

Computer Games Development CW208

Technical Design Document

Year III

|  |
| --- |
| [ZhenzeZhao]  [C00198551] |
| [03/05/2020] |

[Declaration form to be attached]

# Faculty of Computing and Networking Science

# Open-Book and Remote Assessment Cover Page

# 

# Student Name:Zhenze Zhao

# 

# Student Number: C00198551

# 

# Lecturer Name: Lei Shi

# 

# Module:Final Year Project

# 

# Stage/Year:4th

# 

# Date:03/05/20

# 

# Declaration

# 

# This examination/assessment will be submitted using Turnitin as the online submission tool. By submitting my examination/assessment to Turnitin, I am declaring that this examination/assessment is my own work. I understand that I may be required to orally defend any of my answers, to the lecturer, at a given time after the examination/assessment has been completed, as outlined in the student regulations.

Contents

[Game Architecture](#_heading=h.gjdgxs) 3

[Features: 3](#_heading=h.1fob9te)

Board Generation

Game Board

Player

Start Menu

AlphaBeta Search

Monte Carlo Tree Search

Search Node

[CRC Cards](#_heading=h.3znysh7) 6

Class Name : Player 6

Class Name : AI player 6

Class Name : UI 7

Class Name : Board 7

Class Name : Game Manager 7

Class Name : Piece 8

Class Name : MCTS Node 8

[References](#_heading=h.tyjcwt) 8

# **Game Architecture****Features:**

## ***Feature:*** Board Generation

Tasks:

1. The board size can be changed by controlling the size slider at the at menu.
2. Change the size value and the board script attaches that value, then the board script creates a new board on the screen.

## Feature: Game Board

Tasks:

1. The board will generate a list of nodes for available points, and the list is sent to the AI player.
2. The board will renew the list of available nodes after move.
3. If the available nodes reduce to 0, the game ends.
4. After the game ends, the system will count the pieces on the board, and compare the answer for each player.

## Feature: Game Manager

Tasks:

1. The program checks the pieces on the board, if those pieces are surrounded by opponent’s pieces then remove it from the board.
2. If the pieces have ally pieces around, those pieces will group up. If the grouped the pieces have been surrounded, all the pieces in that group will be taken away.
3. The game system checks if the player moves not allowed, the pieces will be removed.
4. The turn is shifted after one of a player places a piece on the board.

## Feature: Player

Tasks:

1. The player checks the user setting. Is this player as an AI player? Is this AI run with Monte Carlo Tree Search?
2. The AI player generates the search node for different functions and copies the current game state.

## Feature: Start Menu

Tasks:

1. Set up the board size.
2. Set up the AI depth.
3. Set up player control or AI control.

## Feature: AlphaBeta Search

Tasks:

1. Get a copy of current board state and create a new empty node for AlphaBeta
2. System checks the current turn, then simulates the game in a copy of the current board.
3. When the AI simulates the game, It will score all possibilities in this game.
4. The node score is based on two parts, enemy movement score and player score.
5. After simulation, the function gets the most valuable points for the next move. And place the piece on that point.

## Feature: Monte Carlo Tree Search

Tasks:

1. Get a copy of current board state and create a new empty node for MCTS
2. The function creates a root node, and the run with tree policy.
3. In the tree policy function, find the end node(the node isn't the end of the game) in the root node and start sending the end node to default policy.
4. After node simulating to the end of the game, back up the score and time visited for every node of this search tree.
5. When the simulation ends, get the UCB value for the next node and use the most valuable as the player's next move.

## Feature: Search Node

Tasks:

1. The AlphaBate node includes the position of current node and score for this node.
2. The MCTS node includes the parents node, child node, available moves, point, score and times visited.

# **CRC Cards**

|  |  |
| --- | --- |
| ***Class Name : Player*** | |
| Subclasses : | |
| Superclasses : | |
| Responsibilities | Collaborators |
| Play piece with mouse click |  |

# 

|  |  |
| --- | --- |
| ***Class Name :*** AI player | |
| Subclasses :Node, MCTSnode | |
| Superclasses : | |
| Responsibilities | Collaborators |
| Generate the node and copy of current board | AlphaBeta node |
| Simulate game with selected function | MCTS node |
| Compare the node’s score and find best node | board |
| Play the best node on the board |  |

|  |  |
| --- | --- |
| Class Name : UI | |
| Subclasses : | |
| Superclasses : | |
| Responsibilities | Collaborators |
| Change the board size | BoardPrefab |
| Set up player state and AI thinking depth |  |

|  |  |
| --- | --- |
| Class Name : Board | |
| Subclasses :Player | |
| Superclasses : | |
| Responsibilities | Collaborators |
| Hold the current game states | Player |
| Check the piece on board and remove the piece when it get surround by enemy piece |  |
| Generate available point list |  |
| Count the piece on the board for each player |  |

|  |  |
| --- | --- |
| Class Name : Game Manager | |
| Subclasses :Player, Board | |
| Superclasses : | |
| Responsibilities | Collaborators |
| Turn change after one player move | Player |
| Set up Player style(user control or AI control) | Board |
| Generate available point list |  |

# 

|  |  |
| --- | --- |
| Class Name : Piece | |
| Subclasses : | |
| Superclasses : | |
| Responsibilities | Collaborators |
| Record the position on the board |  |
| Record the color for different player |  |

|  |  |
| --- | --- |
| Class Name : MCTS Node | |
| Subclasses : | |
| Superclasses : | |
| Responsibilities | Collaborators |
| Create a copy board from current state, but do not display it |  |
| Keep the node position, node score, time visited and parents node |  |
| Expand node and create new leaf node |  |
| Calculate the node score |  |

# 

# **References**

# 

|  |  |  |
| --- | --- | --- |
| **Referenced Publication** | **Citation** | **Reference** |
| Video Course | (Author Year)  Examples  <https://www.youtube.com/watch?v=niIaKaWIRX0&t=1156s> | The Monte Carlo Simulation |
| Web-site | <https://www.researchgate.net/publication/292074166_Mastering_the_game_of_Go_with_deep_neural_networks_and_tree_search>  Author(s) Marc Lanctot. Christopher Maddison | Author(s) Marc Lanctot. Christopher Maddison  Mastering the game of Go with Deep Neural Networks and Tree Search, Nature. |
| Web-site | <http://www.incompleteideas.net/609%20dropbox/other%20readings%20and%20resources/MCTS-survey.pdf>  Cameron Browne  VOL. 4, NO. 1, MARCH 2012 | Survey of Monte Carlo Tree Search Methods, IEEE Transactions on Computational Intelligence and AI in games |
| Book | <https://link.springer.com/chapter/10.1007/11871842_29>  Part of the [Lecture Notes in Computer Science](https://link.springer.com/bookseries/558) book series (LNCS, volume 4212) | Authors:Levente Kocsis,Csaba Szepesvári Bandit Based Monte-Carlo Planning |
| Web-Site | <https://arxiv.org/abs/1611.00625>  (Submitted on 26 Aug 2019, last revised 10 Oct 2019 (this version, v4)) | Author(s) -[Marc Lanctot](https://arxiv.org/search/cs?searchtype=author&query=Lanctot%2C+M), [Edward Lockhart](https://arxiv.org/search/cs?searchtype=author&query=Lockhart%2C+E), [Jean-Baptiste Lespiau](https://arxiv.org/search/cs?searchtype=author&query=Lespiau%2C+J), [Vinicius Zambaldi](https://arxiv.org/search/cs?searchtype=author&query=Zambaldi%2C+V), [Satyaki Upadhyay](https://arxiv.org/search/cs?searchtype=author&query=Upadhyay%2C+S), [Julien Pérolat](https://arxiv.org/search/cs?searchtype=author&query=P%C3%A9rolat%2C+J), [Sriram Srinivasan](https://arxiv.org/search/cs?searchtype=author&query=Srinivasan%2C+S), [Finbarr Timbers](https://arxiv.org/search/cs?searchtype=author&query=Timbers%2C+F), [Karl Tuyls](https://arxiv.org/search/cs?searchtype=author&query=Tuyls%2C+K), [Shayegan Omidshafiei](https://arxiv.org/search/cs?searchtype=author&query=Omidshafiei%2C+S), [Daniel Hennes](https://arxiv.org/search/cs?searchtype=author&query=Hennes%2C+D), [Dustin Morrill](https://arxiv.org/search/cs?searchtype=author&query=Morrill%2C+D), [Paul Muller](https://arxiv.org/search/cs?searchtype=author&query=Muller%2C+P), [Timo Ewalds](https://arxiv.org/search/cs?searchtype=author&query=Ewalds%2C+T), [Ryan Faulkner](https://arxiv.org/search/cs?searchtype=author&query=Faulkner%2C+R), [János Kramár](https://arxiv.org/search/cs?searchtype=author&query=Kram%C3%A1r%2C+J), [Bart De Vylder](https://arxiv.org/search/cs?searchtype=author&query=De+Vylder%2C+B), [Brennan Saeta](https://arxiv.org/search/cs?searchtype=author&query=Saeta%2C+B), [James Bradbury](https://arxiv.org/search/cs?searchtype=author&query=Bradbury%2C+J), [David Ding](https://arxiv.org/search/cs?searchtype=author&query=Ding%2C+D), [Sebastian Borgeaud](https://arxiv.org/search/cs?searchtype=author&query=Borgeaud%2C+S), [Matthew Lai](https://arxiv.org/search/cs?searchtype=author&query=Lai%2C+M), [Julian Schrittwieser](https://arxiv.org/search/cs?searchtype=author&query=Schrittwieser%2C+J), [Thomas Anthony](https://arxiv.org/search/cs?searchtype=author&query=Anthony%2C+T), [Edward Hughes](https://arxiv.org/search/cs?searchtype=author&query=Hughes%2C+E), [Ivo Danihelka](https://arxiv.org/search/cs?searchtype=author&query=Danihelka%2C+I), [Jonah Ryan-Davis](https://arxiv.org/search/cs?searchtype=author&query=Ryan-Davis%2C+J) TorchCraft: a Library for Machine Learning Research on Real-Time Strategy Games |