Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 59 (2015) 435 – 444

Procedia Computer Science

International Conference on Computer Science and Computational Intelligence (ICCSCI 2015)

Dynamic Difficulty Adjustment in Tower Defence

Rhio Sutoyo[a], Davies Winata[a], Katherine Oliviani[a], Dedy Martadinata Supriyadi[a]*

[a]*Bina Nusantara University, Jl. K. H. Syahdan No. 9, Kemanggisan / Palmerah, Indonesia*

Abstract

When we play tower defence game, generally we repeat the same stages several times with the same enemies. Moreover, when the players play a stage that is ridiculously hard or way too easy, they would probably quit the game because it is moderately frustrating or boring. The purpose of this research is to create a game that can adapt to the players' ability so the difficulty of the game becomes dynamic. In other words, the game will have different difficulties of levels according to the players' ability. High difficulty levels will be set if the players use good strategy and low difficulty levels will be set if the players use bad strategy. In this work, we determine the difficulties based on players' lives, enemies' health, and passive skills (skill points) that are chosen by the player. With three of these factors, players will have varies experience of playing tower defence because different combination will give different results to the system and difficulties of the games will be different for each gameplay. The result of this research is a dynamic difficulty tower defence game, dynamic difficulty adjustment (DDA) document, and gameplay outputs for best, average, and worst strategy cases.

Peer-review under responsibility of organizing committee of the International Conference on Computer Science and Computational Intelligence (ICCSCI 2015)

Keywords: dynamic game balancing, tower defence, dynamic difficulty adjustment, computational intelligence

1. Introduction

Technology nowadays has bring games to a new era, there are several game consoles and several devices such as smartphones, tablets, and other portable devices that make games becoming part of our lives. Generally, games are

* Corresponding author. Tel.: +62-838-936-33550; fax: +62-530-0244
E-mail address: rsutoyo@binus.edu

Peer-review under responsibility of organizing committee of the International Conference on Computer Science and Computational Intelligence (ICCSCI 2015)
doi:10.1016/j.procs.2015.07.563

created to be played and purposely for fun. However, games could also become a media to improve creativity and enhance the education system.

There are many elements incorporated into games to make them interesting and appealing. For example, game storyline is important to make the players enjoy the story of the game when they are playing. Interface or aesthetics of the games is crucial to make the games easy to be played and eye-catching for the players. Moreover, challenges in the games have to make the players feels engaged with the games. In this work, we will focus on the challenge element of the games.

Games will not be challenging if they are too easy or difficult to play. With that kind of game experience, the players tend to quit the game after tried several times playing the games. In order to solve this problem, most game designers are allowing the players to choose the difficulties of the games, i.e., easy, medium, hard, at the beginning of the games. Not only it solve the existing problems, it also give players the options to play the game with their level of confidence or game experience. Nevertheless, this solution produces another trouble which is static difficulties[1]. For instance, when the player chose easy difficulties, the games will have low difficulties for the entire gameplay. The problem with this is, we as the players, have the ability to master the game after playing several times. Thus, after we are able to master the game, easy difficulties now becomes too boring because the challenges are not hard enough. On the other hand, for several people, easy difficulties might be still hard for them. Thus, resulting the same trouble with the prior problem. This case also happens for different difficulties, such as: medium and hard.

In general, the weakness of tower defence games is the repetition system. In the tower defence games, the players play same levels with the same patterns (e.g., monsters, type of monsters, bosses, etc.), and with static difficulties. For example, there will be always 30 mud golems with a specific attributes in the level or wave 10. In general, after the players playing the games several times, they will know what to do in order to beat that mud golem's level.

To solve those problems, a game experience developed by computational intelligence[2] and dynamic difficulty adjustment (DDA)[3,4] could help the game designer to bring more interesting, playable, and challenging game experiences for the players at the same time. With DDA, the system is able to calculate players' strategy. Hence, if the users use bad strategy, the system can lower the difficulties of the game and make the game more playable for the players. This technique is great to help new players to learn the basic without having to listen or follow to a static tutorial steps. Moreover, when the players use good strategy, the systems will increase the difficulties of the game and make the game more challenging for the advanced or expert players[9]. The system is also able to add extra attributes for the enemies to maintain the balance of the game. All of those adjustment are done in real-time or game-time (i.e. not at the beginning of the game). Furthermore, DDA could also be used to configure handicap systems in multiplayer games. This method is called Multiplayer Dynamic Difficulty Adjustment (mDDA)[10].

In this paper, the calculation of the DDA system is based on players' lives, enemies' health, and passive skills (skill points). We chose these attributes because these components are general in tower defence games[2]. Thus, our game design could be easily improved or used by other similar tower defence games (e.g. Element TD, Kingdom Rush, etc.). After the analyzing process has been done, the calculation result is used to adjust the enemies' attributes to make the game more appealing for novice and expert players. This game will have single player mode and contains 50 levels.

2. Related Works

2.1. Fundamentals of Game Design[1]

With so many other game design concepts, the conventional notion of balance defies formalization. A balanced game is fair to the players where it is not too hard nor too easy. In serious or strategy games, the most important factor in determining his/her success is the skill of the players. There are several different game features combined to produce these qualities and game balancing refers to a collection of design and tuning processes that creates those "fair" qualities. The game designer is the person who are in responsible to create those qualities in a game that still under development.

2.2. Adaptive Game AI with Dynamic Scripting[5]

Adaptive game AI is game AI that has the ability to adapt successfully to changing circumstances. Game AI is an essential element of game-play, which has become an important selling point of games[6,7]. In contrast, for game developers the term 'game AI' is used in a broader sense to encompass techniques such as pathfinding, animation systems, level geometry, collision physics, vehicle dynamics, and even the generation of random numbers[8]. The computational requirements are necessities: if adaptive game AI does not meet the computational requirements, it is useless in practice. The functional requirements are not so much necessities, as strong preferences by game developers. Failure of adaptive game AI to meet the functional requirements means that many game developers will be unwilling to include the technique in their games, even if the technique yields good results and meets all computational requirements.

2.3. AI for Dynamic Difficulty Adjustment in Games[3]

Hunicke and Chapman stated the game is designed to provide a better experience for users. Games are interactive, which means the player can interact with the environment created from the game. The dynamic difficulty adjustment (DDA) is a balancing game models based on Computational Intelligence. Dynamic adjustment difficulty will replace balancing method manually by the developers or game designers. With this technique, the games are expected to adjust the difficulties level automatically according to the assessment of a variety patterns to players during the game being played.

2.4. Computational Intelligence and Tower Defence Games[2]

Tower defence (TD) games are a genre of strategy games focusing on resource allocation and unit (tower) placement. This genre of games have proved to be a challenging, addictive, and fun way to fill our leisure time. The simplicity of the gameplay and the challenging tactics combined with the wide variety of games available in this genre has created a large population of casual TD gamers. The simplicity of the game makes the TD games an appropriate testing object for Computational Intelligence (CI) research. TD games provide a large user-base with a simple game mechanism, and a range of interesting potential research implementations. Furthermore, the TD genre is easy enough to implement for research purposes, while in the same time still providing enough of a challenge for user interest and complexity for exploration purposes. Many TD games have tactical and strategic depth (e.g. Kingdom Rush, Element TD, Plants vs Zombies, etc.). Thus, there are real challenges to be solved and the results can be solved by the AI research in general. With all of those characteristics, there are many possibilities to improve the gameplay of TD games using CI methods.

3. Dynamic Difficulty Adjustment (DDA) System Overview

As we have stated before, dynamic difficulty adjustment (DDA) technique is a process that is automatically change the parameters or behaviors of objects in the game at real-time. In order to create the DDA, we pick three conditions that will be analyzed by the system to determine the players' skill. Those conditions are:

1. Players' Lives
   In most of tower defence games, the lives system is used as the condition for winning the game. Lives of the players will decrease if enemies reach certain locations. Furthermore, the game will be over when the lives of the players become zero. In this work, we apply that live system in our game design. If the lives is decreased, the system will lower the difficulties and vice versa.

2. Enemies' Health
   In our game design, the tower defence game will have 50 levels or waves. Each of the level contains several enemies with different attributes and specialties. Moreover, there will special enemies, which is boss levels, for every five levels (i.e. 5, 10, 15, 20, 25, 30, 35, 40, 45, and 50). In our game design, if the players able to kill the bosses, they will get +5 for their lives. All of enemies have health. If their health reach zero, they will dead. The towers have to kill the enemies before they reach certain location (finish line). If the towers cannot kill them, the players' lives will be decreased. In our DDA system, the system will check how much health left of the enemies (i.e. total of all enemies) that is able to reach the finish line. This process is done in the end of the level. If the enemies' health is more than 50%, the system will predicts that the players are not having enough tower or they place the towers in bad positions or use bad combinations of towers. With this result, the system will decrease the difficulties of the game.

3. Passive Skills (Skill Points)
   Besides towers, players also able to spent skill points for passive skills. These passive skills are used to increase the power of towers. We categorize skill into three things: offensive, defensive, and support. If the player pick offensive skills, it will affect the enemy's status. Defensive skills will affect the rate of enemy spawned. Support skills will affect the gold rate in the game.

3.1. Variables and Multipliers

To create the dynamic difficulty adjustment for our tower defence game, we use multipliers that affect the objects in our games. These multipliers are changed in the end of every level where the change points depend on the performance gameplay of the players. For instance, if the players use good strategy and did not lost any lives in the certain levels, the multiplier points will be increased and the next levels will be harder.

There are four type of multipliers in our game, which are: status multiplier, spawn multiplier, gold multiplier, and difficulties point global. These multipliers will affect the number on the variables. The variables are status point, spawn point, and gold point. The status point affect the enemies' power, the spawn point affect the number of enemies that will be spawned at that current level, and the gold point affect the number of gold received in every cleared level. Furthermore, the difficulties point global affect all of those variables (i.e. status, spawn, and gold). In table 1 explain the initial numbers for those multipliers and variables:

Table 1. Initial Number of Variables and Multipliers

Variable and Multiplier Names	Initial Points
Difficulties Point Global	10
Status Point	1
Status Multiplier	1
Gold Point	1
Gold Multiplier	1
Spawn Point	1
Spawn Multiplier	1

3.2 Increment of Multipliers

This section discussed the condition and calculation that result the increment point for the multipliers.

1. Players' Lives
   One of the core gameplay of tower defence games is to defend its castle (i.e. lives). If their lives is decreased, it means the players use bad or wrong strategy. For every lives which is decreased, our system will produced different multipliers for the game. Those calculations are shown in table 2. For example, the player lost 5 lives in a level. Thus, in the next level the status multiplier will be decreased by 0.12, spawn multiplier will be decreased by 0.10, gold multiplier will be increased by 0.09, status point will be

decreased by 2, and spawn point will be decreased by 2. This new value is calculated to help the player on the next level.

Table 2. Increment of Multipliers based on Players' Lives

Variables and Multiplier Names	Health Loss 0	1-3	4-7	8-10	>10
Status Multiplier	+0.07	-0.10	-0.12	-0.17	-0.20
Spawn Multiplier	+0.05	-0.08	-0.10	-0.14	-0.16
Gold Multiplier	-0.05	+0.07	+0.09	+0.12	+0.15
Status Point	+1	+0	-2	-3	-4
Spawn Point	+1	+0	-2	-3	-4

2. Enemies' Health
When an enemy spawns, the system will add its health to the accumulative enemies' health point of that level. Moreover, our system calculate all of the remaining enemies' health at the end of every level. After the level has finished, the system will compares the remaining enemies' health point with the accumulative enemies' health point. For example, an enemy with 10 health spawn 5 times. The accumulative enemies' health point is 50 health. In the first case, our towers kill all of the enemies. It means 0 of 50 (0%) enemies' health remaining. In the second case, our towers failed to kill all of the enemies. It leaves 5 health for every enemies. It means 25 of 50 (50%) enemies' health remaining. Thus, the status multiplier will be decreased by 0.1, spawn multiplier will be decreased by 0.075, gold multiplier will be increased by 0.075, and difficulties point global will be decreased by 8. The details of the increment of multipliers based on this condition are listed in the table 3.

Table 3. Increment of Multipliers based on Enemies' Health

Multiplier Names	Enemies' Health Remaining 0%	<4%	<10%	<20%	20-100%
Status Multiplier	+0.06	-0.04	-0.05	-0.07	-0.10
Spawn Multiplier	+0.05	-0.03	-0.04	-0.05	-0.075
Gold Multiplier	-0.04	+0.02	+0.03	+0.05	+0.075
Difficulties Point Global	+4	-2	-4	-6	-8

3. Skill Point Calculations
This calculation is computed when the players start the stage for the first time to initialize difficulty at the beginning of the game. Players that have advanced skills will have tougher enemy at first wave. The details of the increment of multipliers based on skill point calculations are listed in the table 4, 5, and 6. For example, the player chose to increase their support skill to level 4 and their offensive skill to level 3. Thus their gold multiplier will be increased by 0.1 and their status multiplier will be increased to 0.25.

Table 4. Increment of Multipliers based on Support Skill

Multiplier Names	Level 0-3	Level 4-5
Gold Multiplier	+0.5	+0.1

Table 5. Increment of Multipliers based on Offensive Skill

Multiplier Names	Level 1-3	Level 4-5
Status Multiplier	+0.25	+0.5

Table 6. Increment of Multipliers based on Defensive Skill

Multiplier Names	Level 1-3	Level 4-5
Spawn Multiplier	+0.1	+0.25

3.3 Calculation of the Formulas

After the multipliers has been calculated, those points are used to increase or decrease the difficulties of the next level. These formulas are used in our game resulting new difficulties and experiences for the players:

$$Status\ Point = Status\ Point * Status\ Multiplier \tag{1}$$

$$Spawn\ Point = Spawn\ Point * Spawn\ Multiplier \tag{2}$$

$$Gold\ Point = Gold\ Point * Gold\ Multiplier \tag{3}$$

These three formulas above are applied at the end of levels after considering players' lives, enemies' health, and skill points.

$$CPT = 20 + (30 * DPG/100) + SpwPo \tag{4}$$

This formula is used to determine how many enemy will spawn. The starting number of enemies point total (CPT) is 20. It means the minimum number of CPT is 20. Moreover, the number of enemy will also be affected by the difficulty point global (DPG) and the spawn point itself. The spawn point is calculated with the formula 2.

$$Bonus\ Gold = (50 * DPG/100) + GP * 2 \tag{5}$$

This formula is used calculate the bonus gold in the end of levels. The higher the difficult point global, the more gold the players will get. Moreover, the third formula is used to calculate the GP. Then, GP will be multiplied by 2. Finally, the result of that will be added to get the final number of bonus gold.

$$CHM = CHM * StPo + (20 * DPG/100) ** \tag{6}$$

Enemies health max (CHM) is manually arranged by the game designers. Every enemies could have different maximum health. This number will be multiplied with the status point, by using the formula 1, and takes 20% from difficult point global (DPG). The DPG will have small effect on the late levels, but it is necessary to make the game interesting and have various difficulties.

$$CA = CA * (StPo/4) + (20 * DPG/100) ** \tag{7}$$

This formula is used to calculate the enemies' armors. Status point, by using the formula 1, need to be divided by 4. By doing this, the number will not be too high when multiplied with CA. The final result also consider the DPG for the calculation.

Here are the description of the abbreviation in the formulas:

CPT	=	*Enemies Point Total*	*GP*	=	*Gold Point*
DPG	=	*Difficulties Point Global*	*StMul*	=	*Status Multiplier*
CHM	=	*Enemies Health Max*	*SpwMul*	=	*Spawn Multiplier*
CA	=	*Enemies Armor*	*GM*	=	*Gold Multiplier*
StPo	=	*Status Point*			
SpwPo	=	*Spawn Point*	***this apply to all new spawned enemies*		

3.4 Extra Rules for Dynamic Difficulty Adjustment (DDA) System

In order to make the games more balance, we add an extra rules in the game design. In every even level, we add one more additional point into status and spawn. This rule applies until the status and spawn points have not reached 19. We called this phase as the "training phase". This is the phase where we really know the skills of the players in the game. Nevertheless, the multiplier systems explained in the section 2.2 and 2.3 still able to change those points to become more than 19.

4. Research Result and Discussion

After we have implemented our DDA rules, conditions, and algorithms into our games, we test the gameplay to produce graph visualization to better understanding the results of our game design. There are three kinds of graphics that we have created: damage output, players' lives, and difficulties. As we have stated before, the remaining enemies' health and players' lives will affect the difficulties graph. For each of the graphic, we will test the best, average, and worst case type of gameplay (e.g. strategy, tower placement, upgrades, etc.). Furthermore, we will also compare them with the static difficulties balancing.

4.1 Best Case Type of Gameplay

At the best case type of gameplay, the players already know the game mechanic and play well at tower defence game. From the Fig.1 and Fig. 2, we can clearly see at the dynamic balancing damage output tend to be much various than static balancing. Nevertheless, sometimes the players cannot keep up their tower damages with the enemies' health point. For instance, in level 4 the towers failed to kill all of the enemies. Thus, we can see in the players' lives graph, they players now only have 18 lives in the level 5. Furthermore, the loss of lives only increase difficulties in small portions. We can look another example in the level 9. In this level, players is able to defend its castle and did not loss any lives. Hence, they have the same value of lives in the level 10. Furthermore, the difficulties in the level 10 increase with a significant amount that we can clearly see in the graph. The 'x' axis in the graph is level.

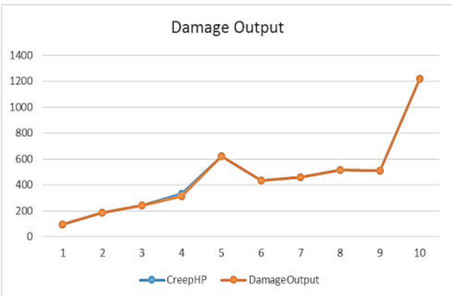

Fig. 1. Damage Output – Best Case (Static)

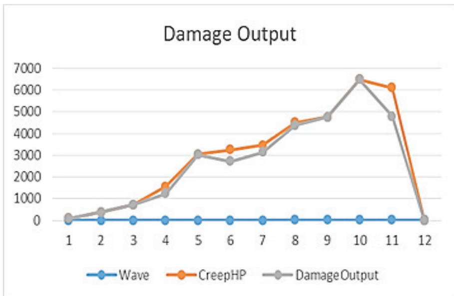

Fig. 2. Damage Output – Best Case (Dynamic)

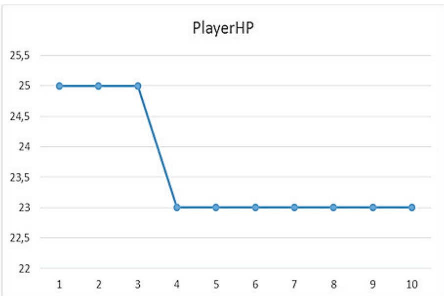

Fig. 3. Players' Lives – Best Case (Static)

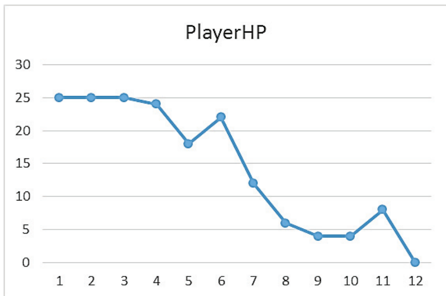

Fig. 4. Players' Lives – Best Case (Dynamic)

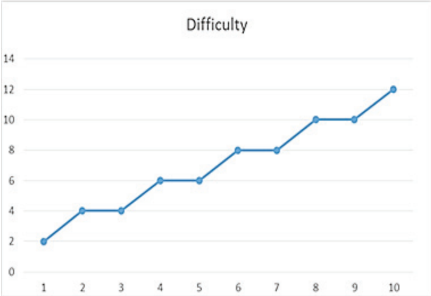

Fig. 5. Difficulty – Best Case (Static)

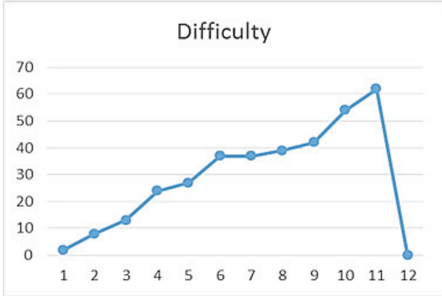

Fig. 6. Difficulty – Best Case (Dynamic)

4.2 Average Case Type of Gameplay

At the average case type of gameplay, the players already know the basic game mechanic. Nevertheless, they build the wrong towers and also put the towers in wrong places. In this case, the difficulties also have varied depends on the performance of the players. As we can see in the Fig. 12, the difficulties of the level 7 is decreased quite substantial. The reason for this is because in the level 6, the players' loss five lives. The lives value is referring to the Fig. 10. We can also see in the Fig. 8, the towers only inflict 1000 of 1500 total health of all enemies. Thus, we decreased the multipliers with the rules of 20-100% based on the section 2.2. In the level 8, the difficulties has increased. This indicates that the players did a good strategy or performance on the level 7. The players succeed to kill all of the enemies in the level 7. Hence, he/she did not loss any lives in that levels.

The increment and decrement of the difficulties in this case are the result of applying DDA in the game. With this system, novice players could learn the games with their pace and limited abilities. Once the players has mastered the basic and become advanced players, they would get new challenges and experiences every time they play better.

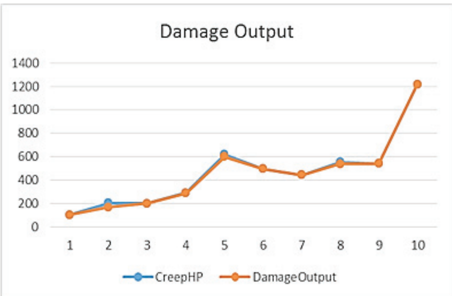

Fig. 7. Damage Output – Average Case (Static)

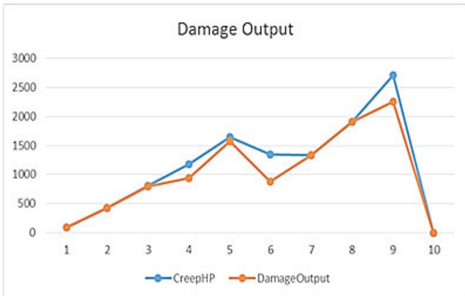

Fig. 8. Damage Output – Average Case (Dynamic)

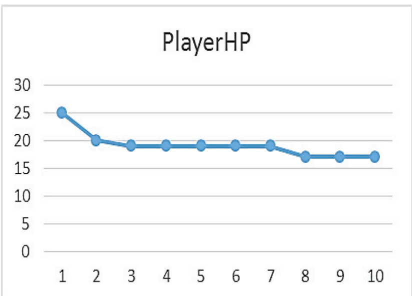

Fig. 9. Players' Lives – Average Case (Static)

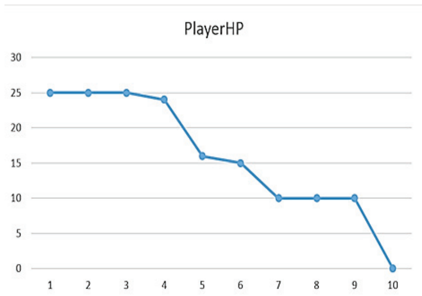

Fig. 10. Players' Lives – Average Case (Dynamic)

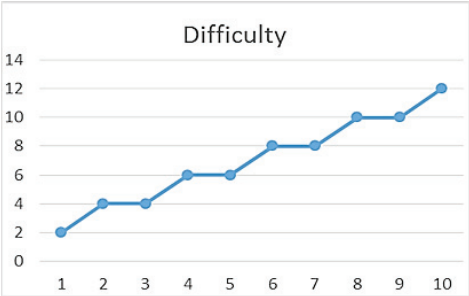

Fig. 11. Difficulty – Average Case (Static)

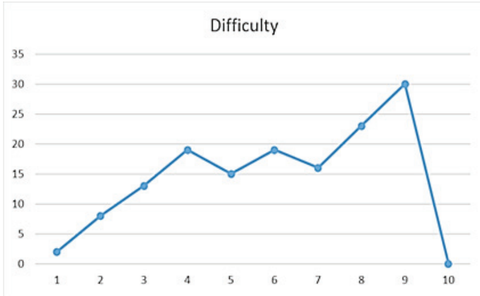

Fig. 12. Difficulty – Average Case (Dynamic)

4.3 Worst Case Type of Gameplay

At the worst case type of gameplay, the players did not know the basic game mechanic. They did not know the function of the towers, they also build wrong towers, and put the towers in the wrong places. In this case, the maximum number of difficulties is 30 in the level 8. While in the best case, the maximum number is 40 in the same level. This type of adjustment could make the players learn with their own knowledge because the game is not too hard for them.

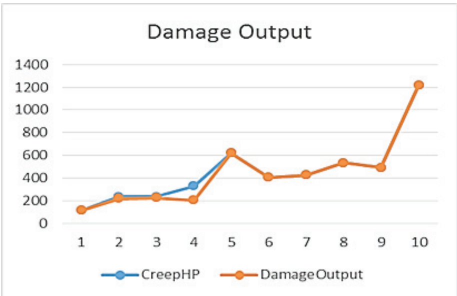

Fig. 13. Damage Output – Worst Case (Static)

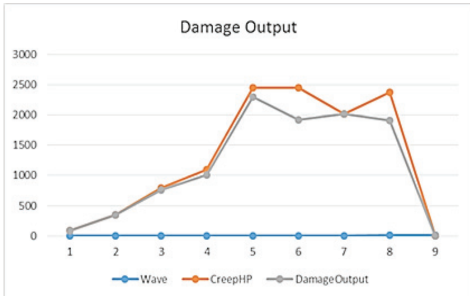

Fig. 14. Damage Output – Worst Case (Dynamic)

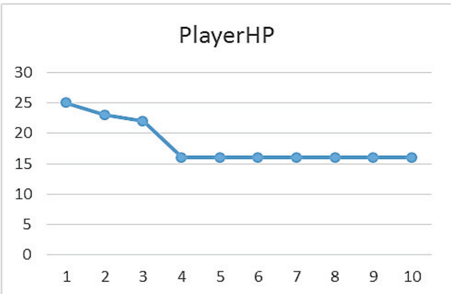

Fig. 15. Players' Lives – Worst Case (Static)

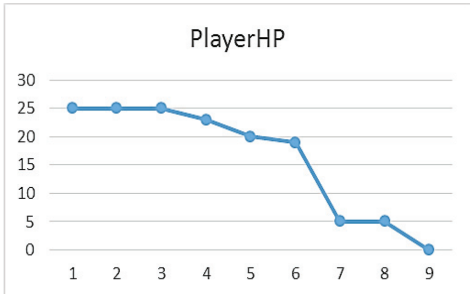

Fig. 16. Players' Lives – Worst Case (Dynamic)

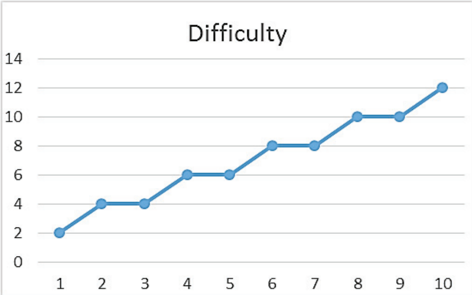

Fig. 17. Difficulty – Worst Case (Static)

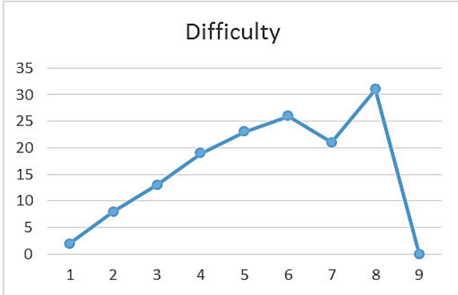

Fig. 18. Difficulty – Worst Case (Dynamic)

5. Conclusions and Future Works

In this paper, we have presented a dynamic difficulty adjustment by considering several condition of the players to ensure the game is balanced yet still welcoming and challenging for novice and advanced players respectively. We are using players' lives, enemies' health, and skill points to determine the multipliers that affect the difficulties of the game. Moreover, this DDA can also be applied to any similar tower defence games. We are using general characteristics such as lives, levels, skill points, etc. that most of tower defence games also have. For the future works, we are planning to make the DDA more dynamic by also considering tower placement component. Moreover, we also planning to readjust several initial attributes to make the game easier for the novice players.

References

1. Ernest Adams. 2009. Fundamentals of Game Design (2nd ed.). New Riders Publishing, Thousand Oaks, CA, USA.
2. Avery, P.; Togelius, J.; Alistar, E.; van Leeuwen, R.P., "Computational intelligence and tower defence games," Evolutionary Computation (CEC), 2011 IEEE Congress on , vol., no., pp.1084,1091, 5-8 June 2011.
3. Robin Hunicke, Vernell Chapman. 2004. AI for dynamic difficulty adjustment in games. In Proceedings of the 2004 Challenges in Game Artificial Intelligence AAAI Workshop. Northwestern University, San Jose, California, USA, 91-96.
4. Robin Hunicke. 2005. The case for dynamic difficulty adjustment in games. In Proceedings of the 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology (ACE '05). ACM, New York, NY, USA, 429-433.
5. Pieter Spronck, Marc Ponsen, Eric Postma. 2006. Adaptive game AI with dynamic scripting. Machine Learning. Kluwer. 217-248.
6. John E. Laird and Michael van Lent. 2000. Human-Level AI's Killer Application: Interactive Computer Games. In Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence. AAAI Press 1171-1178.
7. Forbus, K.D.; Laird, J., "AI and the entertainment industry," Intelligent Systems, IEEE, vol.17, no.4, pp.15-16, July-Aug. 2002.
8. S.L. Tomlinson, Working at thinking about playing or a year in the life of a games AI programmer, in: Q. Mehdi, N. Gough, S. Natkin (Eds.), Proceedings of the 4th International Conference on Intelligent Games and Simulation, GAME-ON 2003, EUROSIS, Ghent, Belgium, 2003, pp. 5–12.
9. Guy Hawkins, Keith Nesbitt, and Scott Brown. 2012. Dynamic difficulty balancing for cautious players and risk takers. Int. J. Comput. Games Technol. 2012, Article 3 (January 2012), 1 pages.
10. Alexander Baldwin, Daniel Johnson, and Peta A. Wyeth. 2014. The effect of multiplayer dynamic difficulty adjustment on the player experience of video games. In Proceedings of the extended abstracts of the 32nd annual ACM conference on Human factors in computing systems (CHI EA '14). ACM, New York, NY, USA, 1489-1494.