# Department of Computing Degree Project Proposal

**Name: Course: Size: double**

**Discussed with (lecturer):**

## Current Modules (and previous modules if computing or direct entrant)

CO2301, CO2401, CO2402, CO2403, CO2409, CO2411, CO2602

CO3301, CO3303, CO3402, CO3717. CO3808.

## The Project Title

Creating a more challenging and responsive RTS AI player.

## Project Context

Current RTS AIs build their base on a semi random placement frequently resulting in buildings blocking the entrance to others and a lack of focussed defence that is easily overpowered by even a moderately skilled player including placing defensive buildings in the middle or worse at the back of the base where they are ineffective. Additionally attack waves are usually programmed in advance and will never change between plays or to respond to the player. The end result is a poor AI player that needs a serious handicap to compete. My goal is to make an AI that can compete to some degree on even terms. This AI will build it’s base sensibly and adapt to the state of the level, more efficiently using limited resources rather than relying on having more than the player.

## Specific Objectives

* Create an AI that will check for blocking before placing a building.
* Focus the defensive units in key locations rather than evenly throughout the base.
* Allow the definition of base to adapt depending on map conditions.
* Unit build queue will respond to the player’s build.
* Run reasonably on a standard PC.

## References

* Avery, P., Louis, S., & Avery, B. (2009). **Evolving coordinated spatial tactics for autonomous entities using influence maps.** Computational Intelligence and Games, 2009. CIG 2009. IEEE Symposium on, 341-348. doi:10.1109/CIG.2009.5286457
* Beume, N., Hein, T., Naujoks, B., Piatkowski, N., Preuss, M., & Wessing, S. (2008)**. Intelligent anti-grouping in real-time strategy games.** Computational Intelligence and Games, 2008. CIG '08. IEEE Symposium on, 63-70. doi:10.1109/CIG.2008.5035622
* Hagelback, J., Johansson, S., & Preuss, M. (2010**). AI and computational intelligence for real-time strategy games**. Computational Intelligence and Games (CIG), 2010 IEEE Symposium on, 1-4. doi:10.1109/ITW.2010.5593384
* Haibo Wang, Ng, P. H. F., Ben Niu, & Shiu, S. C. K. (2009). **Case learning and indexing in real time strategy games.** Natural Computation, 2009. ICNC '09. Fifth International Conference on, , 1 100-104. doi:10.1109/ICNC.2009.729
* Ji-Lung Hsieh, & Sun, C. -. (2008). **Building a player strategy model by analyzing replays of real-time strategy games.** Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on, 3106-3111. doi:10.1109/IJCNN.2008.4634237
* Jon Lau Neilsen, Benjamin Fedder Jensen, Tobias Mahlmann, Julian Togelius, and Georgios N. Yannakakis. (2014). **AI for general strategy game playing.** In Marios C. Angelides, Harry Agius. (Ed.), Handbook of digital games, 1st ed. (pp. 274-304)
* Leow Chin Leong, Gan Kim Soon, Tan Tse Guan, Chin Kim On, Alfred, R., & Anthony, P. (2013). **Self-synthesized controllers for tower defense game using genetic programming. Control System,** Computing and Engineering (ICCSCE), 2013 IEEE International Conference on, 487-492. doi:10.1109/ICCSCE.2013.6720014
* Miles, C., Quiroz, J., Leigh, R., & Louis, S. J. (2007**). Co-evolving influence map tree based strategy game players.** Computational Intelligence and Games, 2007. CIG 2007. IEEE Symposium on, 88-95. doi:10.1109/CIG.2007.368083

## Potential Ethical or Legal Issues

There are no potential ethical or Legal Issues.

## Resources

No unusual resources will be required

## Potential Commercial Considerations - Estimated costs and benefits

Primary cost will be my time - 3-400 man hours for a developer. Benefits will be a better experience in single player RTS type games although elements will be transferable to various other more industry related applications for planning.

## Proposed Approach

\* Reduce game to RTS minimum – build on timers, units move automatically towards base. 3 units – light, medium, heavy – light < medium < heavy < light. Block graphics. At least for the early implementation reduce the game to almost 2 player tower defence. Resource management is desirable but not vital.

\* Pathfind (best first? A\*?) to enemy base, find closest minima of path width. Mark as a chokepoint. Repeat with chokepoint closed until no path can be found. Contained area is defined as base (zone of influence)

\* Ensure no placed buildings block path to enemy base

\* Concentrate defence around the choke point. (again – zone of influence, primary overlap on the choke point)

\* Keep running totals of incoming units and build correct counters in proportion to running total

\* Heuristic: number of units that reach each base. Given building on a timer the possible score should be fairly constant so long as test length is constant.

\* implement a traditional “random” base building AI as a control.

\* Stretch: have the AI defend from other than just the closest choke point if it is viable.

\* Stretch: include air units.