2017-04-10

Implementation Plan 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 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. To 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 - 100 points
Create game - 200 points
Open game menu - 100 points
Enter dungeon - 30 points
Fight with monster - 50 points

Flee from monster - 50 points
Play poker with monster - 50 points
Interact with monster - 60 points
Close game - 20 points

Pick up item - 40 points
Throw item or read about item - 50 points
Use item - 70 points

Opens backpack - 50 points
Open source list - 40 points
Opens inventory - 30 points
Opens map - 30 points
Open help menu - 10 points

Fight with another player - 50 points
Chat with another player - 70 points
Inspect with another player - 30 points
Interact with players in multiplayermode - 60 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)*. According to our estimates we have a total of 1190 story points per person to finish all five iterations. The first iteration is a total of 480 story points and sums up almost half of the time intended to this program. The first iteration is the biggest because of the sheer size and difficulty to implement the most important use cases in our program. The average story points per iteration is 1190/5 = 238. Our maximum amount of story points dedicated on a single iteration is our first because of the reasons mentioned above. The minimal amount of story points is estimated to 160 points which represent time intended to iteration three and four.

Implementation Plan

First iteration:
Create character - 100 points

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

Create game - 200 points

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

Open game menu - 100 points

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

Enter dungeon - 30 points

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

Fight with monster - 50 points

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

Total: 480 points

Second iteration:
Flee from monster - 50 points
Play poker with monster - 50 points
Interact with monster - 60 points
Close game - 20 points

total: 180 points

Third iteration:
Pick up item - 40 points
Throw item or read about item - 50 points
Use item - 70 points

total: 160 points

Fourth iteration:
Opens backpack - 50 points
Open source list - 40 points
Opens inventory - 30 points
Opens map - 30 points
Open help menu - 10 points

total: 160 points

Fifth iteration:
Fight with another player - 50 points
Chat with another player - 70 points
Inspect with another player - 30 points
Interact with players in multiplayer mode - 60 points

total: 210 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. As we can see the first iteration has a lot more story points assigned than the other iterations. The reason for this is that the first iteration is the main activity of the course and that our minimum viable product is a working 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.