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

Name	Social Security	Contribution Thinking (in %)	Contribution Writing(in %)
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

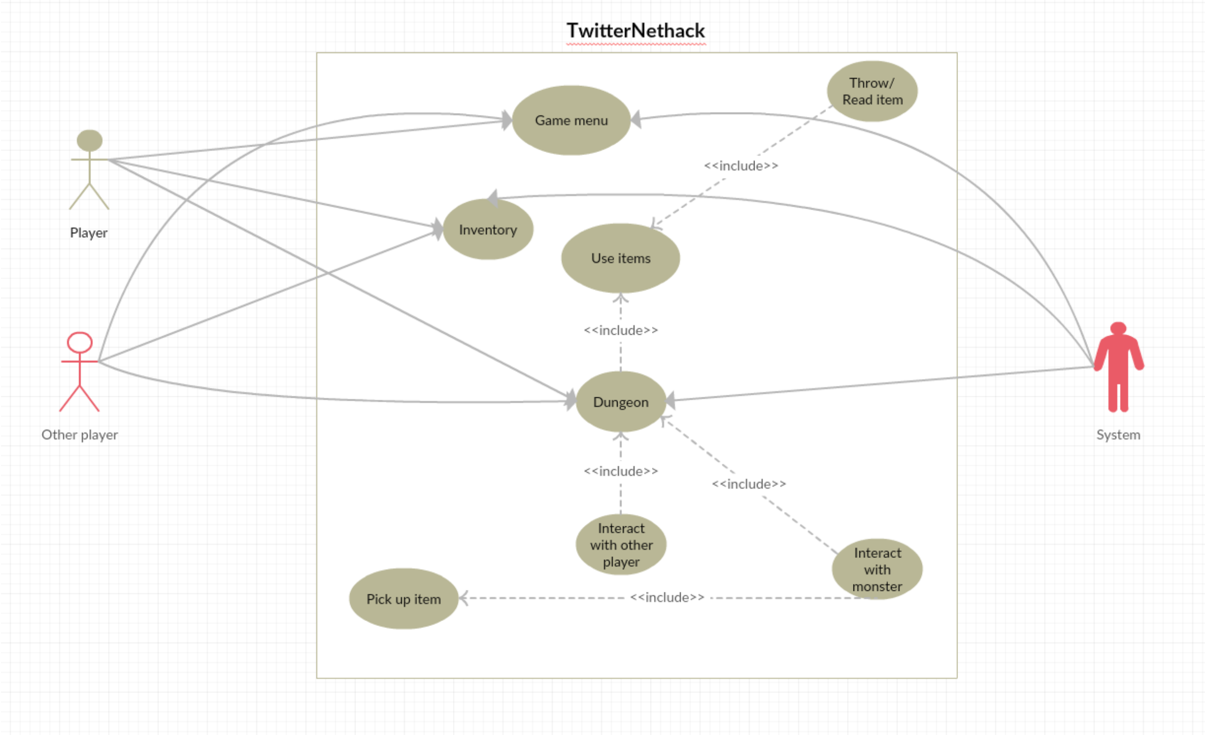
System Description

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, system
Description: The player has to choose between different options to create a character. The player should be able to choose gender, race, type, main skill etc.

Use Case: Create/Join game

Actors: Player, system
Description: In the menu the player can choose to play in multiplayer mode and play the game with others or to create a new game.

Use Case: Opens 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.

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.

Use Case: *Fight monster*

Actors: Player
Description: When a player enters a dungeon the player can choose to fight the monster to get a special drop.

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.

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.

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.

Use Case: *Pick up item*

Actors: Player, system
Description: In a character drop a player can find an item where the player can choose to pick the item up or to leave it be.

Use Case: *Throw item or read about item*

Actors: Player, system
Description: The active player can choose to drop his/her's item to make space in their inventory or they can choose to read about the 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)

Use Case: *Opens 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.

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.

Use Case: Opens *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.

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.

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.

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.

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.

Use Case: *Inspect another player*

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

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.

Use Case: *Close game*

Actors: Player
Description: Close and save the game