2017-04-10

Detailed Use Cases (Iteration 1) for System

Assignment in the course PA1435, Object-oriented 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 a game menu where the player can create his/her character and create a new game or join an already existing game. When the player has 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.

Detailed Use Cases

Use Case: Create character

Actors: Player, system
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.

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 must choose a type of character, for example wizard or hunter.	6. The system gives different alternative 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.

Preconditions: A menu of choices for the character has been implemented

Special requirements: The game can save characters after creating them

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.

Main Course of Events:

Actor	**System**
1.1 The player chooses multiplayer 1.2 The player creates a new game	2.1 The system shows available games 2.2 The system creates and loads a new game
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.

Preconditions: The user has created a character

Special requirements: The game can be saved and joined later.

Use Case: Opens game menu

Actors: Player, system
Description: When the player opens the game a game menu will appear where player can set up the needed information to enter/create a game.

Main Course of Events:

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

Preconditions: Applications that support the format

Special requirements:
1.The menu must be able to update and save after creating a character or game.
2. The application is compatible with the operative system Windows 10

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.

Main Course of Events:

Actor	**System**
1.Player enters dungeon	2. The system creates the dungeon with the twitter API. (Name of monster etc.)
3.1 Use case: Fight with monster 3.2 Use case: Flee from monster	

Alternative Flow of Events: none

Preconditions:
1. The user has created a character.
2. The user has created/joined a game.
3. The character I able to move.

Special requirements: The map gets updated

Use Case: *Fight with monster*

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

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.

Preconditions:
1. Monster has been implemented in the game.
2. The character class has functions for interactions

Special requirements: The information about the monster will be generated with a Twitter API.