Posudek bakalářské práce

Matematicko-fyzikální fakulta Univerzity Karlovy

Autor práce	Michal Počatko		
Název práce	AI for the Board Game Azul		
Rok odevzdání	2020		
Studijní program	Informatika	**Studijní obor**	Programování a softwarové systémy

Autor posudku	Adam Dingle	**Role**	Vedoucí
Pracoviště	KSVI		

Prosím vyplňte hodnocení křížkem u každého kritéria. Hodnocení *OK* označuje práci, která kritérium vhodným způsobem splňuje. Hodnocení *lepší* a *horší* označují splnění nad a pod rámec obvyklý pro bakalářskou práci, hodnocení *nevyhovuje* označuje práci, která by neměla být obhájena. Hodnocení v případě potřeby doplňte komentářem. Komentář prosím doplňte všude, kde je hodnocení jiné než *OK*.

K celé práci	lepší	OK	horší	nevyhovuje
Obtížnost zadání	☐	☒	☐	☐
Splnění zadání	☐	☒	☐	☐
Rozsah práce *... textová i implementační část, zohlednění náročnosti*	☐	☐	☐	☒
Komentář	The author's C++ implementation of the AI agents is reasonable. However the thesis text seems like an incomplete first draft, and therefore the thesis work at a whole is not satisfactory.			

Textová část práce	lepší	OK	horší	nevyhovuje
Formální úprava *... jazyková úroveň, typografická úroveň, citace*	☐	☐	☐	☒
Struktura textu *... kontext, cíle, analýza, návrh, vyhodnocení, úroveň detailu*	☐	☐	☐	☒
Analýza	☐	☐	☐	☒
Vývojová dokumentace	☐	☐	☒	☐
Uživatelská dokumentace	☐	☐	☐	☒

Komentář The thesis text seems like an incomplete first draft, not a completed bachelor's thesis. In various places the text is hard to understand. The description of the minimax algorithm is vague and essentially incomplete. The pseudocode for the static evaluation function for minimax is unclear. The description of the important MCTS (= Monte Carlo Tree Search) algorithm has various problems and is difficult to follow. The text refers to various external programs and algorithms without providing sufficient references to them. The description of MCTS is for a deterministic game, and the author has not explained how he applied it to the stochastic game of Azul. The author has not explained how MCTS playouts work in his implementation; the section that should describe that (4.2.4 Simulation) ends with an incomplete sentence and apparently the author did not finish writing it. The author has not discussed the challenges of applying MCTS to a game with a high branching factor such as Azul, or whether he attempted to use techniques proposed by the authors he cites (such as progressive bias) to overcome those challenges. The experimental results presented in the thesis generally have too few trials to provide precise results, and the author has not provided statistical confidence intervals that would quantify this imprecision. Some pairs of AIs (e.g. MCTS versus a greedy algorithm) were not tested. The author has not provided a script that a reviewer can run to repeat his experimental results. The thesis text does not document the options that can be passed to the command-line interface. The text does not clearly specify which operating system versions the program will run on, or the versions of the build dependencies that are required. The build instructions are sparse in some places.

The paragraph above summarizes the biggest problems in the thesis text. I've attached several pages with more detailed commentary on each thesis section.

Implementační část práce	lepší	OK	horší	nevyhovuje
Kvalita návrhu *... architektura, struktury a algoritmy, použité technologie*	☐	☒	☐	☐
Kvalita zpracování *... jmenné konvence, formátování, komentáře, testování*	☐	☒	☐	☐
Stabilita implementace	☐	☒	☐	☐

Komentář The game implementation has a nice user interface. The implementations of the various AI agents seem reasonable. However, the MCTS agent sometimes makes obviously poor moves near the beginning of the game, and the author has not explained this fact or provided a remedy.

Celkové hodnocení Neprospěl(a)
Práci navrhuji na zvláštní ocenění Ne

Datum 4.9.2020 **Podpis** 

== 0.5 Intended audience ==

- Your first sentence here ("of this thesis...") is a continuation of the section title. In formal writing, each sentence should be a complete sentence on its own. So this sentence should begin "The intended audience of this thesis..."

- "c++" should be "C++".

== 0.6 Thesis structure ==

- "of four main chapters": you should add a colon after "chapters".

== 1 Game rules ==

- The description of the game rules is mostly reasonable. However, it uses the terms "bag" and "lid" without defining what these mean. It's slightly confusing that you use the term "row" to mean a *vertical* column of tiles, since usually a "row" means a horizontal set.

== 2.2 Complexity ==

- In "At the beginning of the game There are 5 factories", the word "There" should not be capitalized.

- In "Through Testing", the word "Testing" should not be capitalized.

- You discuss the number of possible moves at each step, but does not ever say "branching factor", which is the standard term for this concept.

- You claim that the number of possible moves "decreases linearly", but the number of possible moves is not in fact a linear function of the move number. Perhaps you meant "decreases monotonically", but even that is not true, since the table shows that in each round the number may sometimes increase.

== 3 Implementation ==

- In this section you mention various technologies (Godot, GDnative, Gimp) and concepts (MVC design pattern, Command design pattern), but do not provide references or hyperlinks for these.

- What operating system(s) does your game run on? Which version(s) of those operating systems will work?

- Your thesis text does not indicate how to build your program. At the very least you should mention that build instructions are available in the file source/README.md.

- In README.md you should indicate the required versions of the various build requirements (Scons, Visual Studio, Godot, NSIS). Also, it would be best to use the full name "Microsoft Visual Studio". It would also be good to provide links to the sites where these requirements can be downloaded.

- In README.md, step 1 "Compile godot-cpp submodule" is a bit terse. You could mention that the user can do so by following the steps in source/godot-cpp/README.md. However, that file itself is long and many of the steps in it are not relevant, so I think it would be most helpful if you simply showed the required build commands:

```
$ cd godot-cpp
$ scons platform=<your platform> generate_bindings=yes
```

- Since your source tree contains a build file for Linux (meson.build), please also include build instructions for Linux in README.md. (There is no installer for Linux, which is fine.)

== 3.1 Front end ==

- You wrote "The front end was created in the Godot game engine". Where in your source tree can this front end be found?

- Does the front end consist of source code in some programming language?

== 3.1.1 Graphics ==

- You wrote "The graphics for the GUI are a simple set of colored shapes created in Gimp.". Where can these be found in your source tree?

- The sentence "For game pieces combined..." is incomplete: it has no verb.

== 3.1.2 GUI functionality ==

- In the phrase "combined with a library cpp bindings", it's not clear whether "cpp bindings" is the *name* of the library or a *description* of it. I would probably write "a library of C++ bindings".

- You wrote "Godot's public Github site". You should provide a link to this site. Also, the proper capitalization is "GitHub".

== 3.2 ==

- You wrote "the language of choice is C++ for its speed...". This statement is a bit strong, since I think that other programmers might reasonably choose a different language for various reasons. I would just say "we chose C++ for its speed...".

== 3.2.4 View ==

- You wrote "The CUI can be given different options from the command line". What options are available, specifically?

== 4.1 Minimax ==

- Your description of the minimax algorithm is vague and essentially incomplete. It is only three sentences long. The first two sentences are broad statements that apply to any game-playing algorithm at all. The third sentence mentions a heuristic function, but says nothing about how the minimax algorithm uses this function to select a move to play.

If you don't want to actually describe the minimax algorithm, you should provide a reference to a book or article that does so.

== 4.1.2 Static evaluation function ==

- This section should say where the actual evaluation function can be found in the program's source code.

- The game has two players. For any game state, does the pseudocode here run twice, once for each player? If so, how are the output scores combined to form the heuristic function's output?

- The pseudocode is not entirely clear. Presumably pl.count is the number of tiles on the pattern line and pl.size is the number of tiles that the pattern line can hold, but you should say that explicitly. What does the wall_tile_score() function return, exactly? What does the floor_penalty() function return? The text says "An ownership of a starter tile is also accounted for as half of a point", yet that fact is not obviously present in the pseudocode.

- In general, it may not be possible to know whether the current round will be the last. So how does the last_round() function work?

- Really this section should explain *why* this particular evaluation function was chosen. Were other possible evaluation functions tested? If so, how much better was this one?

== 4.1.3 Alpha-beta pruning ==

Your two-sentence description of alpha-beta pruning is a vague summary. If you don't intend to describe the algorithm in detail, you should at least provide a reference to a book or article that describes it.

== 4.1.4 Move ordering ==

- This section should say where the actual move ordering function can be found in the program's source code.

- The pseudocode here is unclear in some ways. The lines that calculate "overflow" and "fraction" involve the pattern line number "pl", but it's not clear whether pattern lines are numbered from 0 to 4 or from 1 to 5. What do the floor_score() and wall_tile_score() functions return, exactly?

- This section should also explain *why* this particular move ordering heuristic was chosen. Were other orderings tested? If so, how much better was this one?

== 4.1.6 Results ==

- n = 50 is not a very large number for generating win percentages such as these, since at n = 50 the percentages might vary quite a bit from run to run. Suppose that I flip 50 coins. The expected number of heads will be 25. The standard deviation of the number of heads will be sqrt(n * p * (1 - p)) = sqrt(50 * .5 * .5) = sqrt(12.5) = 3.54. There is about a 5% chance that the value will be at least two standard deviations from the mean. So if we compare two agents that actually perform identically, then with n = 50 there is about a 5% change that we will report that once agent has a win rate of at least (25 + 7) / 50 = 32 / 50 = 64%. This shows that the win rates here are certainly not precise to within a single digit.

  Really it would be best here to assume some confidence level such as alpha = 0.05 or 0.02, and then report a *confidence interval* for each win rate. I know that you have code to do that since we discussed this early in the project, and so I think it would not be difficult to report that data here. The confidence interval will become narrower as N increases, of course.

- The reader of your thesis may wish to repeat your experiments to verify the numbers in your text. So you should provide one or more scripts that the user can run to generate all of your experimental results, both in this section and other results sections.

== 4.2.1 Algorithm description ==

- The usual capitalization is "Monte Carlo tree search".

- You say that MCTS is a "best-first search method", but you don't explain what "best-first search" means or provide a reference to a work that explains this term.

- Your statement that "MCTS has substantially advanced the state of the art in board games such as Go, Amazons, Hex, Chinese Checkers, Kriegspiel, and Lines of Action" needs a reference, since you have certainly not demonstrated this fact yourself.

- Before the pseudocode the text says "MCTS pseudo code". This is a sentence fragment, which is not acceptable in formal writing. Instead, this should be a sentence such as "Here is pseudocode for the MCTS algorithm".

- The pseudocode here has all lines flush with the left column. Instead, it should be indented nicely.

- In this section you are discussing the algorithm in general, not your agent that plays Azul. And so "MonteCarloAI" should be "Monte Carlo tree search" here.

== 4.2.2 Selection ==

- You mention a "tree", but don't specify what each tree node represents or contains. Specifically, each node in figure 4.1 has two numbers, e.g. 11 / 21. It's not clear what these numbers mean.

- You mention "exploitation" and "exploration", but don't say what these concepts mean.

- In the formula here, "b ∈ argmax" should be "b = argmax".

- You have not explained what "I" means in the formula here.

- In the text "vi is the value of the node i, mi is the visit count of i, and mp is the visit
count of p", the variables vi, mi and mp should be subscripted as in the preceding

formula.

- How is the value v_i calculated for each node?

- You wrote "The formula is applied recursively until an unknown position is reached." This is vague. What does it mean to apply this formula recursively? What is the definition of an "unknown position"?

== 4.2.3 Expansion ==

- You write "an unknown node is selected randomly from all the possible moves". This is vague - how can a node be selected from a set of moves?

== 4.2.4 Simulation ==

- The section ends with an unfinished sentence ("We achieve this by"). The missing information here is important: just how do playouts work in your implementation?

== 4.2.5 Backpropagation ==

- You wrote "If a game is won, the result of a player j is scored as a rt,j 1, in the case of a loss
as rt,j 0, and a draw as rt,j 0.5." The notation "rt,j" is cryptic - what does this mean?

== 4.2.6 Final Move Selection ==

- The first few lines of text here look strange since they wrap too early. Apparently the lower part of Figure 4.4 from the previous page has extended onto this page and caused this wrapping.

- Your two sentences beginning "In practice, the difference in performance between these strategies..." are taken literally from Winands, "Monte-Carlo Tree Search in Board Games". They do not appear in the Chaslot article you have linked to in this section, and you have not indicated that you are quoting Winands.

== 4.2.7 Results ==

- You have said nothing here about how your particular implementation of MCTS works. For example, in 4.2.2 above you mentioned the parameter constant C, "which can be tuned experimentally (e.g., C = 0.4)". What value of C did you use in your implementation? Did you compare performance with different values of C?

- The MCTS algorithm as you described in sections 4.2.1-6 works for deterministic games. But Azul is stochastic, since tiles are chosen from the bag at the beginning of each round. How did you adapt MCTS to work on a stochastic game?

- As you pointed out in the Analysis section, Azul has a fairly high branching factor, which can sometimes be 50 or more. In the most basic form of the MCTS algorithm, from any game state MCTS will try each possible move at least once before it tries any move twice. Does your implementation work this way? From your text it is unclear. This behavior may not be ideal in the presence of a high branching factor, so various authors have proposed progressive bias or progressive widening as enhancements to the MCTS algorithm. With these changes, MCTS will potentially try some (promising) moves multiple times before trying other (less promising) moves at all. Have you implemented either of these in some form? This entire topic (using MCTS on a game with a high branching factor) is highly relevant to your work, and deserves a larger discussion in the text.

- It's strange that table 4.2 appears in the middle of section 4.3 Strategy. Really it should appear in this section. Also, in the title of table 4.2, "Monte carlo" should be "Monte Carlo", or (better yet) "MCTS", which is more accurate.

- Why was MonteCarloAI not tested versus the other AIs (e.g. RandomAI, GreedyAI)?

- I tried playing against MonteCarloAI a couple of times in the GUI, with MonteCarloAI as player 1 and me (Human) as player 2. I noticed that in the first round, MonteCarloAI often makes moves that seem quite poor, i.e. moving tiles directly to the floor when there are clearly free pattern lines where the tiles could be placed. Similarly, it will often place tiles on pattern lines which are small and so some tiles overflow onto the floor, even though it clearly could have avoided this floor overflow. Have you seen this

too? If so, is there some explanation for these poor moves? Could the agent be improved so that it does not make them?

== 4.3.3 Pseudocode ==

- "sort_gropus" should be "sort_groups".

== 4.4 Results ==

- As in the previous results tables, I think that n = 50 is too small to determine an accurate win rate, and you should show confidence intervals for the win rates here.

- It's awkward that Table 3 appears in the middle of a paragraph of text.

== 4.4.1 Further development ==

- "strategyAI" should be "StrategyAI".

== 4.5.2 GreedyAI ==

- You write that the greedy AI algorithm "can be generally described" as minimax with depth 1. The language here is a bit vague. Is this algorithm precisely the same as your minimax algorithm with depth 1? If so, you should say that explicitly. Or is it not exactly the same? If not, what are the differences?

== Conclusion ==

- "MonteCarlo" should be the full name "Monte Carlo tree search".

== Bibliography ==

- Your references for the Knuth/Moore and Winands articles do not indicate where these articles were published or can be found.

== English ==

- There are various minor grammatical errors throughout the text, mostly involving articles. For example:

2.2 Complexity

  "each containing minimum 1 and maximum 4 different colors" => "each containing a minimum of 1 and a maximum of 4 different colors"

  "Assuming it's a beginning" => "Assuming that it's the beginning"

  "or to floor" => "or onto the floor"

4.2.2 Selection

  "from already built tree" => "from the existing tree"

  "it applies following formula" => "it applies the following formula"

  "to calculate value of a node" => "to calculate the value of a node"

4.2.4 Simulation

  "won by current player" => "won by the current player"

4.2.7 Results

  "against a MinimaxAI" => "against MinimaxAI"

4.3 Strategy

  "experience with game" => "experience with the game"

4.3.1

"gonna choose" => "going to choose" ("gonna" is not used in formal writing)