Use Case Overview for System TwitterNethack

Assignment in the course PA1435 Object Oriented Design

April 11th, 2017

Course project on GitHub

Author Name	Social Security Number	Thinking	Writing
Simon Lindgren	960815-1131	20	25
Peter Meunier	951230-0113	10	5
Viktor Enfeldt	960101-9616	25	25
Tobias Fast	961011-6676	25	20
Samuel Asp	960125-1458	20	25

1 System Description

A multi-player dungeon crawler that uses information from Twitter to generate rooms and to populate them with items and characters. The players will be able to freely move around in the room they're in, pick up, drop, equip, craft, and use items, attack monsters and other players, move between rooms in the labyrinth, and chat with other players.

The types of items will range from potions and rocks, to sophisticated weapons like swords and crossbows (the latter of which will require ammunition). It will be possible to craft certain items (arrows for example) by combining two or more other items.

2 Use Case Diagram
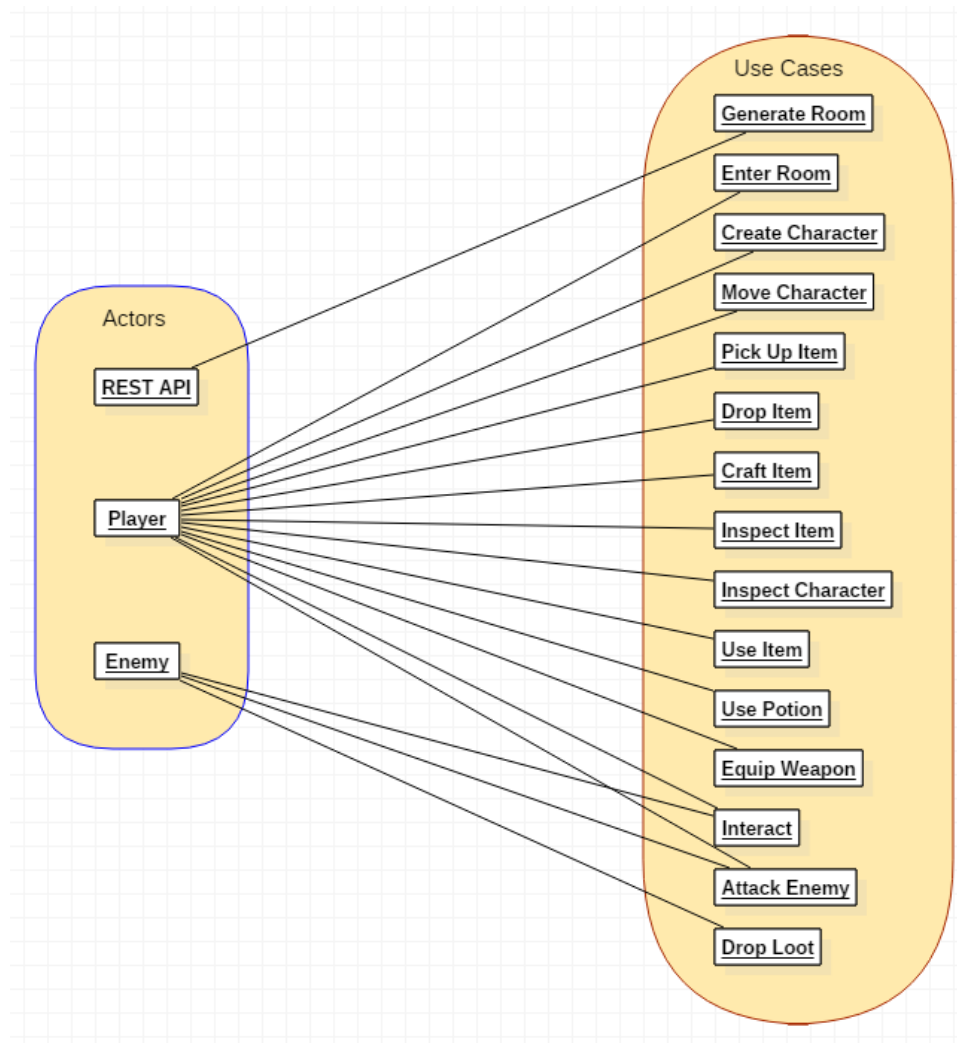
3   Description of Actors

Actor:	**Player (character)**
Description:	Character controlled by the person playing the game.

Actor:	**Enemy (character)**
Description:	System controlled opponent of the Player.

Actor:	**REST API**
Description:	External system used to get information from Twitter.

4   High-Level Use Cases

Use case:	**Move character**
Actors:	Player
Description:	Player requests to move their character in one of eight directions (north, northeast, east, etc) using the WASD keys. System moves the character in the requested direction.

Use case:	**Pick up item**
Actors:	Player
Description:	Player requests to pick up an item by right-clicking on the item with the mouse. Item is added to the player's inventory.

Use case:	**Craft item**
Actors:	Player
Description:	Player selects two or more items in their inventory to combine and turn into a new item. System removes the selected items and places the new one in the inventory.

Use case:	**Inspect item**
Actors:	Player
Description:	Player selects an item in their inventory to inspect. System presents additional information about the item.

Use case:	**Inspect character**
Actors:	Player
Description:	Player right-clicks on another character (monster, another player, shopkeeper, etc). System presents possible actions and further information about the character.

Use case:	**Drop item**
Actors:	Player
Description:	Player selects an item in their inventory to drop. System removes the item from the inventory and places it on the floor where the player is standing.

Use case:	**Attack enemy**
Actors:	Player, enemy
Description:	Player left-clicks on the area they want to attack. System deducts health from the enemy if it's within that area.

Use case:	**Use potion**
Actors:	Player
Description:	Player selects potion in their inventory from a menu. System removes the item from inventory and applies the item's effects to the player.

Use case:	**Generate room**
Actors:	REST API (Twitter)
Description:	System creates a new room based on data from a Twitter profile.

Use case:	**Enter Room**
Actors:	Player
Description:	Player requests to move their character into a new adjacent room. System places the character in the new room.

Use case:	**Interact**
Actors:	Player, Enemy/Player
Description:	Player requests an interaction with a Character (monster, another player, shopkeeper, etc). The Character Replies

Use case:	Use Item
Actors:	Player
Description:	Player selects an item in their inventory from a menu. The system applies any effect the item has.

Use case:	Equip Weapon
Actors:	Player
Description:	Player selects a weapon in their inventory from a menu. System changes the characters attributes depending on the weapon

Use case:	Create character
Actors:	Player
Description:	Player or system requests the creation of a character. System creates a character, monster, or npc.

Use case:	Drop loot
Actors:	Enemy
Description:	When an enemy is killed, they sometimes drop items and/or weapons

5 References