

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 has to 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/hers 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 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:

2.2.The name already exists, the system provides the player with this information.

Preconditions:

1. The game menu has been started up.
2. The player has no character.
3. The player is connected to the internet.

Special requirements: The character's name must be unique.

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 The player creates a new game	2. The system creates and loads a new game

Alternative Flow of Events:

The player is not connected to the internet and the game cannot be created.

Preconditions:

1. The user has created a character
2. The user is connected to the internet.

Special requirements: The player can only be active in one game. The player's character cannot be in multiple games at the same time.

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. The 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.

Preconditions:

1. The player has started up the game.
2. The player computer supports the format of the game.

Special requirements:

The application is compatible with the operative system Windows

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.)
3.1 Use case: Fight with monster 3.2 Use case: Flee from monster	

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

Preconditions:

1. The user has created a character.
2. The user has created/joined a game.
3. The character is able to move
4. The player is connected to the internet.

Special requirements: There will only be one monster inside the dungeon.

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 decides 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.

Preconditions:

1. The user has created a character.
2. The user has created/joined a game.
3. The user has entered the dungeon.

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