

DESIGNING AND DEVELOPING AN INTELLIGENT CONGKAK

A project report submitted to the
Faculty of Information Communication Technology
in partial fulfilment of the requirement for the degree
Master of Science (Information Communication Technology)
Universiti Utara Malaysia

By
Muhammad Safwan Bin Mohd Shahidan (s805747)

PERMISSION TO USE

In presenting this project report in partial fulfillment of the requirements for a postgraduate degree from Universiti Utara Malaysia, I agree that the University Library may make it freely available for inspection. I further agree that permission for copying of this project report in any manner, in whole or in part, for scholarly purpose may be granted by my supervisor(s) or, in their absence by the Dean of the Graduate School. It is understood that any copying or publication or use of this project report or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to Universiti Utara Malaysia for any scholarly use which may be made of any material from my project report.

Requests for permission to copy or to make other use of materials in this project report, in whole or in part, should be addressed to

Dean of Graduate School
Universiti Utara Malaysia
06010 UUM Sintok
Kedah Darul Aman.

ABSTRAK

Congkak ialah permainan tradisional Malaysia dan ia terdedah kepada risiko untuk dilupakan jika kewujudannya tidak diambil serius, tetapi setakat kini tinjauan teks-teks rujukan tidak menjumpai sebarang publikasi yang menyatakan penggunaan algoritma neural-network (NN) ke atas permainan Congkak. Oleh itu projek ini ingin untuk menangani isu ini dengan membina satu sistem Congkak dengan NN dan juga cuba untuk menjawab persoalan-persoalan formal berikut: “Apakah fungsi penilaian Congkak yang sesuai untuk melatih NN bagi permainan Congkak?” (contohnya: adakah penilaian Congkak melalui mengiraan buah terkumpul lebih baik dari penilaian melalui pengiraan 'rumah-terbakar'?) dan “Bolehkah prestasi bagi algoritma Min-Max & Alpha-Beta cut-off (MM) ditingkatkan jika NN digunakan sebagai sejenis teknik 'forward-pruning' untuk MM?”. Permasalahan ini diselesaikan dengan membina satu sistem Congkak berdasarkan kerja-kerja terdahulu yang berkaitan dengan sistem Mancala dan sistem NN, dan kemudiannya merekod prestasi algoritma yang terlibat untuk membuat kesimpulan. Hasilnya: projek ini berjaya mencipta satu sistem Congkak dengan 3 jenis agen kecerdasan buatan (AI agent), dan mendapati bahawa gabungan NN dan MM adalah lebih perlahan daripada MM semata-mata.

ABSTRACT

Congkak is the nation's traditional game which could soon be forgotten if no serious attention is given to it, but literature survey has not yet found any research publication that mentioned the use of neural network algorithm (NN) on Congkak. Therefore the project want to try to rectify this issue by trying to develop an Intelligent Congkak System that also implemented NN and try answer research question such as this: “What is the best Congkak evaluation function for training NN for game playing?” and “Can Min-Max algorithm (MM) be speeded up by using NN as a forward-pruning method?”. This issues can solved by programming the Congkak system based on previous work on Mancala and NN system, and then recording the performance of the related algorithm. As a result: the project had created a Congkak system that had featured 3 Artificial Intelligence (AI) agent, and discovered that the combination of NN and MM is slower than MM alone.

ACKNOWLEDGEMENT

Firstly I would like to thank all the people in the world that have willingly shared their knowledge freely on the internet. Without them the internet would have been empty and the project would fail. Thank you for sharing.

I would also like to thank Madam Latifah binti Abdullah for sharing with me the guideline & protocol for project writing and also allowed me to extend my project report submission date, and I would also like to thank both my project supervisor Miss Noraziah binti Che Pa and Miss Aniza binti Mohamed Din for giving me the encouragement & helpful comment and for checking and suggesting to me the correction for my project, and Dr. Yuhanis for evaluating my project, and also a friend Fadi Shaar Abdulghani who had helped me by sharing the guideline of the project proposal. Without them the project would have failed. Thank you for all the help.

I would also like to thank Ministry of Higher Education (MOHE) for funding my study (under Mini Budget 2009), and also both my parent for giving me the financial support, and also my cousin Rizal Harun for helping me with everything. Life would be difficult if without them. Thank you.

TABLE OF CONTENT

PERMISSION OF USE.....	i
ABSTRACT (MALAY).....	ii
ABSTRACT.....	iii
ACKNOWLEDGEMENT.....	iv
TABLE OF CONTENT.....	v
LIST OF TABLE.....	viii
LIST OF FIGURE.....	ix
CHAPTER 1: INTRODUCTION.....	1
1.1.Problem Statement.....	2
1.2.Research Question.....	3
1.3.Goal and Objective of the Project.....	4
1.4.The Significance of the Project.....	4
1.5.Scope, Assumption and Limitation of the Project.....	5
1.5.1.Scope.....	5
1.5.2.Assumption Made Prior to Project Execution.....	6
1.5.3.Limitation.....	6
1.6.Definition of Terms.....	7
1.7.Organization of Report.....	8
CHAPTER 2: LITERATURE REVIEW.....	9
2.1.Introduction.....	9
2.2.Brief Review on Artificial Intelligence Literature.....	9
2.3.Overall Literature Review.....	11
2.3.1.Neural Network (NN).....	12
2.3.2.Congkak Game.....	13
2.3.3.Min-Max Algorithm with Alpha-Beta function (MM).....	16
2.4.Extra Definition and Term.....	17
2.5.Summary.....	18
CHAPTER 3: PROJECT METHODOLOGY.....	19
3.1.Introduction.....	19
3.2.Project Methodologies.....	20
3.2.1.Project Identification & Initiation.....	21
a) Identify the Problems.....	21

b) Perform Literature Review.....	22
c) Create a Valid Research Question.....	22
3.2.2.System Design & Development.....	23
a) Design a System that Answers Research Question.....	23
i. Neural Network (NN) Sub-system: Intro.....	25
ii. Neural Network (NN) Sub-system: Classes.....	25
iii. Neural Network (NN) Sub-system: Design Intro.....	27
iv. Neural Network (NN) Sub-system: Design.....	27
v. Neural Network (NN) Sub-system: Extra Term and Definitions.	37
3.2.3.Data Collection & Analysis.....	38
a) Identify the Data to be Collected.....	38
b) Perform the Data Collection.....	38
c) Processing the Data and Analyze the Data.....	39
d) Interpret the Data and Answer Research Question.....	40
3.2.4.Project Documentation: Prepare the Report.....	40
3.3.The Limitation of the Project's Methodology.....	42
3.4.Summary.....	42
CHAPTER 4: RESULT.....	43
4.1.Introduction.....	43
4.2.Brief Statement of the Result.....	43
4.3.Result and Processed Data.....	44
4.3.1.The GUI.....	44
4.3.2.Artificial Intelligent Agents.....	49
4.4.Summary.....	58
CHAPTER 5: DISCUSSION OF THE RESULT.....	59
5.1.Issues.....	59
5.1.1.The GUI's Simultaneous Move.....	59
5.1.2.Possible Aesthetic Improvement.....	60
5.1.3.The Move Pause.....	60
5.1.4.Congkak and Numbers.....	61
5.1.5.More Functionality for GUI	61
5.1.6.Real Congkak and Simulated Congkak.....	62
5.1.7.Issue: First Move is not Hidden to Artificial Agent.	62
5.1.8.Issues with the Training Graph.....	62

5.1.9. Noise During Neural-Network Training	63
5.1.10. Possible Reasons for Neural-Network Poor Performance.....	64
5.2.Summary.....	65
CHAPTER 6: CONCLUSION AND RECOMMENDATION.....	66
6.1.Conclusions.....	66
6.2.Recommendation/ Future work.....	68
REFERENCES.....	70
APPENDIX	75

LIST OF TABLE

Table 3.1: Project's Methodology.....	20
Table 4.1: Win-loose count for all agent.....	50
Table 4.2: The speed of NN agent algorithm.	57
Table 4.3: The speed of NMM agent algorithm.....	57
Table 4.4: The speed of RandomMove agent algorithm.....	57
Table 4.5: The speed of MM agent algorithm.....	58

LIST OF FIGURE

Figure 3.1: Congkak board representation for Neural Network (NN).....	29
Figure 3.2: Move state representation for NN input.....	29
Figure 3.3: A typical NN training configuration.....	31
Figure 3.4: Method for selecting move from 2 parallel NN output.....	32
Figure 4.1: GUI waiting for input.	45
Figure 4.2: GUI waiting for input.	45
Figure 4.3: GUI Simultaneous move.....	46
Figure 4.4: Move pause during simultaneous move.....	47
Figure 4.5: Sorting simulation.	48
Figure 4.6: The endgame dialogue.....	48
Figure 4.7: Helper function.	48
Figure 4.8: The Win over TrainingCount graph for “Defensive move” and “Custom strategy” (4000 trainingCount).....	51
Figure 4.9: The Win over TrainingCount graph for “Defensive move” and “Distance from Endgame” (5000 trainingCount).....	52
Figure 4.10: The Win over TrainingCount graph for “Winning Move” and “Distance from Endgame” (11000 trainingCount).....	54
Figure 4.11: Maximum training time for “Defensive Move” and “Custom Strategy” evaluation function.....	55
Figure 4.12: Maximum training time for “Winning Move” and “Distance from Endgame” evaluation function.....	56

Chapter 1

1. INTRODUCTION

This project is focussed on implementing Artificial Intelligence (AI) technique in Congkak game playing. An AI agent was created as a player that could be configured to play with a human or with itself. The agent used Neural Network algorithm (NN), Min-Max algorithm with Alpha-Beta function (MM), and Random-moves-generator to play the game.

AI is an exciting field of research. The goal of AI field is to develop a system that can solve real world problems: such as Chess game, predicting stock market and facial recognition. Most recent and exciting development is an AI agent named Watson developed by IBM; it can answer question posed in natural language and has won a game in an American quiz show called “Jeopardy” (IBM (2011)). Another AI field is visual recognition; which also has become ubiquitous nowadays in form of facial recognition software installed on our laptop, and other exciting development is in computer gaming; where an AI agent named Milo can recognize player's emotion and interact with the player (Gibson. E (2009)).

The project will use AI on a small scale. Several AI technique was used on the game Congkak; Congkak has simpler rule and simpler mechanics than

The contents of
the thesis is for
internal user
only

REFERENCES

- 5up3rJ (aka SuperJ) (Jan 4, 2006). Java Discussion Thread: Time Delay [Msg 3]. Message posted to <http://www.daniweb.com/software-development/java/threads/37443>
- Ahlschwede, J. (2000). *Using Genetic Programming to Play Mancala*. Retrieved on 13.3.2011 from <http://www.corngolem.com/john/gp/project.doc>
- Alex de Voogt (2001). Mancala: A Game That Count. *Expedition Philadelphia*, Issue 2001. Penn Museum. Retrieved on 10.3.2011 from <http://www.penn.museum/documents/publications/expedition/PDFs/43-1/Mancala.pdf>
- Alifia; Frilla Ariani; Tania Krisanty, (2006) Pencarian Solusi Optimal Pemilihan Lubang pada Permainan Congklak dengan Algoritma Greedy dan Program Dinamis. In *Majalah Jurusan Teknik Informatika*. Institut Teknologi Bandung, Bandung
- Blais, A., Mertz, D. (2001) *An introduction to neural networks*. DeveloperWorks, IBM. Retrieved on 13.3.2011 from <http://www.ibm.com/developerworks/library/l-neural/>
- Bylander, T. (2008). CS 3793 Lab 3: Mancala min-max assignment. CS 3793: Introduction to Artificial Intelligence Course. University of Texas at San Antonio. Retrieved on 13.3.2011 from <http://www.cs.utsa.edu/~bylander/cs3793/lab3.pdf>
- Bylander, T. (2007). CS 5233: Best-Case Analysis of Alpha-Beta Pruning. CS 5233: *Artificial Intelligence Course*. University of Texas at San Antonio. Retrieved on 20.3.2011 from <http://www.cs.utsa.edu/~bylander/cs5233/a-b-analysis.pdf>
- Brunei Darussalam Information Department (2008). *Congkak: A Game of Wits*. Brunei Darussalam Newsletter. Retrieved on 3.4.2011 from http://www.information.gov.bn/bdnewsletter/index.php?option=com_content&task=view&id=95&Itemid=44
- Carter-Greaves, L. E. (2009). *Time Series Prediction With Feed-Forward Neural Networks*. Retrieved on 13.3.2011 from <http://neuroph.sourceforge.net/TimeSeriesPredictionTutorial.html>
- Cofer, A. (2003), *Mancala In Java: Experiment in Artificial Intelligence and Game Playing*. Departmental Honor Thesis. University of Tennessee at Chattanooga.
- Congkak (2011). In *Wikipedia*. Retrieved on 13.3.2011 from <http://en.wikipedia.org/wiki/Congkak>

- CPU Time (2011). In *Wikipedia*. Retrieved on 25.5.2011 from http://en.wikipedia.org/wiki/CPU_time
- Davis, J. E., Kendall, G. (2002). An Investigation, using Co-Evolution: to Evolve an Awari Player. In *Proceedings of 2002 Congress on Evolutionary Computation (CEC2002)* (pp. 1409-1413). IEEE Press
- Donkers, J., Alex de Voogt; Uiterwijk, J. (2000). Human versus Machine Problem-Solving: Winning Openings in Dakon. In M. Baud; R.A.H.D. Effert; M. Forrer; F. Hüsken; K. Jongeling; H. Maier; P. Silva; B. Walraven (Eds.), *Board Games Studies* (pp. 79-91), Vol. 3. CNWS Publication, Universiteit Leiden.
- Deitel, P. J.; Deitel, H. M. (2006). *Java: How To Program* (7th Edition). Pearson Education Inc, New Jersey.
- Dennis, A.; Wixom, B.H.; Roth, R. M. (2010). *System Analysis and Design* (4th Edition). John Wiley & Sons Inc, New Jersey.
- Gibson. E (2009). E3: Molyneux and Milo Xbox360 Interview. Eurogamer.net. Retrieved on 19.3.2011 from <http://www.eurogamer.net/articles/e3-project-natals-molyneux-and-milo-interview>
- Eck, D. J. (2006). Introduction to Programming Using Java, Fifth Edition: *Section 7.3: Dynamic Array and ArrayList*. Department of Mathematics and Computer Science, Hobart and William Smith Colleges. Published online under Creative Common Attribution licence. Retrieved on 3.4.2011 from <http://math.hws.edu/javanotes/c7/s3.html>
- Ghory, I. (2004). Reinforcement Learning In Board Game. *Technical Report CSTR04-004*. Retrieved on 20.3.2011 from <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.4.7712>
- H. Jaap van den Herik, Uiterwijk, J, Jack van Rijswijck, (2001) Games solved: Now and in the future. *Artificial Intelligence* (pp. 277–311), Issue 134 (2002). Elsevier Science B.V.
- IBM (2011). *IBM – What is Watson?*. In IBM Watson. Retrieved on 19.3.2011 from <http://www-03.ibm.com/innovation/us/watson/what-is-watson/index.html>
- Irving, G., Donkers, J., Uiterwijk, J. (2000), Solving Kalah. *ICGA Journal*. California Institute of Technology, Pasadena, and Universiteit Maastricht, Maastricht.
- Mohammed Daoud; Nawwaf Kharmah; Ali Haidar; Julius Popoola. (nd). Ayo, the Awari Player: How Better Representation Trumps Deeper Search. *Evolutionary Computation, 2004. CEC2004. Congress on* (pp. 1001-1006), Vol.1. Concordia University, Montreal.

- Newborn, Monty; Newborn, Monroe (2003). *Deep Blue: an Artificial Intelligence milestone (1st Eds.)* (pp. 228-229). Springer-Verlag Inc, New York.
- Pickhard, A. (2007) Final year project of evolving AI for Mancala. Retrieved on 13.3.2011 from <http://amplizine.com/wp-content/uploads/2008/10/mancala.doc>
- Pinto, P. (2002). Introducing the Min-Max Algorithm. Retrieved on 13.3.2011 from http://www.progtools.org/games/tutorials/ai_contest/minmax_contest.pdf
- Perpustakaan Negara Malaysia (2000). *Congkak Warisan Negara*. Retrieved on 13.3.2011 from <http://malaysiana.pnm.my/03/0305conggkak.htm>
- Romein, J. W.; Bal, H. E. (2003). Solving the Game of Awari using Parallel Retrograde Analysis. *IEEE Computer* (pp. 2003), Vol 36.
- Smith, P. (2000) Mancala Games. *MindZine*. Mind Sport Worldwide, msoworld.com Retrieved on 13.3.2011 from <http://www.msoworld.com/mindzine/news/classic/mancala.html>
- Shah Ali Reza Yaacob (2006). Knowledge-Based System Development For The Game Congkak: Bachelor Thesis. *UNIVERSITI TEKNOLOGI MARA Digital Repository*. Universiti Teknologi MARA. Retrieved on 13.3.2011 from <http://eprints.ptar.uitm.edu.my/990/>
- Steinhauer, V. (2009). *Stock market prediction using neural networks: An example for time-series prediction*. Retrieved on 13.3.2011 from <http://neuroph.sourceforge.net/tutorials/StockMarketPredictionTutorial.html>
- Kronenburg, T. (2008). *Towards a Quasi-Endgame-Based Bao Solver*; Master Thesis. Universiteit Maastricht, Maastricht. Retrieved on 13.3.2011 from <http://www.manqala.org/docs/thesistomkronenburg.pdf>
- Kothari, C.R (2004). *Research Methodology: Methods and Techniques (2nd Edition)* (pp. 10-21). New Age International Publisher Ltd., New Delhi.
- Oracle (2011) *Understanding Instance and Class Members*. The Java Tutorials. Oracle, oracle.com. Published online under Java Tutorial SE Tutorial Copyright and Licence. Retrieved on 3.4.2011 from <http://download.oracle.com/javase/tutorial/java/javaOO/classvars.html>
- Oracle (2011). *Java SDK download (JDK 6 Update 24)* [Software]. Oracle, Sun Microsystems Inc., Retrieved on 13.3.2011 from <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
- Wee-Chong, O.; Yew-Jin, L.(2003). An Investigation on Piece Differential Information in Co-Evolution on Games Using Kalah. *Evolutionary Computation, 2003. CEC '03. The 2003 Congress on* (pp. 1632 – 1638), Vol.3.

- Sevarac, Z. ; Goloskokovic, I. ; Tait, J. ; Morgan, A. ; Carter-Greaves, Laura (2008). *Neuroph main page: resource for Neuroph*. Retrieved on 13.3.2011 from <http://neuroph.sourceforge.net/index.html>
- JuMpErFLY (April 24th 2004). Java: JButton and actionPerformed [Msg 7]. Message posted to <http://forums.bit-tech.net/showthread.php?t=56894>
- Nur Liyana Zainal Abidin. (2011). “*Dunia Matematik: Mancala*”, Komuniti Dunia Matematik, online magazine issue: April 2011, volume: 5th. Universiti Teknologi Malaysia, Johor Bahru. Retrieved on 25.5.2011 from <http://mathed.utm.my/duniamatematik/index.php/permainan>
- Bonny, T. (2010). *Congkak*. National Library Board, Singapore. SingaporeInfopedia. Retrieved on 3.4.2011 from http://infopedia.nl.sg/articles/SIP_1733_2010-11-26.html
- taheretaheri (2010). *Benchmarking and Comparing Encog, Neuroph and JOONE Neural Networks*. Retrieved on 6.6.2011 from <http://www.codeproject.com/KB/recipes/benchmark-neuroph-encog.aspx>
- Tesauro, G., Sejnowski, T. J. (1989). A Parallel Network that Learn to Play Backgammon. *Artificial Intelligence* (pp. 357–390), Issue 39 (2002). Elsevier Science Publisher, B.V.
- Tesauro, G. (1992). Practical Issues in Temporal Difference Learning. *Machine Learning* (pp. 257-277), Issue 8, (1992). Kluwer Academic Publishers, Boston.
- Tesauro, G. (1995). Temporal Difference Learning and TD-Gammon. *Communications of the ACM* (No. 3), Issue March 1995 Volume 38. Association for Computing Machinery.
- Tesauro, G. (2002). Programming backgammon using self-teaching neural nets. *Artificial Intelligence* (pp. 181–199), Issue 134 (2002). Elsevier Science Publisher, B.V..
- Temporal difference learning (2011). In *Wikipedia*. Wikipedia Inc. Retrieved on 25.5.2011 from http://en.wikipedia.org/wiki/Temporal_difference_learning
- Goldberg, C. (2005). *StopWatch.java: Java Timer Class*. Goldb.org. Retrieved on 3.4.2011 from <http://www.goldb.org/stopwatchjava.html>
- Yaakub Rashid (1981). *Revival of Congkak*. The Strait Times, National Library Board, Singapore. NewspaperSingapore. Retrieved on 3.4.2011 from <http://newspapers.nl.sg/Digitised/Article/straittimes19811006.2.130.6.1.aspx>
- How To Play Congkak* (2010). In WikiHow: The How To Manual that You can Edit. Retrieved on 3.4.2011 from <http://www.wikihow.com/Play-Congkak>
- Samuel, A. L. (1967). *Some Studies in Machine Learning Using the Game of*

Checkers. II – recent progress. *IBM Journal of research and development*, 1967.

Sutton, R. S., Barto, A. G. (2005). *Reinforcement Learning: An Introduction*. The MIT Press, Cambridge. Retrieved on 22.3.2011 from <http://neuro.bstu.by/ai/RL-3.pdf>

StatSoft, Inc. (2011). Electronic Statistics Textbook. Tulsa, OK: StatSoft. WEB: <http://www.statsoft.com/textbook/neural-networks/>.

Lim, Y.J. (2007). *On Forward Pruning In Game Tree Search*. National University of Singapore, Singapore. Retrieved on 21.3.2011 from <http://www.yewjin.com/storage/papers/PhDThesisLimYewJin.pdf>

Lawrence, S., Giles, C. L., Tsoi, A.C. (1997). Lessons in Neural Network Training: Overfitting May be Harder than Expected. *Proceedings of the Fourteenth National Conference on Artificial Intelligence* (pp. 540–545), Issue AAAI-97. AAAI Press, California.

Nyugen, D., Widrow, B. (1989). The Truck Backer Upper: An Example of Self Learning in Neural Network. In *International Joint Conference on Neural Network (IJCNN)*.