Creating A Real-Time Strategy Game Using Simple Mathematical Equation

A Master's Degree Project Proposal

**Submitted by:**

Thompson Lee

On September 10th, 2015

To the IMGD Steering Committee

**Project Committee:**

Project Supervisor: Dean O'Donnell

Project Reader: Brian Moriarty

Project Reader: Charles Rich

Project Thumbnail

This project is to be built as a real time strategy game, where players use simple and easy-to-grasp math equations for generating units and unit attributes. This project aims to pave the way for future procedurally generated real time strategy game unit balancing.

Problem Statement

1. **Research Question**

A typical real time strategy game requires a lot of gameplay testing to see if units made are balanced for players to play with. Making the balancing process more streamlined for simulating real time strategy units can be done by running algorithms to determine the most optimal unit attributes given. This process allows access to more unit diversity and game designs in the real time strategy genre.

This project needs investigate if this is a viable solution to generating and balancing unit attributes. The best way to do this is by building a real time strategy game using simple mathematical equations to determine usable unit's attributes, and assess the outcome.

1. **Previous Work**

Procedural content generation in real time strategy games is one of the most interesting challenges in the video game development process.[[1]](#footnote-1) The dynamics in real time strategy games alone vary greatly, especially when involving multiple players of varying skill levels and backgrounds. These dynamics can be treated differently depending on how the contents are procedurally generated, therefore paving the way for many possible research routes.

There are video games that have done research in procedural content generation, in which some of them were able to use procedural content generation methods and techniques, but the usage is somewhat limited to a particular type of game content.[[2]](#footnote-2) It has also been proven that it is possible to have automated content generation in mainstream games.[[3]](#footnote-3) Notable examples include *Minecraft[[4]](#footnote-4)* and *Mini Metro[[5]](#footnote-5)*, in which the former uses procedural content generation to generate terrain, and the latter uses procedural audio generation.

There has been research done on production capability for different species of units in a game. Units that rely only on damage per second is not the best, but rather a mix of other unit attributes, such as hit points, defense points, along with other properties, is suggested.[[6]](#footnote-6) Other research involves using procedural map generations built to fulfill requirements in order to maintain interesting and appealing games, suggesting that game balancing can be perfectly achieved only on extremely dull games.[[7]](#footnote-7) It also theorizes having moderate dynamics and moderate balancing can give ample stimuli to players to expand and to seek their enemies.

Games that have moderate dynamics and balancing can be used as references. *Total War: Shogun 2*[[8]](#footnote-8), *Total War: Attila*[[9]](#footnote-9), and *Multiwinia*[[10]](#footnote-10) are all real time strategy games where unit compositions are similar, and require the players to use strategic unit troop placements on the battlefield to win battles. In these games, the battlefield area is large enough to provide ample stimuli for players to venture out and prepare for battle.

Games with more complicated unit attributes and geographical properties that affect player decisions would be *Starcraft II*[[11]](#footnote-11), *Warcraft III*[[12]](#footnote-12), and *Total Annihilation*[[13]](#footnote-13). In these games, unit attributes are affected by unit dynamic properties (speed, regeneration, and cooldowns), which are incrementally increased through tech upgrades. It has been shown that unit attributes can determine the outcome of a real time strategy multiplayer game session[[14]](#footnote-14). Environmental obstacles used in these games, which can lead to players not being able to spot the enemies at a glance can also affect the outcome of the player game session. For example, trees with enemies behind it can block the player’s view from seeing the enemies.

Similarly, there are some real time strategy games, such as *Auralux*[[15]](#footnote-15), which utilizes map layouts designed with a blend of *Footmen Wars*[[16]](#footnote-16)in mind. Research has been done exploring map layout and balance in real time strategy games[[17]](#footnote-17), made similarly as *Auralux*.

The game, *Auralux*, provides the basis of linear upgrade paths that players can use during gameplay, as well as taking into account of the map layout. *Footmen Wars* provides the structure of the game, in which each units of different factions have attributes that players can upgrade accordingly, but ultimately, the players can only use that unit for the rest of the game.

1. **The Project**

The proposed method of evaluating real time strategy gameplay consists of a single unit type which uses a simple mathematical equation to calculate the unit’s attributes and properties. It is easier to come up with a sequential relationship than any other complex relationships[[18]](#footnote-18), and is easily deterministic when compared with other units of the same attributes and properties. Structured to be a 2-players only multiplayer network game, it will put emphasis on unit placement and macro-management, will avoid unit power-ups, special abilities, and anything that reduces player’s attention, such as micro-management[[19]](#footnote-19).

The project will be only built for human multiplayer with no A.I. bots involved. If the game is built for single player, the game would then be required to provide an A.I. bot for the player to compete against. Creating a new A.I. bot for this purpose is too costly, while getting another human player to play is quicker and more efficient. Therefore, we eliminate single player scenarios from this project, and concentrate on human interactions only. Future work can create A.I. bots for single player gameplay and/or simulation of similar real time strategy games.

There are two viable options of game designs the game project can be built around of. The first option is by considering “massing,” in which the players continuously spawn and build the exact same units overwhelming their opponents. It runs the risk of players becoming burdened by the game if not designed correctly[[20]](#footnote-20), and it depends entirely on player skills[[21]](#footnote-21). The second option is by considering unit group placements, and a bit of micromanagement involved, to determine the outcome of the scenario, used in some games such as *Command and Conquer*[[22]](#footnote-22)and *Supreme Commander*[[23]](#footnote-23). The players will not have to worry about skills and will put more focus in strategic planning and execution. The main factor to consider the second method is it can be applied to games which use only one type of unit to play, making it easier to evaluate the results.

The game is given a mathematical equation, or an algorithm, that it can use to generate unit attribute values. These attribute values are then set, therefore the units spawned into the game scene will then have the attributes calculated.

|  |  |  |
| --- | --- | --- |
|  |  | Fig.1 |

For example, if the Fibonacci sequence algorithm (as shown in Fig.1) is plugged into the game, the resulting unit health attribute values will generate for 10 levels, or tiers. For each level, the unit health will increase up to the number in the sequence that is iterated. Note that it does not have to be a series of numbers from the Fibonacci sequence. It can be any other series of numbers, ranging from randomly generated (pseudorandom number generator, Fig.2[[24]](#footnote-24)), or a sine value ranging from -1 to 1 (sine wave, Fig.3).

|  |  |  |
| --- | --- | --- |
|  |  | Fig.2 |
|  |  | Fig.3 |

The project will start with a simple mathematical equation that can be used as a multiplier for leveling up units for each tier level. Each tier level can then use the same mathematical equations to level up, until the unit has reached the maximum amount of levels or tiers allowed in the game. In this way, all upgradeable-tiered units will have consistent, linear (or exponentially, depending on the mathematical equations themselves) relationships between two neighboring tech tree nodes, which represents an upgrade that a unit can have in the game.

The mathematical equations mentioned may change to fit within the scope of the production of the project. Since the only concerned component about the project is by using mathematics to determine unit attributes in real time strategy games, it is not to be used as a reliable method to assess if the game is playable and enjoyable, depending on what mathematical formulae were used. However, it will provide a leeway for future improvements if there is time.

1. **Production and Planning**

It is to be built using Unity3D 5, a popular game engine with a well-written documentation, a fully active community, and easy-to-obtain online tutorials and instructional videos. Since Unity3D 5 is in constant development, future releases, as well as documentations of new features, may affect the whole project.

The scheduled game development plan for this project is shown on the next page.

|  |  |  |
| --- | --- | --- |
| **Milestones** | **Deadline** | **Description** |
| Understand Unity3D | 1-Jun-15 | Need to familiarize Unity development |
| Tech Milestones | 15-Jun-15 | Components for single player clients |
| Networking | 30-Jun-15 | Make two clients connect to each other |
| Unity 5.1 and 5.2 | 1-Oct-15 | New networking features available |
| Prototype | 30-Oct-15 | Implement primitive components |
| Alpha | 15-Nov-15 | Winning/Losing conditions |
| Beta | 30-Dec-15 | Start polishing |
| Frequent Playtesting | 15-Jan-16 | Constant iterative development |
| Assessment | 15-Feb-16 | Come up with a way to assess experiences |
| Unscheduled | 1-May-16 | Future planning in case of emergency |

In order to be sure the project is as accurate as possible, it requires constant playtesting, which is worth the effort, as the project can then be tweaked accordingly to provide accurate results over time once the assessments are done.

Evaluation

This project will produce a playable project game, along with results of testing to see if mathematical equations can allow procedurally generated units be used in real time strategy games, regardless of complexity. The results will be a cumulative analysis of overall gameplay experiences of players playing the project game.

# References

Adams, D. (2006, September 11). *Company of Heroes Review*. Retrieved from IGN: http://www.ign.com/articles/2006/09/11/company-of-heroes-review-2

Adams, E. (1998, October 16). *Designer's Notebook: A Symmetry Lesson*. Retrieved from Gamasutra: http://www.gamasutra.com/view/feature/131699/designers\_notebook\_a\_symmetry\_.php

Bangay, S., & Makin, O. (2013, September 23-25). Modelling Attribute Dependencies in Single Unit. *Games Innovation Conference (IGIC), 2013 IEEE International*, 20-26.

Bergensten, J. (2008, November 26). *RTS Game-play Part 5: Introduction to Unit Balancing*. Retrieved from Oxeye Game Studio News & Development Blog: http://www.oxeyegames.com/rts-game-play-part-5-introduction-to-unit-balancing/

Blizzard Entertainment. (2002, July 3). *Warcraft 3: The Reign of Chaos*. Retrieved from Blizzard Entertainment: http://us.blizzard.com/en-us/games/war3/

Blizzard Entertainment. (2015). *StarCraft II*. Retrieved from Blizzard Entertainment: http://us.battle.net/sc2/en/

Dinosaur Polo Club. (2015, August 28). *Mini Metro - Beta31: Audio!* Retrieved from Steam Community: http://steamcommunity.com/games/287980/announcements/detail/800867231024886989

Dulin, R. (1997, October 1). *Total Annihilation Review*. Retrieved from Gamespot: http://www.gamespot.com/reviews/total-annihilation-review/1900-2535174/

Fayard, T. (2007). Using a Planner to Balance Real Time Strategy Video Game. *Workshop on Planning in Games , ICAPS, vol. 2005.*, 1-8.

Gallegos, A. (2011, November 23). *Minecraft Review*. Retrieved from IGN: http://www.ign.com/articles/2011/11/24/minecraft-review

Goetz, P. (2006, August 23). *Too Many Clicks! Unit-Based Interfaces Considered Harmful*. Retrieved from Gamasutra: http://www.gamasutra.com/view/feature/1839/too\_many\_clicks\_unitbased\_.php

Griliopoulos, D. (2008, September 16). *Multiwinia UK Review*. Retrieved from IGN: http://www.ign.com/articles/2008/09/16/multiwinia-uk-review

Hafer, T. (2015, February 12). *Total War: Attila Review*. Retrieved from IGN: http://www.ign.com/articles/2015/02/12/total-war-attila-review

Hastings, E. J., Guha, R. K., Member, L., IEEE, & Stanley, K. O. (2009, December). Automatic Content Generation in the Galactic Arms Race Video Game. *IEEE Trabsactions on Computational Intelligence and AI in Games, Vol. 1, No. 4*, 245-263.

Johnson, D. M. (2013, September 7). *Real-Time Strategy “Level Design”*. Retrieved from Ultima Ratio Regum: http://www.ultimaratioregum.co.uk/game/2013/09/07/real-time-strategy-level-design/

Lara-Cabrera, R., Cotta, C., & Fernández-Leiva, A. J. (2013). A Procedural Balanced Map Generator with Self-adaptive Complexity for the Real-Time Strategy Game Planet Wars. *EvoApplications 2013, LNCS 7835*, 274–283.

Lara-Cabrera, R., Cotta, C., & Fernández-Leiva, A. J. (2014). On balance and dynamism in procedural content generation with self-adaptive evolutionary algorithms. *Springer Science+Business Media Dordrecht*, 157–168.

Lara-Cabrera, R., Nogueira-Collazo, M., Cotta, C., & Fernández-Leiva, A. J. (2015). Procedural Content Generation for Real-Time Strategy Games. *International Journal of Artificial Intelligence and Interactive Multimedia, Vol. 3, Nº 2.*, 40-48.

Mark Hendrikx, S. M. (2013, February). Procedural Content Generation for Games: A Survey. *ACM Transactions on Multimedia Computing, Communications and Applications, Vol. 9, No. 1, Article 1*, 22.

Onyett, C. (2007, February 16). *Supreme Commander Review*. Retrieved from IGN: http://www.ign.com/articles/2007/02/16/supreme-commander-review-2

Onyett, C. (2010, March 18). *Command & Conquer 4 Review*. Retrieved from IGN: http://www.ign.com/articles/2010/03/18/command-conquer-4-review?page=1

Onyett, C. (2011, March 16). *Total War: Shogun 2 Review*. Retrieved from IGN: http://www.ign.com/articles/2011/03/17/total-war-shogun-2-review

Parker, J. (2013, May 10). *Auralux Review*. Retrieved from CNET: http://www.cnet.com/products/auralux/

StrategyWiki. (2014, October 4). *Warcraft III: Reign of Chaos/Footmen Wars*. Retrieved from Wayback Machine: https://web.archive.org/web/20141004065215/http://strategywiki.org/wiki/Warcraft\_III:\_Reign\_of\_Chaos/Footmen\_Wars

The Numerical Algorithms Group Ltd. (2012). *Random Number Generators.* Retrieved September 17, 2015, from NAG Library Manual, Mark 23 Online Documentation: http://www.nag.co.uk/numeric/fl/nagdoc\_fl23/pdf/G05/g05intro.pdf

Wayward Strategist. (2014, December 18). *Random Thoughts on Resource Management in RTS*. Retrieved from Wayward Strategist: http://waywardstrategist.com/2014/12/18/random-thoughts-on-resource-management-in-rts/

1. (Lara-Cabrera, Nogueira-Collazo, Cotta, & Fernández-Leiva, 2015) [↑](#footnote-ref-1)
2. (Mark Hendrikx, 2013) [↑](#footnote-ref-2)
3. (Hastings, Guha, Member, IEEE, & Stanley, 2009) [↑](#footnote-ref-3)
4. (Gallegos, 2011) [↑](#footnote-ref-4)
5. (Dinosaur Polo Club, 2015) [↑](#footnote-ref-5)
6. (Fayard, 2007) [↑](#footnote-ref-6)
7. (Lara-Cabrera, Cotta, & Fernández-Leiva, On balance and dynamism in procedural content generation with self-adaptive evolutionary algorithms, 2014) [↑](#footnote-ref-7)
8. (Onyett, Total War: Shogun 2 Review, 2011) [↑](#footnote-ref-8)
9. (Hafer, 2015) [↑](#footnote-ref-9)
10. (Griliopoulos, 2008) [↑](#footnote-ref-10)
11. (Blizzard Entertainment, 2015) [↑](#footnote-ref-11)
12. (Blizzard Entertainment, 2002) [↑](#footnote-ref-12)
13. (Dulin, 1997) [↑](#footnote-ref-13)
14. (Bangay & Makin, 2013) [↑](#footnote-ref-14)
15. A minimalistic real time strategy game for Android, based in outer space. (Parker, 2013) [↑](#footnote-ref-15)
16. A real time tactics custom map game for the real time strategy game, *WarCraft III* and its expansion, *WarCraft III: The Frozen Throne*. (StrategyWiki, 2014) [↑](#footnote-ref-16)
17. (Lara-Cabrera, Cotta, & Fernández-Leiva, A Procedural Balanced Map Generator with Self-adaptive Complexity for the Real-Time Strategy Game Planet Wars, 2013) [↑](#footnote-ref-17)
18. (Adams E. , 1998) [↑](#footnote-ref-18)
19. (Wayward Strategist, 2014) [↑](#footnote-ref-19)
20. (Goetz, 2006) [↑](#footnote-ref-20)
21. (Bergensten, 2008) [↑](#footnote-ref-21)
22. (Johnson, 2013) [↑](#footnote-ref-22)
23. (Onyett, Supreme Commander Review, 2007) [↑](#footnote-ref-23)
24. (The Numerical Algorithms Group Ltd., 2012) [↑](#footnote-ref-24)