Greedy vs Greedy in Dragon Age: Origins

Rido Ramadan – 13509049
Informatics Engineering
School of Electrical Engineering and Informatics
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13509049@std.stei.itb.ac.id

***Abstract*—In a large scale game world, it is always beneficial to have a simple but efficient artificial intelligence inside enemy. Greedy algorithm is a simple approach in making a simple but efficient AI based on several rules. Dragon Age, developed by Bioware, was one of the highly successful games which applied greedy mechanism inside the enemy AI. The greey itself run by seeking player character which has highest Threat Level. Can the player defeat the enemy CPU greedy by using another personal greedy? Yes, it is. Another greedy can be used to counter the enemy CPU greedy.**

***Index Terms*—Greedy, Dragon Age, Artificial Intelligence, Threat Level.**

I. INTRODUCTION

Gaming industry has shifted from a simple, single logic objective into a huge game world, a lot of interacting objects, and automated self-managing objects. In order to mangaed those interacted object, they are controlled via algorithm mechanism. People usually called this mechanism as AI, abbreviated from Artificial Intelligence.

Dragon Age: Origins is one of the most successful game in 2009. It was even nominated and won the title of Game of the Year 2009 (GotY 2009). Dragon Age: Origins is a Role-Playing Game with strategy elements included. The sequel, Dragon Age 2 was released in April 2010 with similar mechanism.

As an RPG, it is normal to have a huge world populated with people and enemies. Both player and the enemies in Dragon Age can roam in the world freely.

A bit about Dragon Age, it is a game in a medieval time where there lived three races. They are human-which dominating the whole surface world, dwarves-short people which lived deep below the surface, and elves-pointy-ears people whose life enslaved by human, or choose to live far from humanity.

In addition to races, there are three major classes. These classes heavily impact on how the AI in the game would react inside battle. They are Warriors-job for those who prefer melee and protected behind heavy armor, Magician-job who prefer long-range magic due to their weakness in physical contact, and Rogue-job for those who think that the speed is everything.

The enemies are varies, from a lowly animal like rats, bandits, wizards, spirits, monsters, even the mighty dragons. Even though there are so many variety, they all have the same tendency, that is always prefer those who meet this criteria: has the strongest armor, closer to current CPU position, and deal most damage. Through observation and experience playing Dragon Age, it can be assumed that the enemies' AI implements greedy algorithm.

Now, the question is, in such a complex mechanism inside a game like Dragon Age, why do the developer choose greedy algorithm?

Greedy, by far, is the easiest optimation algorithm to implements in a system. Although it cannot always return the most optimal solution, it can return result which is almost optimum.

AI controlled by greedy has an obvious behavior. It will always target the same enemy, or do the same thing over and over until the condition met is not satisfied anymore. Although some randomizer may be implemented in that kind of AI, it is still safe to assume that enemy or CPU has a greedy mechanism.

Greedy, though simple, is quite efficient to reach the desirable result. Even a simple algorithm like greedy can be tailored into a beautiful automated system by complexing the mathematical formulation behind selection function.

Now, also a new question appeared. Can a greedy approach done by human playing Dragon Age can defeat the CPU AI? We will see if some algorithm can be derived to solve this problem.

II. GREEDY ALGORITHM

Greedy algorithm, also known as greedy, is a step-by-step problem solving algorithm used in optimation problem. It is a straighforward approach to solve problem, because in sense it only take decisions on the basis of current information without worrying what will happen in the future.

Greedy is a simple algorithm which only take the most profitable path to reach the optimal solution. Because of this behavior, it is named greedy. In the runtime, greedy algorithm always choose local optimal solution in hope that by always taking local optimum, it can reach the global optimum.

However, sometimes a hope is just a mere hope.

Greedy doesn't always return the optimal solution. Even though it doesn't awalys get the real optimal result, usually the result given from greedy algorithm is desirable. More over, due to the fact that greedy algorithm usually reduce the duration taken by brute force algorithm in finding solution, many people use this approach during solving a problem.

Since greedy is a starightforward algorithm, it cannot forecast what will happen next, whether it is beneficial or not.

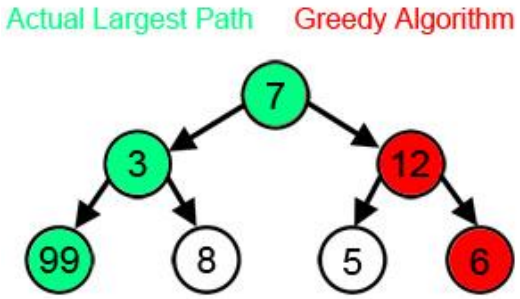

Fig. 1 How greedy fail to get the optimum

Based on the statements above, greedy can be described as such:

- An algorithm which solve a problem step by step.
- An algorithm which choose the current best path without worrying the future consequences
- An algorithm which follow the heuristic that state current best choice taken will eventually lead to the global optimum

Greedy is composed of:

- Candidate set, a set consists of existing elements from which a solution is formed.
- Solution set, a set consists of elements chosen from candidate set which is assumed as an optimal solution.
- A selection function, which chooses the best candidate to be added to the solution.
- A feasibility function, which is used to determine if a candidate can be used to contribute to a solution.
- An objective function, which determine what needs to be optimized.

III. IN-GAME TERMS AND GLOSSARY

First thing first, it is important to know several basic mechanism inside the Dragon Age: Origins before learning the background process behind AI behavior.

Some glossary related to Dragon Age: Origins are explained below. You may skip this part if you are already familiar with, but if you are not an expert, it is highly recommended for you to read this glossary first.

Glossary	Means
PC	Player Character, the in-game character which directly controlled by human player.
NPC	Non-playable Character, in-game uncontrollable character.
Enemy or CPU	CPU-controlled enemy, always hostile to the PC.
HP	Hit Points, number which represents a character/enemy's life and how much damage can one withstand before dying. Deplete with each attack received.
MP	Mana Points, represents pool of magic power. Deplete with each magic use.
Stamina	Stamina represents power to do some ability. Depletes with each abitily use.
Attack, Magic Attack or Damage	Represents a number which directly substract a character's HP.
Armor Rating Defense, or Magic Defense	Represents a number which can reduce the damage received by a character.
Evasion Rate	Represents probability of one completely dodges incoming attacks. A dodged attack will result miss and no damage dealt. (No HP reduce)
Party	A group of player characters, capped at four people only. Can consists of multiple classes.
AoE	Area of Effect, spherical area where an ability effect is applied within it's radius.
Class	A term that represents job for a character. Each class has distinctive characteristics.
Tank or tanker.	Sacrificing a PC as a decoy for receiving most of the incoming enemies' attacks. Also known as meat shield.
Monster rank	Self-explanatory, there are five enemy ranks in game, ordered from the weakest to srongest: 1. Critter, weaker than average party, they pose no challenge to party, unless they come in a huge number. 2. Normal, slightly weaker to average party, they pose little challenge to party. 3. Lieutenant, as strong as average party. They usually lead a pack of normal rank enemies behind. May pose some challenge as they slightly harder to kill than any average enemies 4. Boss, stronger than party, usually on of these can easily obliterate unprepared party in no time. Challenging and need preparation. 5. Elite Boss, super boss rank.

	Specifically design to pose a highly challenging battle to party. They have unusually very high HP and can even obliterate prepared party easily.
Threat or aggro	Aggressiveness factor, number which attracts enemies to attack. The higher the threat the character has, the higher the chance that the character will be swarmed by enemy. Directly affect the enemy AI.

Table 1 Glossary in Dragon Age: Origins

In game there are three major races each with it's own characteristics. Those are:

1. Humans
   A Human usually has high defense.
2. Elves
   An Elf usually has higher magic attack, weaker defense, and less HP.
3. Dwarves
   A Dwarf usually is a powerhouse, has more HP than Human, higher defense and magic defense, but slower and lower evasion rate.

Now, take a look at three classes. The explanations are as follow:

1. Warriors
   A Warrior is the powerhouse, has a lot of HP, ability to wear Heavy Armor, which result in highest defense as possible, and capable in dealing heavy damage to a single enemy. Usually become tank for the entire party.
2. Rogue
   A Rogue is an agile attacker. In sacrifice for better evasion rate, Rogue has less HP than warriors and only able to wear light armor, which provide less defense than heavy armor.
3. Mage
   A Mage is a fragile class. Mage can deal very large damage to a group of enemies, but ultimately is the weakest classes in the Dragon Age: Origins world. Cannot withstand damage as much as a Rogue does, and die easily when swarmed by a group of enemies. Has the least HP and defense in the game.

For convenience, classes has much more significant effect in gameplay rather that races. So, in the overall gameplay, the only worthy thing to be considered in a party is the classes which compose the entire party. To sum up, Warriors are the highest defense and the main tanker in the group, Mages are the weakest in term of defense, but the deadliest in term of attack, and Rogue stand between the two of those.

IV. Analysis on Greedy Inside Enemy's AI

In gameplay, most of the time player will control a party consists of four people. When at least an enemy sighted, the game will pause, and player can queue multiple actions to PC which attacks which. Each enemy will also automatically choose which PC will it attacks. The battle begins when player decide to unpause the game. Then, both sides will clash into each other. PCs will automatically attack according to the tactics defined in Tactics Menu, although player can manually directs which PC will do what to who.

In multiple observation, most enemies initially target the same PC, which most likely is a Warrior. At first, Warrior will be the meat shield while let the rest of team attack from afar or flanking them. In a prolonged battle, sometimes a mage might be the focus of the attacks.

Let's study this party setup and battle strategy:

- Edgar
  Mage, offensive, AoE massive damage dealer.
  Equipment: Robes of Avernus
  Defense rate: 0.
- Alistair
  Warrior, tanker, high defense, aggro puller.
  Equipment: Blood Dragon Armor set
  Defense rate: 27.31.
- Leliana
  Rogue, archer, long range attacker.
  Equipment: Wade's Dragonskin Armor set
  Defense rate: 15.01.
- Wynne
  Mage, healer, mainly healing the entire party.
  Equipment: Senior's Enchanter Robe
  Defense rate: 0

For the basic strategy, let's devise a tactics as below:

Tactics:

- Have Alistair go to frontline, closer to enemies than any other party.
- Have Edgar and Leliana stand behind while attack from afar.
- Have Wynne periodically heals Alistair to prevent him from dying.

Lets use this tactics against a Revenant Battle. A Revenant Battle consists of a Revenant (Boss rank, very strong and hard to kill), a skeleton archer, and two skeleton warriors.

Result:

- Alistair will be the main focus of enemies attacks. Revenant and two skeleton warriors will engage him right on. The skeleton archer may or may not.
- Any other party will attack from afar in a safe distance, assuming they will not be the target of the enemies attack.

Looking from the fact above, it is clearly that enemies is more attracted to Warriors rather than any other class. The enemy CPU maybe has a simple algorithm only to decide which enemy should they attack (in case, from the example above, choose Warriors first), or maybe it has a more complex system than that. Let's revise the strategy as below:

Tactics:

- Have Alistair go to frontline, closer to

enemies than any other party.
- Have Edgar and Leliana stand behind while attack from afar.
- Have Wynne periodically heals Alistair to prevent him from dying.
- Let Edgar cast Storm of the Century, the most powerful AoE magic.

Result:
- At first, every enemy, including the Revenant will be interested to attack Alistair.
- After Storm of the Century is casted, any enemy survivor will try to attack Edgar.

Well, it is undesirable that Edgar become the main target. It can be concluded that the enemy CPU has much more complex AI system to determine which PC should they attack. Let's revise the strategy again as below:

Tactics:
- Have Alistair go to frontline, closer to enemies than any other party.
- Have Edgar and Leliana stand behind while attack from afar.
- Have Wynne periodically heals Alistair to prevent him from dying.
- Let Edgar cast Storm of the Century, the most powerful AoE magic.
- If Edgar is attacked, have him casts Mind Blast to stuns any enemies around him.

Result:
- At first, every enemy, including the Revenant will be interested to attack Alistair.
- After Storm of the Century is casted, any survivor will try to attack Edgar.
- After the Mind Blast is casted, any enemies around Edgar may be stunned. Although it is impossible to stun Revenant, it can divert Revenant attention from attacking Edgar.

The result above show that some skills may divert the enemy attention, some may attract them to one person, some may make them lose interest in one person.

From those three experiments, we can assumed hypotheses like below:
1. Enemy prefer Warriors over any other classes.
2. Enemy prefer PC who are close to them.
3. Enemy prefer PC who has dealt major damage.

Three statement above are just hypotheses. Let's try another experiment regarding enemy preferences. Make a party consisting these people:
- Edgar
  Mage, offensive, AoE massive damage dealer.
  Equipment: Robes of Avernus
  Defense rate: 0
- Alistair
  Warrior, tanker, high defense, aggro puller.
  Equipment: Blood Dragon Armor set
  Defense rate: 27.31
- Leliana
  Rogue, archer, long range attacker.
  Equipment: Wade's Dragonskin Armor set
  Defense rate: 15.01
- Sten
  Warrior, tanker, high defense, high attack, stunner.
  Equipment: Legion of the Dead Armor set
  Defense rate: 31.89

Since this time there is no healer, the strategy deployed will be different.

Tactics:
- Have both Alistair and Sten go to frontline, closer to enemies than any other party.
- Have Edgar and Leliana stand behind while attack from afar.
- If either of Alistair or Sten's HP falls below 50%, have Leliana throw potion to restore some HP.
- If Edgar is attacked, have him casts Mind Blast to stuns any enemies around him.

Since this time Edgar does not cast his mighty Storm of the Century, the result this are as follow:

Result:
- Enemies prefer to attack Sten rather than Alistair, except the one whom Alistair engage.
- Both Edgar and Leliana will do their usual long range attack.
- While Sten is currently handle another enemy (e.g. skeleton warrior) and Revenant currently attacking Sten, if Alistair attacks the Revenant, then Revenant will change target to Alistair.
- While Alistair and Revenant are clashing, if Sten attacks the Revenant, then Revenant will change it's current target back to Sten.
- Revenant prefer Sten rather than Alistair, despite the two of them are attacking it from front.

The experiment above compares between two different defense rate Warriors, which one that will be picked by the enemies as their main target. From the result above, it is clear that enemy will prefer those who has higher defense rate.

To sum up the four experiments above, the enemy CPU has tendency to target PC who meet these criteria:
1. Has the highest defense
2. Closer to some enemies' current position
3. Has dealt major damage to many enemies
4. Has the highest factor X than any other PC.

What is factor X all about? Is this factor X dependent to the three other factors? According to contributors in Dragon Age Wikia, Bioware called this factor X as "threat level" or simply "threat".

Threat is defined as a mechanism in Dragon Age Series whereby enemies decide which characters in your party to attack. Threat level is a number which represents the enemy hostility and tendency to attacks.

Threat directly affects the enemy CPU. A PC who has

higher threat than any other PCs is most likely to be targetted by enemies. Threat is affected by many factors, some of them will lower the threat number, some will rise the threat number.

How to calculate threat level for each PC? The formula is not that complex, but the condition which affect the threat level are many. Based on Dragon Age Wikia, every PC initial threat is 10. The closer they are to enemy, the higher the threat level increase. Plus, equipment also determine the threat level, such as light armor adds extra 5 points to threat level, and medium, heavy, massive armor adds extra 10 points. These three components are major point in threat level which cannot be cleared by any skill in the game. In addition to those three, some skill affect threat level, such as Mage's Mind Blast clears the threat level to distance-based only, Warrior's Taunt adds extra 300 points, and so on. Damage dealt to enemy also affect threat level preceived by the victim, so powerful spell like Storm of the Century will increase the threat level.

What is the relation between threat level and greedy algorithm? It's just that we can say that the developer made the enemy CPU to target PC by implementing "Greedy by most threathening". Since greedy is a step-by-step decision problem solving algorithm, each time interval in game runtime is a step in greedy algorithm. Elements in this "Greedy by most threatening" explained as follow:

- Candidate set: each PC in party.
- Solution set: a PC in party which is chosen by the enemy.
- Selection function: a PC who possess the highest threat level.
- Feasibility function: is the PC is still alive in the battlefield?
- Objective function: maximize the threat which a PC has.

In term of pseudocode, we can assume that the enemy CPU has something simlar like this.

```
function
getNextTarget(C:{Edgar,Alistair,Leliana,Sten}) → PC
variables
  S:PC
  i:integer
algorithm
  i ← 0
  if (Ci is not dead) then
    S ← Ci
  else
    i ← i + 1
  endif
  C ← C - {Ci}
  while (C is not empty) do
    if (Threat Ci > S) and (Ci is not dead) then
      S ← Ci
    endif
    C ← C - {Ci}
  Endwhile
```

Then, while the battle still ensues, each enemy will attack the PC according to result gotten from the function above. For each clock tick and PC activity, enemy will reevaluate which PC will be the next target. Repeat this function until either every PC dies or the whole group of enemy die. When the latter condition happen, it is a mark end of battle, and PC party may continue their journey.

For each ensuing battles, each enemy CPU will have their own "threat meter" to decide which one should be picked. Take a look at chart below, it represents the threat level for each PC for the Revenant.

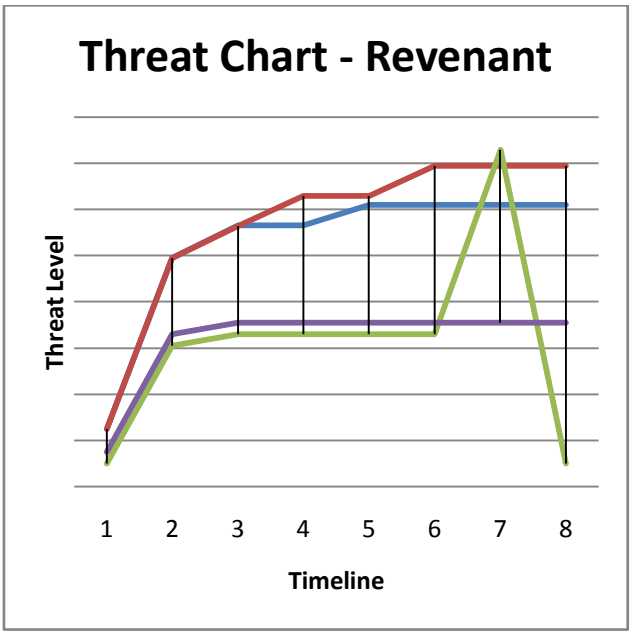

Fig. 2 Revenant perceived threat for each PC in battle

In the chart above, there are eight stages of battle. The explaination as follow:

1. Initial threat, enemy sighted.
2. Battle ensues, Alistair, Sten, Edgar, and Leliana come closer.
3. Both Edgar and Leliana start attacking the Revenant while Sten and Alistair come even closer to Revenant.
4. Alistair and Sten attack skeleton warrior.
5. Alistair switches target to Revenant and start attacking.
6. Sten start attacking Revenant too.
7. Edgar casts the Storm of the Century.
8. Edgar clears his threat by casting Mind Blast.

The result of each threat will be close to the nuber shown in the chart above. At one point, the PC who has the most threat level will be the target of Revenant. Rinse and repeat until the battle is over.

For the enemy CPU greedy, it is a bit weird for a group of enemy to target the PC which is the hardest to die. Consider this, if the most threatening PC is a warrior which always has the most HP, wear heavy armor that has a high defense rating against incoming attacks, and hardest to kill, why do they still target him?

There is one logical reason to explain that kind of behavior. Imagine that there is a giant sturdy castle wall, and the three archers . The enemy is a group of pikeman with a battering ram specifically designed to destroy the wall. When the sturdy wall collapse, those archer can do nothing. They are just waiting for their fate to be sealed. Then, if the sturdy wall is the Warriors and three archers are mages or rogues, then you can imagine what will

happen when the Warriors, as the trump card, down. It is most likely that the rest party will down too.

Greedy by the most threatening has a characteristics to face the hardest enemy first and save the weakest for the last. It can be shown as the chart below:

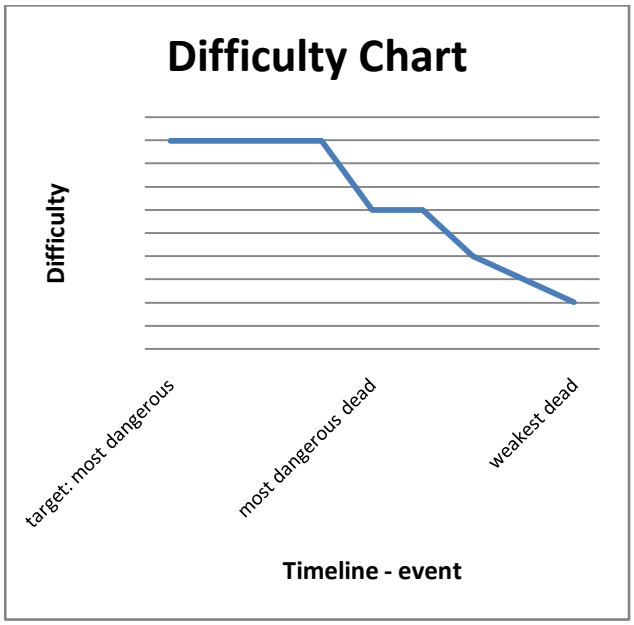

Fig. 3 Enemy Difficulty Chart

V. CPU GREEDY VS PLAYER GREEDY

Is it possible to defeat such kind of greedy AI by implementing our own greedy method? It is possible to counter that kind of greedy by using a different approach In the algorithm above, the enemy CPU greedy choose which one is the most threatening PC. This greedy has one big flaw. Most the time, this greedy will return target which is the strongest and hard to kill PC. This greedy, although good to ensure the first enemy to down is the main trouble-maker, it often failed to ensure victory because the trouble-maker is strong enough to survive and can launch counter attack.

As a fellow player, let's try the complete opposite of the enemy greedy. If the previous one was about kill the strongest enemy, in this greedy, we seek the weakest enemy in presence, and try our best to kill it as fast as possible. In this kind of approach, our target is to reduce the enemy number as fast as possible. It also known as reduce and conquer (RnC) strategy.

Why would player reduce the enemy number? This main objective is to reduce the total amount of damage which the party have to suffer. Suffering a long term damage is very serious danger, and this is what player must avoid at all cost.

Consider this, if a player devise a strategy which attack the boss first (in this case, Revenant), there are still some dangers from the minions. While attacking the Revenant, those minions constanly deal damage to each party. In a prolonged battle, there is still some high risks, such as will the party survive the minions attacks, or will the party even survive the whole Boss and minions attacks? Maybe no.

By reducing the enemy number, it will be easier to maintain party health while dealing damage to enemy. So, we will test our so called "Greedy by the Weakest" to anihilate the whole enemy party. Let's first define the greedy elements:

- Candidate set: each enemy in the battlefield.
- Solution set: the weakest enemy at the moment.
- Selection function: enemy which is easiest to kill, whether it has a relatively low HP or low defense rating.
- Feasibilty function: is the enemy still alive?
- Objective function: attack enemy which is the weakest.

In the test case, deploy the party setup and strategy below:

Party setup:

- Edgar
  Mage, offensive, AoE massive damage dealer.
  Equipment: Robes of Avernus
  Defense rate: 0
- Alistair
  Warrior, tanker, high defense, aggro puller.
  Equipment: Blood Dragon Armor set
  Defense rate: 27.31
- Leliana
  Rogue, archer, long range attacker.
  Equipment: Wade's Dragonskin Armor set
  Defense rate: 15.01
- Sten
  Warrior, tanker, high defense, high attack, stunner.
  Equipment: Legion of the Dead Armor set
  Defense rate: 31.89

Tactic slots:

- Edgar

Target	Action
Self: MP < 50%	Lesser Lyrium Potion
Enemy: lowest health	Winter's Grasp
Self: attacked	Mind Blast

- Alistair

Target	Action
Self: HP < 50%	Strongest Health Poultice
Enemy: lowest health	Attack
Enemy: rank - Boss	Assault

- Sten

Target	Action
Self: HP < 50%	Strongest Health Poultice
Enemy: any	Taunt
Enemy: lowest health	Attack

- Leliana

Target	Action
Self: HP < 50%	Lesser Health Poultice

Enemy: lowest health	Attack
Enemy: rank - Boss	Arrow of Slaying

The result is quite satisfying. The number of potion usage can be reduced from 6 to 8 vials to just 5 vials. Although the damage Edgar and Leliana suffer do not change obviously, the damage both Sten and Alistair have gone better. The difficulty faced by party against the enemy is shown below:

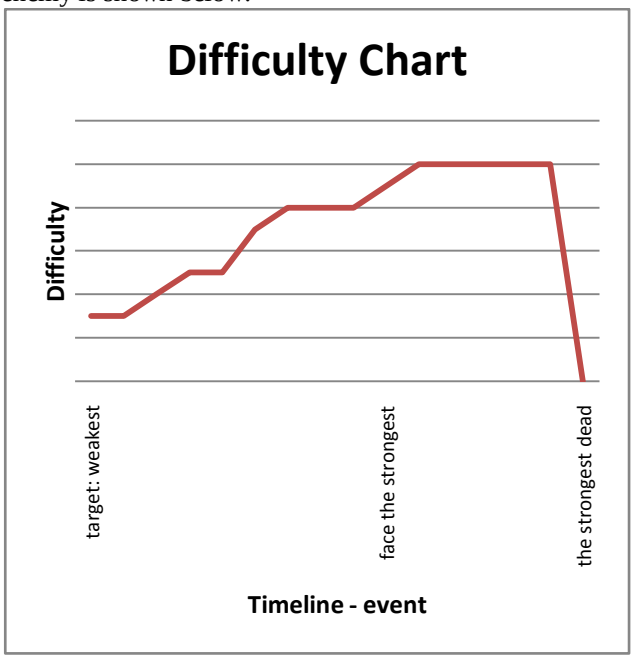

Fig. 4 Player Difficulty Chart

Most of the time, this strategy quite handy and give satisfying result. For some battles, a little adjustment in this greedy is essential, since this greedy is just too offensive. There are some battles which require patient and tactics rather than a simple hack-and-slash to weakest enemy.

However, in some instance, simply targetting the weakest enemy will not give you the expected result. For example, in the Final Battle against the Archdemon (rank: Elite Boss), Archdemon will constantly spawn lesser minions like darkspawn (rank: normal) as long as the battle ensues. If we still use greedy by the weakest, the battle will not over, since all our party do is killing the minions, while the Archdemon is unscathed.

VI. Conclusion

To sum up, greedy algorithm can be implemented almost in any game which requires decision making. The algorithm can be easy, but the formula to decide which is local optimum may be complex. Greedy implemented in Dragon Age CPU choose target which is the most threatening or the most dangerous PC in the battlefield. It may result some kind of faults, where the result may not be satisfying.

On the other hand, player using strategy greedy by the weakest can reduce the number of current enemy to deal with. By doing so, damage sustain in a prolonged battle may be relieved little by little. This strategy deal with the most problematic enemy last, in order to ensure that every minion die first. This is an ideal strategy to win most of the battles.

However, it is clear that the greedy algorithm does not always give the expected or the optimum result.

References

Dragon Age Wikia. (2010). *Threat*. Retrieved December 2011, from Dragon Age Wikia: http://dragonage.wikia.com/wiki/Threat

Muhammad, R. b. (n.d.). *Greedy Algorithms*. Retrieved December 2011, from Computer Science Kent: http://www.personal.kent.edu/~rmuhamma/Algorithms/MyAlgorithms/Greedy/greedyIntro.htm

Munir, Rinaldi. (2009). *Strategi Algoritma*. Bandung: Program Studi Teknik Informatika.

Wikipedia. (n.d.). *Greedy Algorithm*. Retrieved December 2011, from Wikipedia: http://en.wikipedia.org/wiki/Greedy_algorithm

Statement of Originality

In this statement of originality, I officially state that this paper is my original work, not an editted or translated from another paper, nor any form of plagiarism from other person works.

Bandung, 29 April 2010

Rido Ramadan - 13509049