and roleplaying game where you create a character and goes through random generated dungeons. In the dungeons you can meet various different monsters and collect items. The final goal is to find the amulet of Yendor and escape with it alive.

Nethack is a very advanced game and because of this and the limited time we are given, our version of NetHack is not going to include all of the features that the original have. The functionalities we are aiming to implement:

- An opening screen where you can start a new game or join an existing one.
- A screen where you can setup the character.
- Monsters and creatures that freely move around the dungeons.
- Different items to pick up and drop.
- Multiplayer support.
- Generate dungeons with help of twitter.

The dungeons in our NetHack version is going to be generated by the Twitter API. Its job is to read certain keywords in Twitter posts and use these to create specific types of monsters and items for every room in the game. When you enter a new cave, your system shall briefly display information about the “owner” of the cave.

## Detailed Use Cases

Use Case:

Generate Room With Twitter

Actor:

Player, Twitter

Description:

A Twitter feed is read and certain keywords will influence the generation of rooms in the game.

Preconditions:

1. The player has joined/created a game session.
2. The system have access to a twitter feed.

Main Course of Events:

Actor	System
1. The player enters an undiscovered cave	
	2. The system requests a tweet from selected twitter feed
3. Twitter returns a random tweet from the current twitter profile	
	4. The system checks the tweet for keywords.
	5. The system generate the room based on those keywords.

Alternative Flow of Events:

2. The system have no access to twitter and pick random keywords from the keyword storage.

Use Case:

Select a Twitter feed

Actor:

Player

Description:

The Player is inputting someone's Twitter profile URL and the system will monitor the tweets and use the data to generate rooms.

Preconditions:

1. The player have access to a Twitter Profile URL.

Main Course of Events:

<b>Actor</b>	<b>System</b>
1. The player is starting a new game	
	2. The system is asking for a Twitter feed
3. The player types in the Twitter feed URL	
	4. The system stores the URL

Alternative Flow of Events:

3. The player fails to type in the URL

Use Case

Start a singleplayer game / Start Game

Actor

Player

Description

The Player chooses to start a singleplayer game session. No other players are able to join.

Preconditions

1. TwitterNethack is started.

Main Course of Events:

<b>Actor</b>	<b>System</b>
1. Player presses the “singleplayer” button	
2. Player <i>selects a Twitter feed</i>	
	3. System generates a cave with the Twitter-feed and starts the game

Alternative Flow of Events:

2. Player presses multiplayer button

Use Case:

Spawn an Enemy

Actor:

Player, Twitter

Description:

A Twitter feed is read and certain keywords will influence the generation of enemies in the room.

Precondition:

1. The player must be in a cave

Main Course of Events:

<b>Actor</b>	<b>System</b>
1. The player enters an undiscovered room	
	2. The system requests a tweet from selected twitter feed
3. Twitter returns a random tweet from the current twitter profile	
	4. The system checks the tweet for keywords.
	5. The system generate the enemies based on the keywords.

Alternative Flow of Events:

2. The system have no access to twitter and pick random keywords from the keyword storage.

Use Case:

Enemy Attack Player

Actor:

Player

Description:

Enemy attack the player and deals damage. The damage is based of the enemy's attributes levels.

Preconditions:

1. The player is in a game session
2. An enemy exists

Main Course of Events:

<b>Actor</b>	<b>System</b>
1. The player walks in detection range of the Enemy	
	2. The Enemy moves towards the player
	3. The player is in attack range of the enemy
	4. The Enemy attacks the player
	5. The system calculates the damage based on the enemy and the players attributes
6. The player takes damage	

Alternative Flow of Events:

6. the player blocks the damage

Use Case:

Attacks Enemy With Melee Weapon

Actor:

Player

Description:

The player hits the enemy with a melee weapon. The damage is based of the players attributes levels.

Preconditions:

1. The player is in a game session
2. An enemy exists
3. Player has have a melee weapon equipped

Main Course of Events:

<b>Actor</b>	<b>System</b>
1. The player moves towards an enemy	
2. The player is in attack range and attacks the enemy	
	3. The system calculates the damage based on the player's and enemy attributes
	4. The enemy takes damage

Alternative Flow of Events:

4. The enemy block the damage