Upgrade balancing using simulated annealing

Leon Berghorst
500704147
Leon.Berghorst@hva.nl

Timo Veenman
500709657
Timo.Veenman@hva.nl

Amanda Herders
500708886
Amanda.Herders@hva.nl

1. Abstract

Game balancing in terms of fairness and competitiveness is an important factor in user satisfaction (Tavassolian, Stanley & Gutwin, 2013). To adapt balancing in our game, we have applied a simulated annealing algorithm, which searches for the best value. This is applied in our game, Tower defence, where the player has to protect his base from enemy ships. The player can place towers and upgrade these. To make the upgrades equally appealing to the player, we hope to balance the value of the upgrades in terms of fairness. To see if the algorithm is working and if the value balances, we have set up a research where we have two versions of the same game, but save the values and their ratings in two separate databases. The results show that the values are balancing and that the some of the upgrades in both databases are around the same value.

2. Introduction

This paper discusses the implementation of an automated game design concept in the game Tower Defence. This concept is based on mainly two papers in which game balancing is discussed and our own interests. The concept comprises of balancing the upgrades for placeable towers.

3. Inspiration

For this paper the main inspirations are *Dynamic Game Balancing: an Evaluation of User satisfaction* (Andrade, Ramalho, Gomes & Corruble, 2006) and *The effect of temporal adaptation granularity and game genre on the time-balancing of adaptive time-varying minigames* (Tavassolian et al., 2013). The first paper describes how user satisfaction is influenced by game balancing, while the latter goes more into depth on how to measure if game balancing in term of fairness or competitiveness has any effect. These papers piqued our interest to approach the subject balancing in terms of fairness in a game and research whether or not the balancing is working. We first wanted to modify the actual projectiles you were shooting, For this we used the paper from the Galactic Arms race **(**Hastings, Guha & Stanley 2009), sadly we found out that it was out of scope because of the lack of time, which is why we chose to alter the tower upgrades instead having to change the way the turret fires we added different boosts i.e. increased health or fire rate.

4. Tower Defence

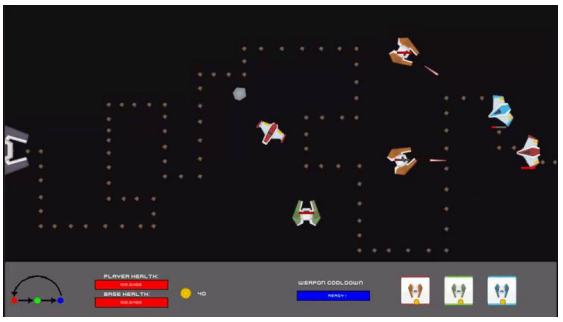

Figure 1. A screenshot of Tower Defence

The Tower Defence game implements a concept based on the aforementioned. The game is a tower defence game, in which the player and the enemies can have three different colours: red, blue or green. Red beats green, green beats blue and blue beats red. Unlike the player, the enemies cannot switch between the colours. The player receives money when destroying enemies and protecting his base. With this money it is possible to place towers that shoot bullets at enemies and upgrade placed towers with seven different types of upgrades.

5. Upgrade balancing

Each tower can be upgraded. The value of each upgrade is calculated based on player input from previous rounds. This input is received at the end of each game, where the player can give each upgrade a rating. These ratings are saved in a database. For each rating, a specific range will be used to calculate the next value of the upgrade. So when an upgrade is favored, the next value of the upgrade will be chosen from a range of five% of the previous value, whereas an upgrade with rating one will have a value from the range 20%.

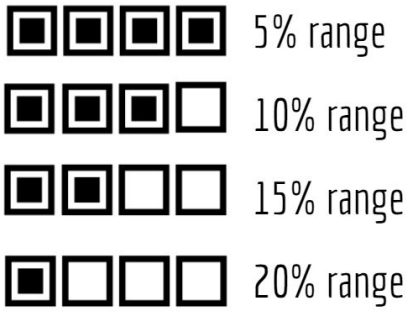

Figure 2. The ranges for each rating

The idea is for the upgrades to be equally appealing for the player. We use an algorithm called Simulated annealing to help us change the values of each upgrade at the start of each game. This algorithm helps to find an optimal value for a certain function (Geltman, 2014). We use this algorithm to find an optimal balance between the values for our upgrades and the cost of the upgrades. We hope to achieve a situation where all upgrades are valued equally by the player. This way all the created content will be used by the players.

6. Research

6.1 Hypothesis

To find the optimal value which is appreciated by the majority of the players, we have set up a small research. In this research we want to test if it is possible to find an optimal value for our upgrades with the help of the Simulated annealing algorithm. Our hypothesis is that after running the algorithm several times on both machines, the final value on both machines will be close to each other .

6.2 Procedure

For our research we have created two identical versions of the Tower Defence game, each with their own database to save the player input in. Each version of the game has been played 25 times by different participants. The database contains the data for each generated value for the upgrades and the fairness rating the player has giving the upgrade value (on a scale of one to four). After both games have been played 25 times we have taken the data from both databases and we have compared the average of the last three fair upgrades from both databases for each type of upgrade.

6.3. Design and analyse

To extract the needed information out of the results, the data has been visualized in graphs. The red and blue color indicate the two different databases

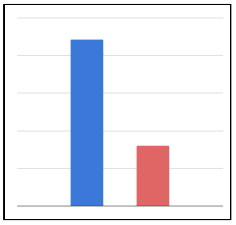

Figure 3. Agro upgrade

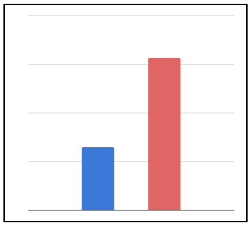

Figure 4. Money upgrade

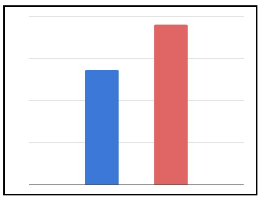

Figure 5. Fire-rate upgrade

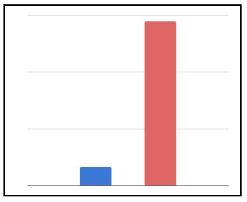

Figure 6. Damage upgrade

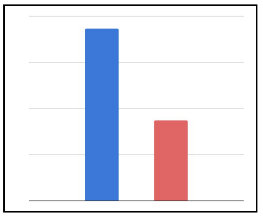

Figure 7. Range upgrade

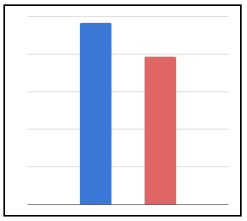

Figure 8. Regen upgrade

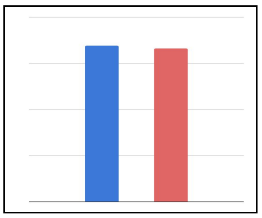

Figure 9. Health upgrade

7. Conclusions

After analyzing the data, we came to the conclusion that our hypothesis was partially correct as it only applied to a few of the upgrades. We also found out that during testing whenever the values became really low it was hard to get higher values again. This was caused because we are using a percentage range to find new values, this means that if you have a small number with the same percentage range it fluctuates less than if you have a higher number. This could be one of the reasons why some of the values did not match each other. Of course it could also be that the preference of the players differ.

8. Future work

During this course we only worked on balancing the values of the upgrades for the turrets. However we think that having these upgrades use a set improvement factor is a bit weird, so we would like to find the optimal improvement factor for each upgrade in the same way as we did for the actual improvement value. Further we think that the enemies are too easy and we would like to make them more difficult over time. We could try and find the best value via playtesting a lot but since we are finding the other values via algorithms we think it would make it easier and less time consuming to find the optimal difficulty curve for the enemies.

9. Contribution

The original game was made by Leon Berghorst, however it was expanded by Timo Veenman, Amanda Herders and Leon Berghorst. Timo worked on the upgrade UI and the improvement factor of the upgrades. Leon worked on the database and algorithm. And Amanda worked on the enemies with increasing difficulties and together with Leon on improvements on the database. For the research, all three set up the research and conducted the user experiment. Afterwards all three analyzed the data and wrote the rapport tailored to the received data. Timo made the graphs and wrote about the game, Leon added to the introduction what our initial concept was and what we used to research this. and future work and Amanda about the research, conclusion and future work. However, everyone has checked and improved the content where change was needed.

10. References

Tavassolian, A., Stanley, K., & Gutwin, C. (2013). The effect of temporal adaptation granularity and game genre on the time-balancing abilities of adaptive time-varying minigames. Sciencedirect.com. Retrieved 16 June 2017, from http://www.sciencedirect.com/science/article/pii/S1875952113000098

-

Geltman, K. (2014). The Simulated Annealing Algorithm. Katrinaeg.com. Retrieved 19 June 2017, from http://katrinaeg.com/simulated-annealing.html

Andrade, G., Ramalho, G., Gomes, A., & Corruble, V. (2006). Dynamic Game Balancing: an Evaluation of User Satisfaction. Retrieved 16 June 2017, from https://www.aaai.org/Papers/AIIDE/2006/AIIDE06-005.pdf

Hastings, E. J., Guha, R. K., & Stanley, K. O. (2009). Automatic Content Generation in the Galactic Arms Race Video Game. Florida: School of Electrical Engineering and Computer Science University of Florida.