

2017-04-10

# Implementation Plan 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 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.

## Prioritised List of Use Cases

The ordering of the use cases that we have made are based on the use case's importance to contribute to a minimum viable game.

To be able to construct a minimum viable product, which in our case is a game, we have ranked the importance of the use cases above. From those use cases we have chosen five of them for the first iteration and motivated why they are ordered as they are. These five use cases are considered by us, the far most important use cases that needs to be implemented/created to construct a minimum viable product. Our minimum viable product will be a simplified version of the TwitterNethack game that is being constructed. In the simplified version we still need a player that plays the game, a game that contains the player and a map that contains at least one dungeon. And to actually be able to play the game we need to implement the enter dungeon case and at least one of the several interaction alternatives between the player and the monster inside the dungeon.

Create character - 50 points

- We chose this use case as the most important because there can't be a game without a player

Create game - 75 points

- This use case represent the game creation and serve as one the most important use cases for running the game

Open game menu - 40 points

- Open game menu work as a menu in which u can either create game och create a character

Enter dungeon - 20 points

- This is the use case which creates the world in which we navigate through and generate also generates monsters.

Fight with monster - 28.5 points

- We choose this as our fifth most important use case because it lets the player fight with the monster.

Flee from monster - 60 points

Play poker with monster - 40 points

Interact with monster - 80 points

Close game - 33.5points

Pick up item - 80 points

Throw item or read about item - 63.5 points

Use item - 70 points

Opens backpack - 50 points

Open source list - 53.5 points

Opens inventory - 40 points

Opens map - 40 points

Open help menu - 30 points

Fight with another player - 50 points

Chat with another player - 50 points

Inspect with another player - 33.5 points

Interact with players in multiplayer mode - 60 points

Join game - 20 points

## Estimated Velocity Per Iteration

We have calculated that we need five iterations to implement all 21 use cases, that we described in the first document (*Use Case Overview*). We have calculated the time it will take to complete the game based on the credits received from the assignment. Each credit is assumed to take 26.7 hours and the assignment provides 4 credits, therefore we presume that the project will take 106.7 hours to complete. We convert these ours into 1067 story points, that should be finished in five iterations in a duration of five weeks. The total amount of story points divided with five gives us the number of story points per iteration and week ( $1067/5=213.5$  story points per iteration). We divide 213,4 with five to get the number of story points each member should complete per week( $213.5 / 5=42.7$ ). Finally we divide 42.7 with seven to get the number of story points a day for each member(6.1 story points per day).

- Length per iteration: 7 days
- Number of developers: 5
- Full velocity of each developer: 6,1 SP per day
- Best scenario per iteration: 213,5 SP
- Worst scenario per iteration:  $(213.5*0.5)= 106.75$  SP
- Average scenario per scenario:  $(213.5*0.75)=160.1$  SP

## Implementation Plan

### First iteration:

Create character - 50 points

- We chose this use case as the most important because there can't be a game without a player

Create game - 75 points

- This use case represent the game creation and serve as one the most important use cases for running the game

Open game menu - 40 points

- Open game menu work as a menu in which u can either create game och create a character

Enter dungeon - 20 points

- This is the use case which creates the world in which we navigate through and generate also generates monsters.

Fight with monster - 28.5 points

- We choose this as our fifth most important use case because it lets the player fight with the monster.

### Second iteration:

Flee from monster - 60 points

Play poker with monster - 40 points

Interact with monster - 80 points

Close game - 33.5points

**Third iteration:**

Pick up item - 80 points

Throw item or read about item - 63.5 points

Use item - 70 points

**Fourth iteration:**

Opens backpack - 50 points

Open source list - 53.5 points

Opens inventory - 40 points

Opens map - 40 points

Open help menu - 30 points

**Fifth iteration:**

Fight with another player - 50 points

Chat with another player - 50 points

Inspect with another player - 33.5 points

Interact with players in multiplayer mode - 60 points

Join game - 20 points

All the cases in the first iteration combined defines our view of a minimum viable product and as we described earlier the first iteration will represent a simplified version of the complete TwitterNethack game. To be able to start on the other iterations we first need to complete the first which in our case is the base of the whole game. We can exemplify it in this way, if we are building a house the first iteration will then be to build up the walls and the roof of the house. The other iterations will include adding specific functions to the house, for example the second iteration will be to create bed rooms in the house and the third will be to create a kitchen inside the house etc.

In the four other iterations we have divided the cases into the different iterations based on the ordering of the use cases and also tried to have the same category of use cases in the same iteration.