Application of Greedy Algorithm in Brigandine : The Legend of Forsena

Pandu Kartika Putra 13511090
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13511090@std.stei.itb.ac.id

***Abstract*—Brigandine is a tactical role-playing game in which player choose to control a nation and try to conquer all cities in the continent by invading other nation's cities and defending his own cities against other nation's invasion. However, there are several problems that a player must think about other than invading or defending their cities, which is monster summons and selection.**

The paper will cover an analysis of greedy algorithm to solve several interesting problems in the game of Brigandine : The Legend of Forsena. The analysis will include detailed example and case study as well as thorough explanation of the application of greedy algorithm in this game.

***Index Terms*—Brigandine, Greedy Algorithm, Monster, Selection, Summons.**

I. INTRODUCTION

A. Tactical Role-Playing Games

Tactical role-playing game is one of genre in the world of gaming. Often mentioned as Strategy Role Playing Game (Strategy RPG), tactical role playing games have been one of game enthusiasts's favorite because of its unique characteristics.

A tactical role playing game is a variant of role playing game (RPG) which aside from emphasizing gameplay type of deliberately paced tactical battle where each side politely takes turns moving and attacking one another[1], it usually also incorporates a strategic gameplay such as tactical movement or exploration in an isometric grid. The combat system in this genre varies depending on the games, making each game has its own uniqueness and appeal to players.

The principle of this genre is combining the traditional role playing game (RPG) system with various forms of strategy game. Just as typical RPGs games, the player will control a certain amount of character in a party and battles enemies in a group of similar amount with various map and backgrounds.

A significance difference between tactical role-playing game and standard role-playing games is its lack of exploration aspect. In standard role-playing games, the player typically can control his character(s) to explore the map or terrain on the game to certain amount of degree. Exploration in a third person or first person point of view to caves, dungeons, and towns are free to be enjoyed by the player and become one of certain unique aspect of the game itself. The character could then meet monsters, sellers, quests, new areas, or even other players (in certain form of RPGs that feature multiplayer such as MMORPG). Regarding the battle scene, various approachs are found even in standard RPG game. There are games that use a new different screen for the battle scene and there are others that don't, obviously.

Rather than featuring this, a tactical role-playing game tends to focus on the strategy of the battle itself. Player are most of the times faced with different choices in party member to build and to be leveled up, including melee damage per second (DPS) classes such as warriors, magic DPS classes such as wizards, as well as support classes such as priests. Various combinations as well as skill and stats combination of these classes are available for players to choose and develop, depending on the game. Game mechanics such as limitation on the number of party member that can be included in one battle and types of classes that can join certain place of battle naturally make the task to maximize the potential of this party build become harder and appealing at the same time for players. These characters gain experience points on participating in a battle and execute an attack or skill (healing, strengthen, or other support skill, for example). This experience points will usually be then used for advancement in levels and classes.

The battle itself typically has certain winning conditions, including but not limited to defeating all the enemies, surviving until certain turns, or simply to reach certain tiles in the map. Winning a battle will results an advancement in the game, be it an availability of new map, a conquering status of a city, or a completion of a quest. When players are not in battles, just as standard RPG does, a tactical RPG games let players to do a lot of other features such as sell items found on the quests/battles, buy items, weapons, as well as armors for the characters, and even change classes of specific available characters, all depend on the game.

B. Brigandine : The Legend of Forsena

Brigandine : The Legend of Forsena is a once popular game of tactical role-playing game genre. The game takes place in the land of Forsena, a region that was until recently wracked by years of war. Unfortunately, a prominent admiral named Zemeckis who once was an important figure that deliver victory to the Almekia kingdom decided to betray the King, capture his lands, and use them as a base to conquer the continent[2].

The player control a chosen nation from five different choices. The choices are different in nature and have their own implications to it, such as the number of initial knight, initial monsters, initial region that are conquered, and so on. Picture 1.1 shows the initial state of one of the nation.

Picture 1.1 Initial stats of one nation

Once a player choose his nation, he will be in control of a number of cities that become a garrison for a certain number of Knights, which in the previous section on this paper are typically called characters. These characters, as mentioned in previous section, have their own classes and stats that can be developed from battles and quests. Player can assign unlimited number of these characters to a city, although only a party of 3 member are allowed to join a battle, whether it's defending or invading.[3]

However, Brigandine : The Legend of Forsena also features monsters as subordinates of these characters. These monsters can be summoned through the use of Mana, and these mana points can be obtained from cities each game turn (not battle turn). These monsters, just as the characters, have their own levels, stats, skills, and even classes. Although the classes of these monsters are unique and different in each monsters type, the advancement system is similar with the character's classes. The monsters types are varies greatly, from a support type such as fairy, a melee DPS type such as golem, until a magic DPS type such as demon. These monsters can be assigned to the characters as their subordinates, and these assignments depend on the characters mana pool and each monsters mana costs. The task to assign monsters to characters often confuse the player, because there are a lot of limitations present, such as amount of monsters, characters mana pool, and monsters mana costs. This problem will be covered later in this paper. Picture 2 shows how the process look like. The leader is the characters, and the monsters are their subordinates.

Picture 1.2 Assigning monsters to characters

Once the player finish organizing the nation, such as assigning monsters to the characters and assigning characters (and the monsters as their subordinates) to the cities, player can finish their organizing turn and start the attacking turn. In this turn, player can invade other nation's cities that adjacent to their border city with the party that reside in the mentioned border city. The game will then change screens to a unique battle screen for each city that are invaded/defended, one by one per battle.

Picture 1.3 Battle screen

After the battle win or lose, player will gain or lose cities, hence expand or shrink his domain in the world map.

Picture 1.4 Domain Map

II. THEORIES

A. Greedy Algorithm

Greedy Algorithm is a variant of algorithm that is widely used to solve optimization problems. Typically, these optimization problems are further divided into two type which is maximization and minimization problems.

The main principle of Greedy algorithm is "take what you can get now", which means that the algorithm will choose (or pick) the best option that is available at the moment. Another aspect of this algorithm is that it always hopes that by taking the best option at each step (local optimum), the best solution will be reach (global optimum). [4]

Three main characteristics from the algorithm is :

a. It is a done step by step
b. Always taking the best option at the moment
c. Hopest that picking all of the local optimum will results in global optimum at the end.

Formally Greedy algorithm can be stated by 5 important elements, which are:

1. Candidate set, C,
   The candidate set (C) is the set of all the elements that are possibly will be the solutions. An element of this set may be a set as well. At each step in the algorithm, an element is taken from this set and inserted into solution set (S).
2. Solution set, S,
   The solution set (S) is the set of elements that are selected throughout the process of algorithm. This set, at the end of the process, will represents the solution of the problem. As its elements are taken from the candidate set (C), solution set is a subset of candidate set.
3. Selection function,
   Selection function are the function that assigns values to solution candidates in the candidate set and selects a candidate that has the best value (in terms of the problem instance, maximum, for example) and removes it from candidate set. The selection function of greedy algorithm can be more than one, depending the problem instance.
4. Feasibility function, and
   Feasibility function is the function that tests whether the candidate selected by the selection function is feasible to put to the solution set. Being feasible means that when it's combined with other candidates in the solution set, the new candidate does not violate any constraints that exists. A new candidate will be taken from the candidate set by the selection function if the existing candidate is not feasible to put to the solution set. The process are to keep going until no elements are left in the candidate set or until a certain goal or limit is reached by the solution set.
5. Objective function.
   The objective function is the function that become the goal of the solution of the problem. Typically, objective function can't be expressed in the code, but rather become the global goal of the code itself. It makes the best global solution (maximum or minimum). [4]

The overal scheme of greedy algorithm is :

a. Create an empty set as a solution set
b. While solution doesn't reach its goal and the candidate set is not empty, execute following steps
c. Remove a candidate from the candidate set by selecting it with selection function
d. Test whether the selected candidate is feasible (with feasibility function), and add it to the solution set if it pass.

One thing to remember about greedy algorithm that the fact that it doesn't always yield the best solutions. The reason behind this important characteristics of greedy algorithm is that :

1. Greedy algorithm seeks for local optimum without considering the consequences toward the global solutions, it only hopes that it will lead to the global solutions.
2. As mentioned before, in any problem instance there may be more than one selection function available. The selection function we choose for the algorithm will affect the global solutions that are made.

Finally, although greedy algorithm has a weakness that are mentioned above, this algorithm is still popular and widely used because of its simplicity and efficiency in implementation, particularly in optimization problems.

III. ANALYSIS OF PROBLEMS

In the game of Brigandine, there are a lot of problems that can be solved with certain applications of algorithm, such as the routes to conquer the map, the knight selection, the monster selections, and many others. This paper will cover one of the problems, which is monster selection problems, in terms of balancing it among the characters.

As it has been mentioned before, monsters in Brigandine : Legend of Forsena can be assigned to characters. It is then become a problem for the player since the amount of mana points of each characters are limited and different, and the amount of mana costs of each monster is different. The sum of mana costs of monsters tha are assigned to a character cannot be more than the mana pool of the character.

To solve this problem of balancing the monster assignment among the characters, we can approach the problem with picking monster that has least mana cost.

If there are monsters that have the same mana costs, then pick the monster that has the highest level among them. If there are still more than monsters that qualify, then pick monster that has the most health points.

For the assignment process, we will iterate through the characters, starting with the one that has bigger mana pool first. The assignment will end once all the monsters has been picked and tried to be assigned. There are several approach for the assignment process, such as picking the characters that has less mana pool first, but that is beyond the coverage of this paper.

IV. APPLICATION

First, we will have a set of problem instance for the application. Suppose we have three characters with each stats :

1. Dinadan (Lvl 22)
   Class : Paladin
   Mana Pool : 300
2. Felicia (Lvl 15)
   Class : Lancer
   Mana Pool : 202
3. Irvin (Lvl 7)
   Class : Lancer
   Mana Pool : 176

We will also provide monsters to be assigned :

1. Dragon1A (Lvl 1)
   Mana Cost = 85
   HP = 100
2. Dragon1B (Lvl 2)
   Mana Cost = 87
   HP = 115
3. Dragon1C (Lvl 1)
   Mana Cost = 85
   HP = 103
4. Golem1A (Lvl 3)
   Mana Cost = 48
   HP = 120
5. Golem1B (Lvl 2)
   Mana Cost = 45
   HP = 110
6. Fairy1C (Lvl 1)
   Mana Cost = 30
   HP = 50
7. Centaur1D (Lvl 1)
   Mana Cost = 35
   HP = 70
8. Djinnie1E (Lvl 1)
   Mana Cost = 35
   HP = 60
9. Wyvern2C (Lvl 1)
   Mana Cost = 60
   HP = 80
10. Wyvern3B (Lvl 1)
    Mana Cost = 60
    HP = 83
11. Pegasus2A (Lvl 1)
    Mana Cost = 45
    HP = 55

In this application, the picking of each object in detail will not be explained because the focus of the section is to see the result of the greedy algorithm mentioned.

a. Greedy by max monsters mana cost

In this greedy algorithm, we will choose monsters that has biggest mana cost first, and then iterates to try if it fits in a character, start with a bigger mana pool ones.

Candidate set :
Every monster in the problem instance.

Solution set ;
Monsters in assignment respective to the available characters.

Selection function :
The monster with biggest mana cost first will be chosen.

Feasibility function :
Test whether the monster can still be placed in the characters, start with the bigger mana pool ones.

Objective function :
Have the maximum mana points used in each characters to holds monsters (most close sum of total mana costs of monsters that are held with the mana pool).

The solution set is

Dinadan :
Dragon 1B (Mana Cost = 87)
Wyvern3B (Mana Cost= 60)
Golem 1B (Mana Cost=45)
Fairy1C (Mana Cost=30)
Total Mana used = 232

Felicia :
Dragon1C (Mana Cost = 85)
Wyvern2C (Mana Cost= 60)
Centaur1D (Mana Cost=35)
Total Mana used = 180

Irvin :
Dragon 1A (Mana Cost =85)
Golem 1A(Mana Cost = 48)
Djinnie1E (Mana Cost =35)
Total Mana used = 168

V. CONCLUSION

From the description, examples, and analysis in previous sections, we can summarize some points :

1. Greedy algorithm for the monster assigning problems work quite well in terms of balancing the monster and the mana costs. It works well if the number of monster is greater because the choice varies more.
2. In solving the moster assigning problems, the focus of the algorithm is not the semantic of the monster, such as its potential use and unique characteristics in the battle field. This lead to unbalanced proportion on the monster that are assigned in the characters.
3. The most effective result shown in the character that has least mana pool, which only throw away 8 points of mana points that are not used.

VII. ACKNOWLEDGMENT

Pandu Kartika Putra as the author wants to express his grateful thanks to God for His blessing during the process of making this paper, deepest gratitude to Mr. Rinaldi Munir and Masayu Leylia Khodra, ST., as the lecturers of IF 2211 – Strategi Algoritma, and my superb family which always strengthen my spirit, as well as my awesome friends in ASCII 2011, IF and STI ITB 2011.

REFERENCES

[1] ___. "Tactical Turn-Based Combat". December 19, 2013 (18.30 PM). <http://www.giantbomb.com/tactical-turn-based-combat/3015-234/games/>

[2] ____. "Brigandine : The Legend of Forsen". December 19, 2013 (19.23 PM). <http://www.ign.com/articles/1999/03/05/brigandine-2>

[3] ____. December 19, 2013 (20.02 PM). <http://www.forsena.org/Brig/Countries/index.htm>

[4] Munir, Rinaldi, "Diktat Kuliah IF3051 Strategi Algoritma", Program. December 19, 2013 (21.00 PM).

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 20 Desember 2013

Pandu Kartika Putra, 13511090