

2017-04-10

# Use Case Overview for System

**Assignment in the course PA1435, Objektoriented design**

Igor Radulovic, Fredrik Engvall, David Fröström, Sara Huslid, Jonathan Andersson

<b>Name</b>	<b>Social Security</b>	<b>Contribution Thinking (in %)</b>	<b>Contribution Writing(in %)</b>
Igor Radulovic	940524-8858	20	20
Fredrik Engvall	930329-2453	20	20
David Fröström	930222-5512	20	20
Sara Huslid	950804-1002	20	20
Jonathan Andersson	900522-2556	20	20

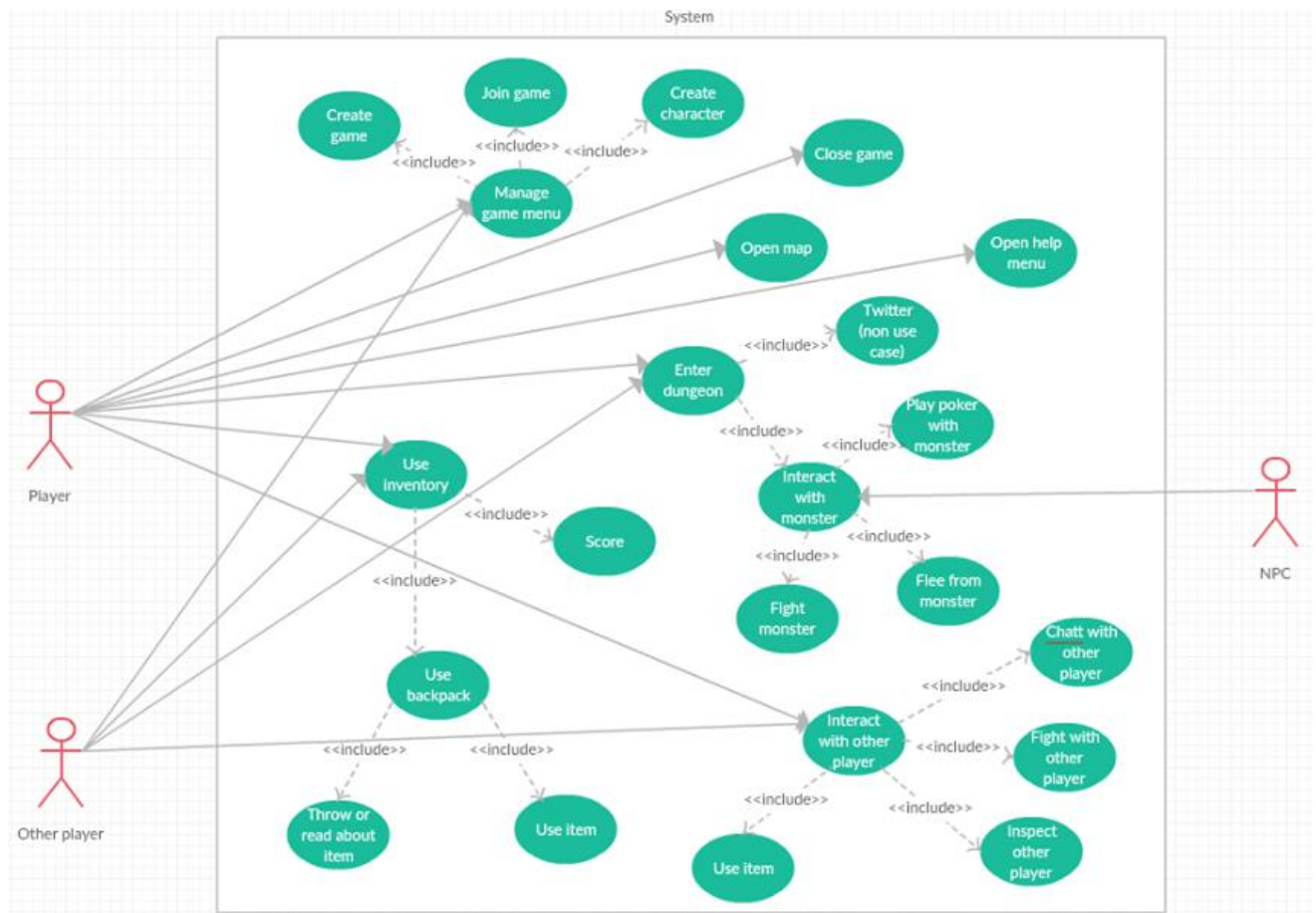
## System Description

This system is a basic NetHack game that generates its dungeons with the help of a Twitter API. The system will interact with a player and execute the player's choice/choices. The game will be structured very simply by first opening up a game menu where the player can create his/her's character and create a new game or join an already existing game. When the player have entered a game at least one dungeon will appear on the screen in which the player can enter if he/she chooses. When the player enters a dungeon a monster will appear inside the dungeon. Here the player can choose between several interaction alternatives. The interaction alternatives with the monster are *Fight the monster*, *flee* or *play poker*.

The system that we are constructing will also include a multiplayer mode which will be activated when two or more players are in the same game and in the same dungeon. The multiplayer mode will contain a *dungeon chat* where the players that are in the cave can communicate. In the multiplayer mode the players will be able to interact with each other through the items that the system contains. The item types in the game will be weapons and potions. For example, if a player chooses to use a weapon on another player he will inflict a certain amount of damage on the targeted player. If the player chooses to use potions on another player then the other player will gain health/mana points based on what potion is being used. A player will also be able to use potions on himself to gain health/mana points. The player should be able to exit the game at any point. When the player has chosen to exit the game, the player's character is saved and the game closes.

The goal with our system is to create a functioning TwitterNethack game. We have created use cases for this system by meeting the minimal requirements for the game.

# Use Case Diagram



## Description of Actors

**Player** is the actor that plays the game, in other words interacts with the system.  
**System** is the program/code that will execute the game's mechanics.

## High-Level Use Cases

### Use Case: Create character

**Actors:** Player

**Description:** The player must choose between different options to create a character. The player should be able to choose gender, race, type, main skill etc. The player modifies the character based on his/her desires.

**Main Course of Events:**

Actor	System
1. The actor chooses a name and gender	2. The system shows different characters which have the gender chosen.
3. The player can now choose which race the character should have. For example troll or elf.	4. The system is changing the skin and form of the character
5. The player now have to choose a type of character, for example wizard or hunter.	6. The system gives different alternativ of main skills
7. The player chooses main skill.	8. The system creates the character

**Alternative Flow of Events:**

The system generates characters and the player just must choose one.

### Use Case: Join game

**Actors:** Player, system

**Description:** In the menu, the player can choose to play in multiplayer mode and play the game with other players. The system presents available games and the player chooses which one he/she wants to enter.

**Main Course of Events:**

Actor	System
1.1 The player chooses multiplayer	2.1 The system shows available games
3.1 The player choose game	4.1 The system loads the game

**Alternative Flow of Events:**

The multiplayer game that the player is trying to join is currently full, if this is the case the system will display this with a message.

## Use Case: Create game

**Actors:** Player, system

**Description:** In the menu the player can choose to create a new game. The player chooses the option to create a game, the system creates the new game.

### Main Course of Events:

Actor	System
1.1 The player creates a new game	2.2 The system creates and loads a new game

### Alternative Flow of Events:

The system couldn't create the game. The system will show a message to the player why the game couldn't be created.

## Use Case: Manage game menu

**Actors:** Player, system

**Description:** When the player opens up the game a game menu will appear where player can set up the needed information to enter/create a game. This use case includes Create game and Join game.

### Main Course of Events:

Actor	System
1.The player opens the game up.	2. System displays the game menu.
3. Use case: Create Character.	
4. Use case: Create/join game	

**Alternative Flow of Events:** The player closes the game menu, system shuts down.

## Use Case: *Enter dungeon*

**Actors:** Player, system

**Description:** On the game map there will always be one dungeon that the player can enter if he/she chooses. The player enters the dungeon, the system creates the dungeon with the help of a Twitter API. This use case also includes Interact with monster.

### Main Course of Events:

Actor	System
1.Player enters dungeon	2. The system creates the dungeon with the twitter API. (Name of monster etc)
3 Use case: Interact with monster	

**Alternative Flow of Events:** The system could not create the dungeon (bad connection to the twitter api), player receives a message about this.

## Use Case: *Fight with monster*

**Actors:** Player

**Description:** Player interacts with a monster and the player is given three options to choose from: fight, flee or play. Player now chooses to fight. The fight will be based on a randomizer, that randomizes who wins. The randomizer takes the players experience into account. This use case includes Pickup item.

### Main Course of Events:

Actor	System
1. Player chooses to fight the monster.	2. The system decides who wins.Monster dies and drops a special reward and generates experience points to the player.
3. Use Case: Pickup item.	

### Alternative Flow of Events:

Player dies and the game ends.

## Use Case: *Flee from monster*

**Actors:** Player

**Description:** Player interacts with a monster and instead of fighting/playing poker with the monster the player chooses to flee. It is not every time the players success to flee and instead they have to fight the monster.

**Main Course of Events:**

Actor	System
1. Player runs from monster cause the player thinks it's too hard.	2. System randoms a number between 0-100 and if the number is below 80 the player succeeds to flee from the monster.

**Alternative Flow of Events:** Use Case: Fight monster.

## Use Case: *Play poker with monster*

**Actors:** Player, system

**Description:** When player face a monster he may choose to either fight it, flee from it or play poker with it. If the player lose they have to fight the monster. But if the player wins he/she can go away unharmed.

**Main Course of Events:**

Actor	System
1.Player choose to play poker with the monster.	2. The system is randomizing who is winning the poker game with a 50 percent chance of winning for either part.

**Alternative Flow of Events:**

The map is always shown and the player can see its position.

## Use Case: *Interact with monster*

**Actors:** Player, system

**Description:** When a player enters a dungeon he/she can choose to interact with the monsters in different ways. The alternatives are fight with monster, flee from monster or play poker with monster.

**Main Course of Events:**

Actor	System
1.Player chooses in which way he/she wants to interact with the monster.	2.1 Use Case: Fight with monster 2.2. Use Case: Flee from monster 2.3 Use Case: Play poker with monster

## Use Case: *Pick up item*

**Actors:** Player, system

**Description:** In the game the player can encounter different items. The player can choose to pick it up or leave it. If the player picks it up the item will be shown in the player's inventory and can be used later.

### Main Course of Events:

Actor	System
1.Player finds the loot from the killed character.	2. The system presents the dropped items in the general chat.
4.The player chooses to pick up the items.	3. The system puts the dropped items in the player's inventory.

**Alternative Flow of Events:** The active player's inventory might be full, if so this will be presented in the general chat.

The active player may choose to not pick up the items, the items will remain until the player leaves the dungeon.

## Use Case: *Throw item or read about item*

**Actors:** Player, system

**Description:** The active player clicks on a item and can choose to drop his/her's item to make space in their inventory or they can choose to read about the item. Player chooses to read about item, system presents the information. Player chooses to throw out the selected item, the item destroys (If no other player is in the dungeon).

### Main Course of Events:

Actor	System
1.Player chooses to throw out a specific item on the ground. 1.1 Player chooses to read about the item.	2. The system registers what item/items that are dropped. 2.2 The system presents information about the item.
	3. The dropped items are destroyed.

**Alternative Flow of Events:** If there is another player in the same dungeon as the active player the item will not be destroyed, so that the other player can pick up the dropped item.



## Use Case: *Use item*

**Actors:** Player, other player, system

**Description:** The active player can choose to use one of the existing items in his/her's backpack on a specific target. Depending on whether the item is of a weapon type or a potion type the item will either drain health from the target or generate health/mana points to the target.

(The player can be the target)

### Main Course of Events:

Actor	System
1.Player chooses to use an item on himself, on a character or on another player.	2. The system executes the player's choice.  2.1.1 (Potion on himself) The player will gain health/mana points base on what potion is being used.  2.1.2 (Potion on other player) The targeted character/player will gain health/mana points base on what potion is being used.  2.2 (Weapon on other player) The targeted player will lose health points based on the weapon's attack damage.

**Alternative Flow of Events:** If a player uses a potion on a target that already has full hp/mana points the potion will not be used and this information will be displayed in the general chat by the system.

## Use Case: *Use backpack*

**Actors:** Player, system

**Description:** When player is in the game and presses "I" on the keyboard the players backpack opens up and all the items are shown. The player can now manage his/hers items in the backpack.

### Main Course of Events:

Actor	System
1.Player presses "I" to check the items in his/her's backpack.	2. The system presents the items owned in a backpack window.
3. Use case: Throw or read about item 3.1 Use case: Use item.	

### Alternative Flow of Events:

The player closes the backpack and the backpack window closes.

## Use Case: Opens score list

**Actors:** Player, system

**Description:** The player can choose to display a score list that includes the player's experience/health/mana points. The player chooses to display the score list, the system displays information about the active player.

### Main Course of Events:

Actor	System
1. The player clicks on "I"	2. The system generates the score list with information about the player.

**Alternative Flow of Events:** The player chooses to close the score list.

## Use Case: Use *inventory*

**Actors:** Player, system

**Description:** When player is in the game and presses "I" on the keyboard the player's inventory opens up and all equipments are shown including the health/mana points and experience points.

### Main Course of Events:

Actor	System
1. Player presses "I" to check the items in his/her's inventory.	2. The system presents equipment and player information. 2 Use Case: Backpack 2 Use Case: Score
3. Use case: Throw or read about item 3.1 Use case: Use item 3.2. Player equips item from backpack	4.2 The item get equipped in the chosen inventory slot

### Alternative Flow of Events:

If the active player is trying to equip an item that is not meant to equip, an error message will be shown that will inform the player that this item couldn't be equipped.

## Use Case: Opens *map*

**Actors:** Player, system

**Description:** When player is in the game and presses M on the keyboard a mapwindow on the game will show and present where the player is located.

**Main Course of Events:**

Actor	System
1.Player can move around and this is showed on the mapwindow.	2. The system presents where the player is located.

**Alternative Flow of Events:**

The map is always shown and the player can see it's position.

## Use Case: Open help menu

**Actors:** Player

**Description:** The player can at anytime in the game choose to open up the help menu to get information about the binded keys in the game. For example Movement: WASD and some general information about the game, how to use an item etc.

**Main Course of Events:**

Actor	System
1. Player chooses to open help menu (Button:X)	2.The system presents the help menu in a new window inside the game window.
3. Player chooses to close the help menu (Button:X)	4. The system closes the help menu window.

## Use Case: *Fight with another player*

**Actors:** Player1, other player

**Description:** If a player finds himself in a cave with other players, then multiplayer mode is unlocked and the player may fight which ever player he choose to.

**Main Course of Events:**

Actor	System
1. Player1 uses targets the other player.	2. Use Case: Use item 2.2
3.Player1 kills the other player.	12.The other player has no more Hp so the system removes the other player from the current game.

**Alternative Flow of Events:** Player1 or the other player may flee from the battle at any point of the fight but they will be punished with one level deduction.

## Use Case: *Chat with another player*

**Actors:** Player1 and other players

**Description:** If a player finds himself in a cave with other players, then multiplayer mode is unlocked and a cave-chat is automatically opened where the player can chat with the other players that are present in the cave.

**Main Course of Events:**

Actor	System
Player1 enters the cave	The System opens a chat (dungeon chat) where the player may chat with the other players
Player1 writes a question in the chat	The system is displaying what player1 was writing.
Other players answer the question in the chat	The system is displaying what the other players wrote.

**Alternative Flow of Events:**

System recognize “bad languages” and replace it with “\*\*\*\*”.

## Use Case: *Inspect another player*

**Actors:** Player

**Description:** Player can inspect another player to see that players inventory.

**Main Course of Events:**

Actor	System
1. Player right click on another player and the other players inventory will show.	2. System show the other players inventory to the player who chooses inspect.
3. Use Case: inventory.	

**Alternative Flow of Events:** The player loses connection while spectating, other players inventory closes.

## Use Case: *Interact with other players*

**Actors:** Player, system, other player

**Description:** When player enters a cave where there are other players, multiplayer interactions are enabled and the player is given a few choices in the realm of the multiplayer world.

**Main Course of Events:**

Actor	System
1.Player choose in which way he wants to interact with the other players.	2.1 Use Case: Fight with the other players 2.2 Use Case: Use item 2.2. Use Case: Chat with other players 2.3 Use Case: inspect other player

## Use Case: *Close game*

**Actors:** Player

**Description:** Close and save the game

**Main Course of Events:**

Actor	System
1. The player click on end game	2. The system counts the experience points earned.
3.1 The Player click on close 3.2 The player click on close and save	4.1 The system ends the session 4.2 The system saves and ends the session

**Alternative Flow of Events:**

4.2 The game could not be saved.