International Journal on Advanced Science Engineering Information Technology

Vol.7 (2017) No. 5
ISSN: 2088-5334

Rule-based Procedural Generation of Item in Role-playing Game

Chin Kim On#, Ng Wai Foong#, Jason Teo#, Ag Asri Ag Ibrahim#, and Tan Tse Guan*

Faculty of Informatics and Computing, Universiti Malaysia Sabah, 88400 Kota Kinabalu, Sabah, Malaysia.
E-mail: [1] kimonchin@ums.edu.my, [2] ngwaifoong92@gmail.com, [3] jtwteo@ums.edu.my, [4] aaai500@gmail.com

**Faculty of Creative Technology & Heritage, Universiti Malaysia Kelantan,*
Kelantan, Malaysia.
E-mail: tan.tg@umk.edu.my

***Abstract*— This paper demonstrates the significance of rule-based procedural generation of items in Role-Playing Game (RPG). The main aims of this project are to: implement rule-based randomized algorithm and totally randomized algorithm in generating item procedurally in RPG, and then compare the advantageous of rule-based randomized algorithm against totally randomized algorithm in term of item generating mechanism. Experimental results demonstrate success with all aims: rule-based randomized algorithm is proven to be a better game changing factor in procedural generation of item as it can control the prolific generation of strong items in the early stage of the game. This helps to balance the game and prevents any snow-balling effect as the game progresses.**

***Keywords*— role-playing game; procedural generation; rule-based randomized algorithm; totally randomized algorithm.**

I. INTRODUCTION

Procedural generation is the concept of creating something dynamically rather than manually. It is mostly used in creating the process of making content in systems. In the world of games, it can be applied in generating the content such as textures, quests, items and others. Procedural generation allows the development of complex gameplay and creation of game's world within short amount of time [1]. Other advantages of using procedural generation in game includes smaller file size, more content can be generated manually and a less predictable gameplay due to the random-ness factor which improve the gameplay value. There are incredibly increase of gaming AI research since last decade. Researchers pointed the challenges, tools/simulator, possible solutions, and newly proposed algorithms in [2-7]. However, procedural generation of content has been gaining momentum and looks promising as the future style in games [8]. It even has its own tag in the popular game platform Steam for the user to search for games that contains procedural generation. This is simply because it provides a different feel in each gameplay of the same game [9].

A number of randomized algorithms can use in the item generation mechanism of a RPG. The most common algorithms being used are Las Vegas algorithm [10], Monte Carlo algorithm [11], totally randomized algorithm and modified rule-based algorithm [12]. For the past years, procedural generation has been frequently used by games developers for content generation. The most frequent usage of procedural generation is on terrain generation, item generation and monster generation. Examples of RPG that rely heavily on procedural generation are Diablo [13-14], Left4Dead [15], Minecraft [16-17] and No Man;s Sky [18].

In this research, the main objectives are to create two randomized algorithm for generation of item in the game prototype and compare both of them in terms of the number of attributes generated and value of the attributes generated. The first randomized algorithm is rules free while the second randomized algorithm involved certain complex rules. Rules serve as a parameter that can control and skew the outcome of the probability function. It works by restricting the domain to a certain area based on pre-defined rules. The new domain is usually smaller than the original domain. Additional rules can be applied together to form a complex rule. A practical usage of this algorithm includes field of science that require controlled uncertainty such as neural networks, search engine, data mining and mapping of starts in astronomy [12]. We use a game prototype that we designed in the form of RPG that has been created previously to conduct this experiment. RPG is chosen simply because it is the most common genre in representing generation of item. This study showed that rule-based randomized algorithm has the ability to control the prolific generation of strong items which is essential for game designers to balance the gameplay.

II. Material and Methods

This section consists of item parameters, detail of items parameters involved and later used in the game, summary of all rules proposed for the game, few screenshots of scenarios involved in the game, and the proposed methods.

A. Item Parameters

Item parameters are attributes that are present in both unit and item. It affects the value of the respective attribute upon equipped by the player. These parameters are used for comparison between rule-based randomized algorithm and totally randomized algorithm in this project. The type of parameters used in designing RPG game could be different based on the game's story board. In science fiction game, it does not involve magic attribute rather focus more on strength and speed. In fact, no one ever suggested to kill zombie with magic as peoples believe zombie does not exist in the magic world.

In this research work, we planned to design a game that involves spell casting capability. Hence, magic attribute is one of the important parameter. Item generated could have certain combination of attributes but any item generated cannot have all attributes. Normally, the attributes involved are hitpoints, strength, speed, magic, defense, critical point, evasion, and quality find. Table 1 shows the detail explanation of item parameters involved in the project.

TABLE I
Details of Item Parameters in The Game

Attribute	Description
Hitpoint	The amount of health of the unit. If unit health falls to 0, the unit is considered dead.
Strength	The amount of attack power a unit has. It affects the damage output by the unit in normal attack.
Speed	The amount of speed a unit has. It affects base attack time of the unit.
Magic	The amount of spell power a unit has. It affects the damage output by the unit in a spell attack.
Defense	The amount of defense a unit has. It reduces damage dealt by opponents in a normal attack.
Critical	The chance to deliver extra damage based on multiplier.
Evasion	The chance to dodge a normal attack from opponents.
Quality Find	The chance to roll better quality item from monsters (diminishing returns).

B. Item Generation Rules

Rules are essential in building the structure of the game which provides meaningful actions to the players. Besides, rules can be designed to bring both positive and negative experiences to the players in a game [19]. We reviewed some of the rules used in designing RPG games and then adopted, redesigned and proposed some rules that can fit to our game. Table 2 shows the summary of the rules used in the designed game. There are two formulas involved in the Rules 7 and 8 as shown in Table 2. The formulas (1) and (2) are as below.

*Modified quality chance = base quality chance + (base quality chance * effective quality find / 100)* (1)

*Effective quality find = (quality find * factor) / (quality find + factor)* (2)

TABLE II
Summary of Rules Used in The Game

Rule	Description
1	There is no requirement for item. Unit can equip any item regardless of level or attributes.
2	The chance to find an item is based on the strongest monster in battle. The details of the item generated chance is as follow: • Junior (80%) • Senior (90%) • Boss (100%).
3	Item level is randomly generated using the treasure class of a monster as the limit. There will be more than one item generated in a treasure rather than single item in most of the cases. The items generated in treasure class will be more valuable.
4	If the item level is 0, no item will be generated from the monster.
5	There will be chance for penalty of item level based on the player level. Higher level of player will have a lower chance to trigger the penalty. The penalty of item level is as follow: • If item level is greater or equal to 8, item level is reduced by 3. • If item level is greater or equal to 5 and less than 8, item level is reduced by 2. • If item level is greater or equal to 2 and less than 5, item level is reduced by 1.
6	The base quality chance is as follow: • Magic (95%) • Rare (25%) • Unique (5%).
7	Quality chance is affected by the quality of the player. The chance to roll a type of quality item find is calculated using formula (1). Formula (1) is at the bottom of this table.
8	Effective quality find is the diminishing returns formula that applied for rare and unique quality. The value is calculated using formula (2).
9	The factor input for effective quality find is as follow: Rare (600), Unique (250).
10	The number of attributes for item is based on the item quality as shown below: • Magic (1 – 2 attributes) • Rare (2 – 4 attributes) • Unique (3 – 8 attributes).
11	The attributes that will be generated for item is as follow: Hitpoint, Strength, Speed, Magic, Defense, Evasion, Critical and Quality Find.
12	The value of the attributes is randomly generated using limit that is defined by the item level respectively. The minimum (item level 1) and the maximum (item level 10) limit is as follow • Hit point (50 – 500) • Strength (5 – 50) • Speed (5 – 15) • Magic (3 – 30) • Defense (3 – 20) • Evasion (3 – 12) • Critical (6 – 15) • Quality Find (5 – 50).

C. Scenario of the Game

The game prototype set in an open world 2-dimensioanl RPG where player will combat with monsters in an area. Fig. 1 shows the graphical player interface of the game prototype.

Player can attacks, casts spells or buffs and attempts to run away from the battle while in combat (Fig. 2). Upon winning the fight, player will be awarded with gold, experience and item (if it is generated). Fig. 3 shows the battle result interface where player can decide to keep the item or skip it.

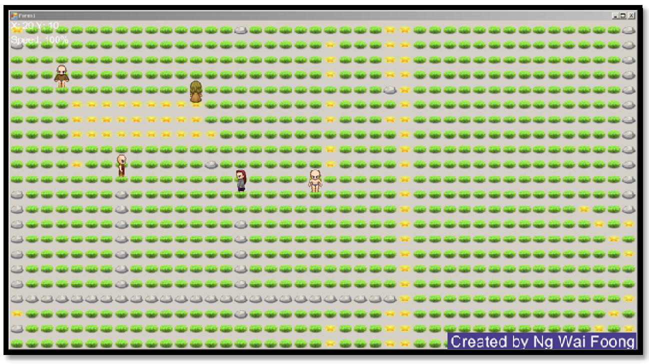
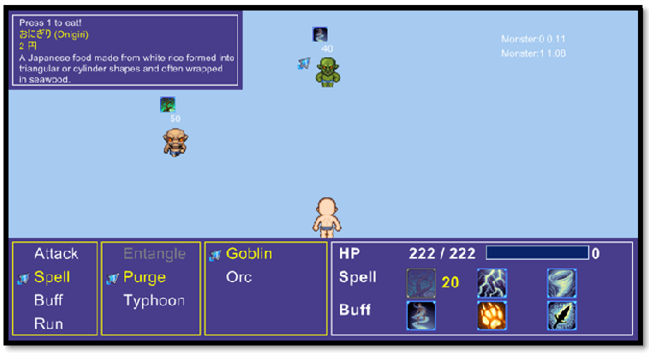


Fig. 1 Graphical user interface of the game prototype

Initially, the generated items can be divided into four categories depends on rules shown in Table 2:

1. Magic
2. Rare
3. Unique
4. Normal (neither magic, rare nor unique)

The generated items may be different in term of quality as well and this depends on the combination of rules shown in Table 1, Table 2, and the formulas (1) and (2) used. The generated items are normally categorised as weak if it does not fit the player level and it is a normal item. On the other hand, it is considered as strong if it is having high attributes with either magic, rare or unique items.

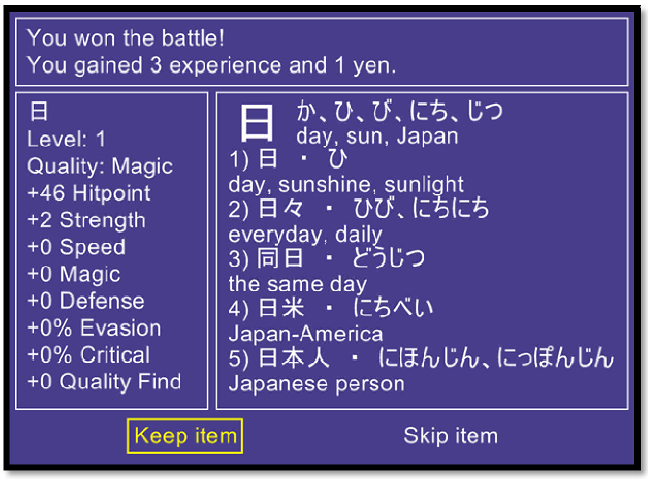


Press 1 to eat!
おにぎり (Onigiri)
2 円
A Japanese food made from white rice formed into triangular or cylinder shapes and often wrapped in seawood.

Fig. 2 Combat system in the game prototype

D. Methods Used

The testing approach used in this project is based on simulations instead of real players. This is simply because this method is more suitable compared to real players. The data obtained through the simulations is the same as when using real players. Besides, the data obtained using simulations is to be accurate while data obtained from players might be different from the actual output data. In addition, the number of simulations needed for comparison is impractical to be obtained from players. The simulation is performed using just one map level as the constant. In the map level, there are 3 junior monsters with treasure class level 1, 2 and 5. The range of levels in the prototype is 1 to 100. However, the initial input level for simulation is 10 instead. This is mainly because there is not much difference in the result between level 1 and 10. Level 10 indicates the early stage of the game. Players are expected to obtain mostly weak items from monster. On the other hand, level 50 refers to the mid game progression in the game. The player will receive a mixture of weak items from monster. Furthermore, level 90 is the peak of the game progression in which player will be expecting weak, medium and strong items generated from monsters. In other words, there will be six experimental setups involved using level 10, 50 and 90 that are paired with rule-based randomized algorithm and totally randomized algorithm (Table 3).



You won the battle!
You gained 3 experience and 1 yen.

日
Level: 1
Quality: Magic
+46 Hitpoint
+2 Strength
+0 Speed
+0 Magic
+0 Defense
+0% Evasion
+0% Critical
+0 Quality Find

日 か、ひ、び、にち、じつ
day, sun, Japan
1) 日 ・ ひ
day, sunshine, sunlight
2) 日々 ・ ひび、にちにち
everyday, daily
3) 同日 ・ どうじつ
the same day
4) 日米 ・ にちべい
Japan-America
5) 日本人 ・ にほんじん、にっぽんじん
Japanese person

Keep item Skip item

Fig. 3 Battle result interface of the game prototype

TABLE III
EXPERIMENTAL SETUPS IN THE GAME

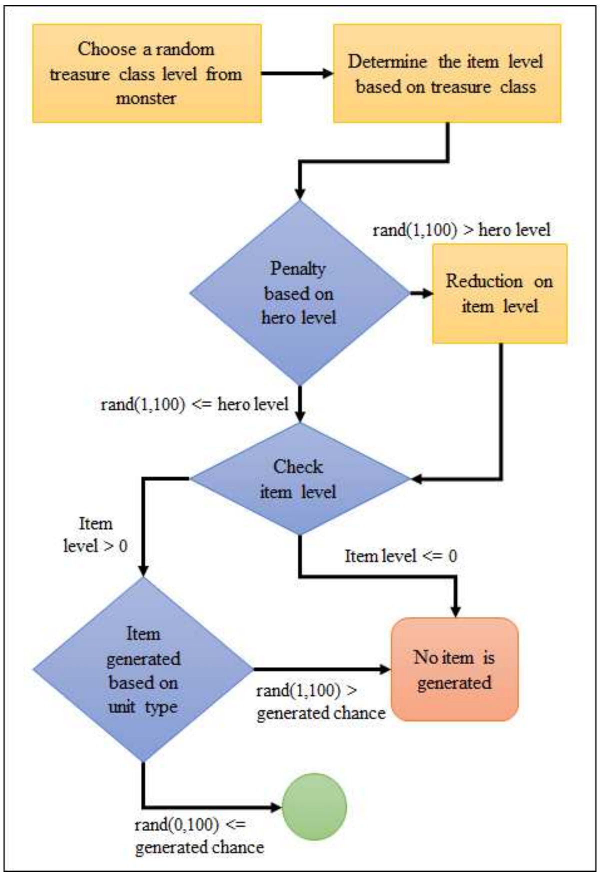
Input level	Algorithm
10	Rule-based randomized algorithm
10	Totally randomized algorithm
50	Rule-based randomized algorithm
50	Totally randomized algorithm
90	Rule-based randomized algorithm
90	Totally randomized algorithm

Generally, the item generation mechanism used in the game prototype is as follow:

- A random treasure will be generated based on the class level of the monster. More valuable items will be generated by higher level monster.
- Then, the item level will be determined based on the treasure class.
- Penalty will be applied based on player's level. The item's level will be reduced if the item generated is in the higher hierarchy than player's level.
- No item will then be generated if the randomized generate chance is smaller or the item generated is less than level 0.
- Next process determines the type of the item generated. Item generated could be either in the form of:
-

  - Unique item or
  - Rare item or
  - Magic item
- No item will be generated in the game if the randomized mechanism is neither unique, nor rare, nor magic.
- Last process involves determining the number of attributes involve and their values.

Both rule-based randomized algorithm and totally randomized algorithm are having same mechanism except that there are more steps involved in rule-based randomized algorithm due to the constraints used. Figs. 4 and 5 shows the flowchart of the item generated mechanism using rule-based randomized algorithm while Fig. 6 represents totally randomized algorithm.



Choose a random treasure class level from monster
Determine the item level based on treasure class
rand(1,100) > hero level
Penalty based on hero level
Reduction on item level
rand(1,100) <= hero level
Check item level
Item level > 0
Item level <= 0
Item generated based on unit type
No item is generated
rand(1,100) > generated chance
rand(0,100) <= generated chance

Fig. 4 Item generated mechanism using rule-based randomized algorithm

The totally randomized algorithm consists of much simple item generated mechanism and it is as follow:

- Firstly, the item level will be randomized.
- No item will be generated if the randomized item's level is less than 0.
- Then, the probability of item generated will be randomized.
- No item will be generated if the probability is less than 0.
- The next process determines the item quality.
- It determines the number of attributes for the item generated.
- Attributes' values will be determined in the last stage.

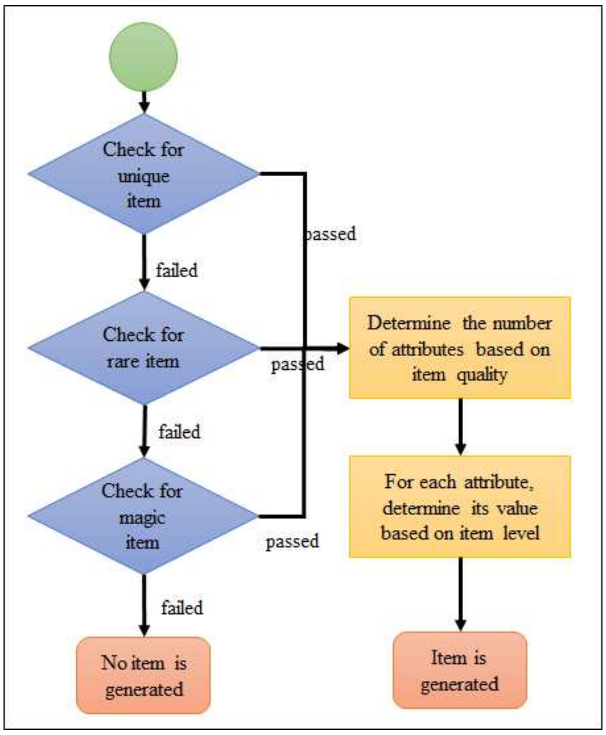
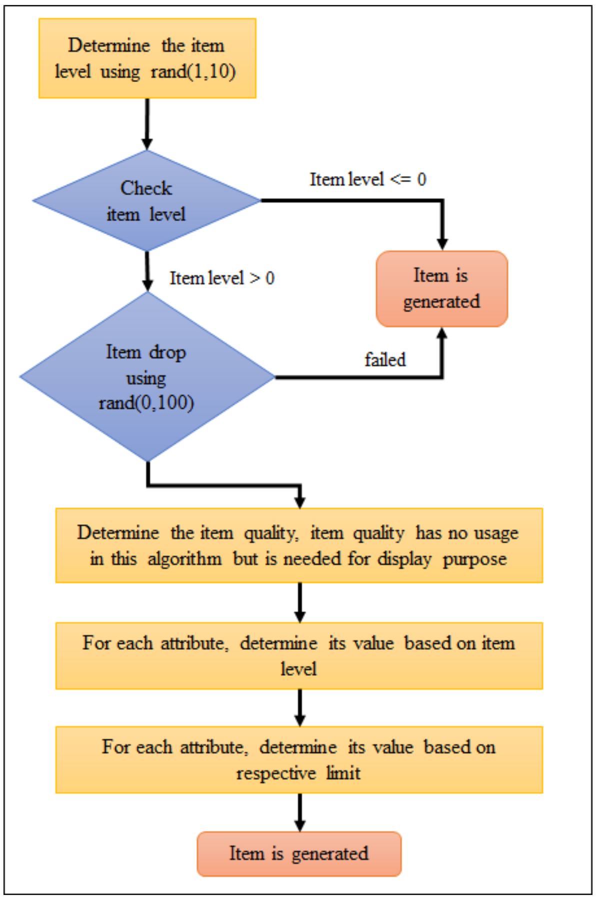


Fig. 5 Item generated mechanism using rule-based randomized algorithm (continue)



Determine the item level using rand(1,10)
Check item level
Item level <= 0
Item level > 0
Item is generated
failed
Item drop using rand(0,100)
Determine the item quality, item quality has no usage in this algorithm but is needed for display purpose
For each attribute, determine its value based on item level
For each attribute, determine its value based on respective limit
Item is generated

Fig. 6 Item generated mechanism using totally randomized algorithm

III. Results and Discussions

Table 4 shows the number of item generated after all simulations have been completed. Each number is used as the base total item generated in the calculations of average number of attributes per item generated and average value of respective attribute per item generated.

TABLE IV
NUMBER OF ITEMS GENERATED

Algorithm (Level)	Test 1	Test 2	Test 3
Random (10)	437	429	429
Rule (10)	534	521	534
Random (50)	453	423	444
Rule (50)	511	525	511
Random (90)	419	437	417
Rule (90)	525	509	524

Average number of attributes is calculated using the following formula:

Total number of attributes / Total item generated (3)

The number of attributes range from one to eight if an item is generated. Higher number of attributes indicates a stronger the item.

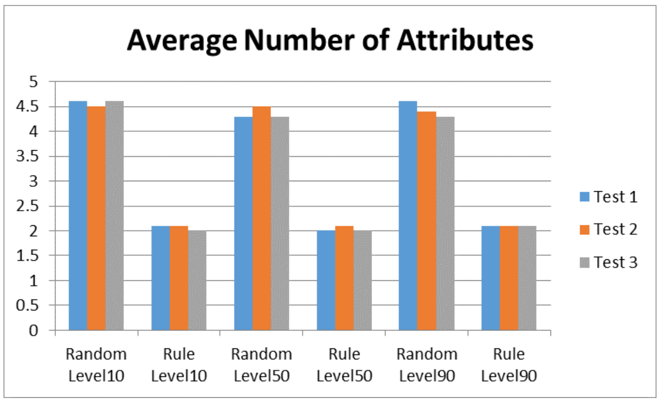


Fig. 7 Average number of attributes for all algorithm

From Fig. 7 above, totally randomized algorithm has about 4.5 attributes per item generated for all three levels. This value surpassed the average number of attributes per item generated for rule-based randomized algorithm by a huge margin. It is more than double the value generated for rule-based randomized algorithm which is about 2 attributes per item for all three levels. Since more attributes is being generated per item, totally randomized algorithm is expected to have better average for the value of attributes as compared to rule-based randomized algorithm. This indicates that rule-based randomized algorithm is capable to control and limit of strong item being generated by the game. The average value of attribute is used as one of the comparison method in this research. The average value of attribute serves as an indicator in determining whether an item is strong, medium or weak. Strong items are desired to empower and enhance the combat capabilities of the character. However, strong items should not be easily obtained by the player in the early stage of the game. Attributes involves in this research are hitpoint, strength, speed, magic, defense, critical, evasion and quality find. In this paper, only one attribute is shown in graph as all attributes displayed similar pattern and result. The attribute chosen is hitpoint in which the average value can be calculated using the following formula:

Total number of hitpoint / Total item generated (4)

Higher value of hitpoint provides more sustainability to the character in combat.

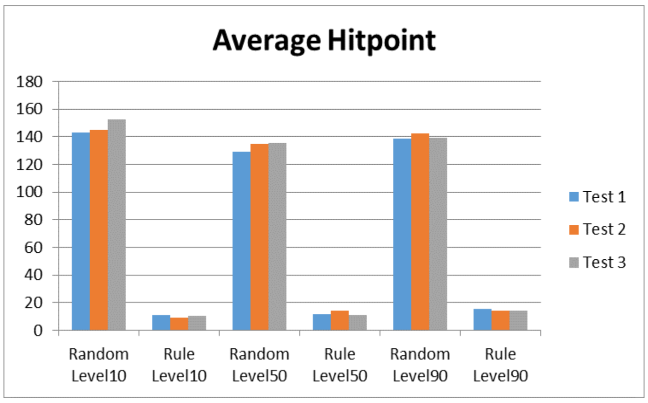


Fig. 8 Average value of hitpoint for all algorithms

Based on Fig. 8, it can be seen that the average hitpoint per item using totally randomized algorithm immensely outperformed the average hitpoint per item using rule-based randomized algorithm. It indicates that the player have better and higher sustainability in combat even at early game when using the totally randomized algorithm. This will result in lesser difficulties and obstacles which consequently affects the gameplay. Meanwhile, rule-based randomized algorithm shows a more reasonable value for the hitpoint generated. It provides slightly increase in sustainability of the character without affecting much of the gameplay. Players will still face difficulties in the combat. Besides, the chart also shows slightly increase in the average hitpoint for rule-based randomized algorithm as the level of character increased. This means that players will be able to get better value in high level compared to low level. The other comparison method used in this research is the max value of respective attribute. Max value of attribute can be used to determine the highest value of attribute that is generated in item using different algorithm. In actual gameplay, low level character should not be able to obtain the highest value of attribute. The max value of attribute should be on increasing trend as the level of character increases. Only one attribute is shown in graph for this paper as the pattern and result of all attributes are almost similar. The attribute chosen is quality find in which its value range from 1 to 50. Higher value of quality find indicates better chance of getting higher quality items after winning a combat.

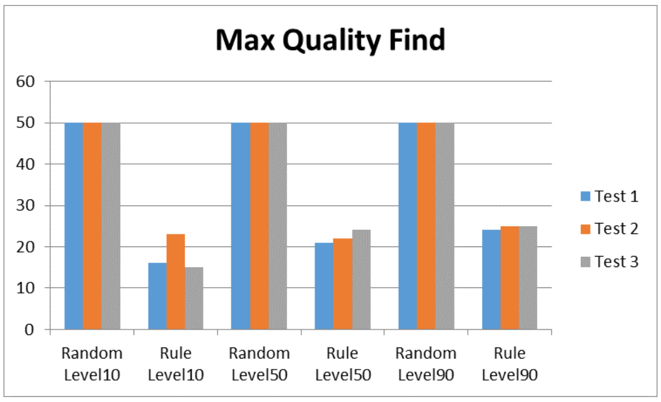


Fig. 9 Max value of quality find for all algorithms

From Fig. 9, the maximum value of quality find generated by totally randomized algorithm tends to be the maximum value of quality find which is 50. If a low level character able to get such high value item in the early stage of the game, the tune of the game will be changed in favour of the player. This is simply because this attribute has the snowballing effect as the game progress. It enables the player to get higher quality items which in turn boost the performance of the character in combat. On the other hand, rule-based randomized algorithm provides more reasonable maximum value which is about a third of the maximum value available. This value increases as the level of the character increases as shown from level 10 to level 50 and finally level 90 which recorded about half of the maximum value available.

IV. Conclusion

From all the graphs obtained in the research, totally randomized algorithm produced strong item with high number of attributes regardless of the level of the character. Low level character is able to obtain high-end gear at the early stage of the game. It is possible to obtain the maximum value of each respective attributes even at the beginning level. In addition, playing with low level character or high level character will not affect how the item is being generated in the game. This tremendously affects the game play as player will be able to pass through the challenges and obstacles easily once they obtain an item or two. On the other hand, rule-based randomized algorithm produces much weaker item with mostly low number of attributes. The level of the character has an effect on the value of attributes being generated. Low level character tend to get lower value of attribute while high level character got mixture of low value of attribute and high level of attributes. Unlike totally randomized algorithm, player is not able to obtain the maximum value of each respective attributes at the beginning level. However, player can expect to obtain higher maximum value of attribute as the level of their character increases. In other words, playing with low level character will obtain low-gear item while playing with high level character will obtain both low-gear and high-gear item. This helps to balance and improve the game play value of the game. Furthermore, the rules can be modified to further improve and balance the item generated. This allows game developer to easily tweak and enhance their game according to the needs of the player.

In a nutshell, having rules and constraints is essential in the item generated mechanism of role-playing game as it improves the game play value by restricting and limiting the player from obtaining high-gear item at the early stage of the game. With specific rules and constraints, better item will be generated to the players as the game progress which involves the increase in level. As a result, game developer will be able to keep the game in balance at the early stage of the game. Besides that, it can also prevent the player from losing interest as well as motivating the player to continue playing to obtain better gear as the game progress. Hence, rule-based randomized algorithm is proven to be a good and effective algorithm to procedurally generate items in role-playing games as compared to totally randomized algorithm. The benefits of rule-based randomized algorithm can be applied in other platforms as well. Other type of genres such as First Person Shooter can implement rule-based randomized algorithm for generation of guns and ammo. The game designers only need to change the rules according to the nature of their game.

Acknowledgment

This research work is partly supported and funded by Universiti Malaysia Sabah, Artificial Intelligence Research Unit (AiRU) and Center of Excellent in Semantic Agents (COESA).

References

[1] T. Hatfield. (2013) Rise Of The Roguelikes: A Genre Evolves. [Online]. Available: http://pc.gamespy.com/pc/ftl-faster-than-light/1227287p1.html

[2] J. L. Shi, T. G. Tan, T. Jason, K. O. Chin, R. Alfred, and P. Anthony, “Evolving Controllers For Simulated Car Racing Using Differential Evolution,” in *Asia-Pacific Journal of Information Technology and Multimedia*, 2013, vol. 2(1), pp. 57-68.

[3] K.B. Tan, J. Teo, K.O. Chin, and P. Anthony, *An Evolutionary Multi-Objective Optimization Approach to Computer Go Controller Synthesis*, Springer Berlin Heidelberg, pp. 801-806, 2012.

[4] K.T. Chang, K.O. Chin, J. Teo, and B.L. Chua, "Game AI generation using evolutionary multi-objective optimization," in Sustainable Utilization and Development in Engineering and Technology (STUDENT), 2011, pp. 101-106.

[5] K.T. Chang, K.O. Chin, J. Teo, and J. Mountstephens, "Game AI generation using evolutionary multi-objective optimization," in IEEE Congress on Evolutionary Computation, pp. 1-8, 2012.

[6] T. G. Tan, Y. N. Yong, K. O. Chin, J. Teo, and R. Alfred, *Automated Evaluation for AI Controllers in Tower Defense Game Using Genetic Algorithm*, in Soft Computing Applications and Intelligent Systems Heidelberg: Springer Berlin, 2013, pp. 135-146.

[7] T.G. Tan, J. Teo, and K.O. Chin, "Single-versus Multiobjective Optimization for Evolution of Neural Controllers in Ms. Pac-Man," International Journal of Computer Games Technology, vol. 2(2), pp. 53-61, 2013.

[8] S. Young. (2009) The Future is Procedural. [Online]. Available: http://www.escapistmagazine.com/articles/view/video-games/columns/experienced-points/6418-The-Future-is-Procedural

[9] M. Ericksen. (2010) Procedural Generation as Future of Game Design. [Online]. Available: http://playaslife.com/2010/05/03/procedural-generation/

[10] L. Babai, “Monte-Carlo algorithms in graph isomorphism testing,” Université de Montréal, D.M.S, pp. 79-10, 1979.

[11] P. Jonathan. (2002) Monte Carlo Methods. [Online]. Available: http://www.cs.otago.ac.nz/cosc453/student_tutorials/monte_carlo.pdf

[12] R. V. Samuel, “Item Generation Using Rule-Based Randomization Algorithms in Role Playing Games,” University of Abertay Dundee, Institute of Arts, Media and Computer Games, pp.31, 2011.

[13] (2016) Item Generation Tutorial. (n.d.). [Online]. Available: http://diablo2.diablowiki.net/Item_Generation_Tutorial

[14] Map Generation Analysis: The Weeping Hollow. (2012) [Online]. Available: http://www.diablofans.com/news/47228-map-generation-analysis-the-weeping-hollow-groups

[15] Left 4 Dead Hands-on Preview. (2008) [Online]. Available: http://www.shacknews.com/article/50799/left-4-dead-hands-on

[16] A. Meer. (2010) BiomeShock: The New Minecraft Worlds, Rock, Paper, Shotgun. [Online]. Available: http://www.rockpapershotgun.com/2010/10/27/biomeshock-the-new-minecraft-worlds/

[17] J. Bergensten. (2011) A Short Demystification of the Map Seed. [Online]. Available: https://mojang.com/2011/02/a-short-demystification-of-the-map-seed/

[18] R. Hiranand. (2015) 18 quintillion planets: The video game that imagines an entire galaxy. [Online]. Available: http://edition.cnn.com/2015/06/18/tech/no-mans-sky-sean-murray/

[19] J. Whitehead. “Definitions of Games and Play, Magic Circle, Rules as Limitations and Affordances,” Creative Commons Attribution 2.5 ed. Santa Cruz: University of California, 2007.