Check the following elements:

- There is something in the git repository
- The "auteur" file, if required by the subject, is present and valid
- The Makefile, if required, is present and has the required rules

If one of these elements is not in confirmity with what the subject requires, the session stops. You may still debate on the project, but you are not to grade the student(s).

During the rest of this session, if the program has an inappropriate behaviour (Segfault, bus error, double-free, uncaught exception, etc ...), the session stops.

✓ Yes ✗ No

Actually running the program

Rules

The students have to show that all the required game rules are implemented correctly.

If they are not, or the students can not prove it, do not grade this section.

✓ Yes ✗ No

UI and AI performance

It must be possible to play as two human players, either on the same computer or over the network, AND to play against the AI.

Grade this question according to the average performance of the AI against a player who is actually trying to win:

- Any performance but AI takes more than half a second to find a move, or the students did not include a timer to indicate how much time the AI takes -> 0
- Player victory in under 10 turns -> 0
- Player victory in 10 to 20 turns -> 1
- Player victory after 20+ turns -> 2
- Draw -> 3
- AI victory after 20+ turns -> 4
- AI victory in under 20 turns -> 5

Rate it from 0 (failed) through 5 (excellent)

Algorithm and implementation

In this section, the students must be able to THOROUGHLY explain their Minimax-family algorithm. If they can not explain it well, then they do not understand it well enough, so do not grade this section. REPEAT : IF THE STUDENTS CAN NOT EXPLAIN THEIR ALGORITHM IN DETAIL, THEIR IMPLEMENTATION IS WORTH EXACTLY NOTHING, SO DO NOT GRADE THIS SECTION.

Minimax algorithm

Look at the implementation of the Minimax algorithm :

- No actual Minimax-type algorithm -> 0
- "Naive" Minimax implementation (minimax, negamax, ...) -> 3
- "Improved" Minimax implementation (Alpha-beta pruning, negascout, mtdf, ...) -> 5

Rate it from 0 (failed) through 5 (excellent)

Move search depth

Evaluate the search depth of the Minimax tree here. If the implementation is a pruning one, like Alphabeta, take into account the actual effective search depth, not the initial one.

- Only 1 level -> 0
- 2 levels -> 1
- 3 to 5 levels -> 2
- 5 to 10 levels -> 4
- 10 or more levels -> 5

Rate it from 0 (failed) through 5 (excellent)

Search space

Evaluate the search space of the algorithm

- Entire board -> 0
- Rectangular window around all placed stones -> 3
- Multiple rectangular windows emcompassing placed stones but minimizing wasted space -> 5

Rate it from 0 (failed) through 5 (excellent)

Heuristic

In this section, the students must be able to THOROUGHLY explain their heuristic function. If they can not explain it well, then they do not understand it well enough, so do not grade this section. REPEAT : IF THE STUDENTS CAN NOT EXPLAIN THEIR HEURISTIC IN DETAIL, THEIR IMPLEMENTATION IS WORTH EXACTLY NOTHING, SO DO NOT GRADE THIS SECTION.

Static part - Alignments

Does the heuristic take current alignments into account ?

✓ Yes | ✕ No

Static part - Potential win by alignment

Does the heuristic check whether an alignment has enough space to develop into a 5-in-a-row ?

✓ Yes | ✕ No

Static part - Freedom

Does the heuristic weigh an alignment according to its freedom (Free, half-free, flanked) ?

✓ Yes | ✕ No

Static part - Potential captures

Does the heuristic take potential captures into account ?

✓ Yes | ✕ No

Static part - Captures

Does the heuristic take current captured stones into account ?

✓ Yes | ✕ No

Static part - Figures

Does the heuristic check for advanteageous combinations ?

✓ Yes | ✕ No

Static part - Players

Does the heuristic take both players into account ?

✓ Yes | ✕ No

Dynamic part

Does the heuristic take past player actions into account to identify patterns and weigh board states accordingly ?

✓ Yes | ✕ No

Bonuses

Bonuses

Rate interesting and/or useful and/or just plain cool bonuses.

1 point per identifiable, separate bonus

Rate it from 0 (failed) through 5 (excellent)

Ratings

Don’t forget to check the flag corresponding to the defense

Ok | Outstanding project

Empty work | Incomplete work | No author file | Invalid compilation | Norme | Cheat

Crash | Incomplete group | Forbidden function

Conclusion

Leave a comment on this evaluation

Finish evaluation