

Software Lab

Implementation of Heuristic Algorithms for Board Games

Harold Bruintjes, Christian Dehnert

April 8th, 2014

Contents

- 1 Organisation
- 2 Goals of this lab
- 3 Marking of this lab
- 4 Reversi
- 5 Infrastructure
- 6 Assignment 1

Contents

- 1 Organisation
- 2 Goals of this lab
- 3 Marking of this lab
- 4 Reversi
- 5 Infrastructure
- 6 Assignment 1

Organisation

- ▶ Course webpage:

<http://moves.rwth-aachen.de/teaching/ss-14/swp14/>

- ▶ Fortnightly assignments and meetings
- ▶ SSELab for group and shared projects
- ▶ Deadlines: Reports and code in the repo
- ▶ Reports in Latex (template in the shared repo)
- ▶ Code in Java 7 or C++11 (gcc 4.8 or clang 3.2)

Meetings

Two fortnightly meetings:

- ▶ Plenary meeting:
WE make announcements, introduce new assignments
YOU ask questions, raise issues
- ▶ Group meeting:
YOU give a *short* presentation of your work
WE ask questions, give feedback

Meetings

Two fortnightly meetings:

- ▶ Plenary meeting:
WE make announcements, introduce new assignments
YOU ask questions, raise issues
- ▶ Group meeting:
YOU give a *short* presentation of your work
WE ask questions, give feedback

Best-case suggestion:

Reserve **one day** (e.g. Friday 15:30) for plenary meeting and meet all groups throughout that day

21 participants registered \Rightarrow 7 groups with 3 people each

21 participants registered \Rightarrow 7 groups with 3 people each

You are free to form your own groups

Contents

- 1 Organisation
- 2 Goals of this lab
- 3 Marking of this lab
- 4 Reversi
- 5 Infrastructure
- 6 Assignment 1

Goals

- ▶ work proactively
- ▶ team work
- ▶ writing, presentation, coding skills
- ▶ have fun
- ▶ beat all others in the tournament (reward!)

Contents

- 1 Organisation
- 2 Goals of this lab
- 3 Marking of this lab**
- 4 Reversi
- 5 Infrastructure
- 6 Assignment 1

Marking

Each fortnight we will give you feedback and a mark based on your

- ▶ team work
- ▶ written report
- ▶ code quality

Marking

Each fortnight we will give you feedback and a mark based on your

- ▶ team work
 - ▶ presentation
 - ▶ communication
 - ▶ equal contribution

Marking

Each fortnight we will give you feedback and a mark based on your

- ▶ written report
 - ▶ **evaluation and discussion**
 - ▶ problem statement
 - ▶ solution description and **justification**
 - ▶ grammar, spelling, structure

Marking

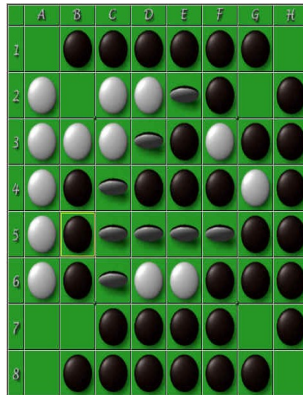
Each fortnight we will give you feedback and a mark based on your

- ▶ code quality
 - ▶ **fulfils the given task?**
 - ▶ javadoc (or doxygen) comments describing the input/output and behaviour of a method
 - ▶ inline comments where necessary
 - ▶ modularity
 - ▶ consistency
 - ▶ self-documenting code
 - ▶ assertions
 - ▶ unit testing
 - ▶ code reuse

Contents

- 1 Organisation
- 2 Goals of this lab
- 3 Marking of this lab
- 4 Reversi**
- 5 Infrastructure
- 6 Assignment 1

(A variant of) Reversi



Reversi Plus X

where X is ...

- ▶ up to 8 players per map

Reversi Plus X

where X is ...

- ▶ up to 8 players per map
- ▶ arbitrary transitions

Reversi Plus X

where X is ...

- ▶ up to 8 players per map
- ▶ arbitrary transitions
- ▶ override stones

Reversi Plus X

where X is ...

- ▶ up to 8 players per map
- ▶ arbitrary transitions
- ▶ override stones
- ▶ expansion, choice, inversion, bonus fields

Reversi Plus X

where X is ...

- ▶ up to 8 players per map
- ▶ arbitrary transitions
- ▶ override stones
- ▶ expansion, choice, inversion, bonus fields
- ▶ bombing phase.

Contents

- 1 Organisation
- 2 Goals of this lab
- 3 Marking of this lab
- 4 Reversi
- 5 Infrastructure**
- 6 Assignment 1

bundles various services needed for coding projects at

www.sselab.de

bundles various services needed for coding projects at

www.sselab.de

- ▶ **git repository** (incl. commit notifications)

put your solutions here!

bundles various services needed for coding projects at

www.sselab.de

- ▶ **git repository** (incl. commit notifications)

put your solutions here!

- ▶ mailing list

bundles various services needed for coding projects at

www.sselab.de

- ▶ **git repository** (incl. commit notifications)

put your solutions here!

- ▶ mailing list

optimal: make use of everything!

necessary: show activity!

You will be invited to two SSELab projects:

swp-i2-2014-pm (shared repo and mailing list)

and

swp-i2-2014-gX where X = group number
(private repo and mailing list)

After joining, you will receive information about all services.

Technical Details I

- ▶ Code and reports must be provided on a 'stable' branch
- ▶ Makefiles/Antfiles must be provided for automated compilation
- ▶ your code needs to compile properly (Java 7, g++ 4.8, clang 3.2)
- ▶ only standard libraries
- ▶ communication via network (use sockets)
- ▶ no multi-threading
- ▶ no busy waiting, only idling when it's someone else's turn

Technical Details I

- ▶ Code and reports must be provided on a 'stable' branch
- ▶ Makefiles/Antfiles must be provided for automated compilation
- ▶ your code needs to compile properly (Java 7, g++ 4.8, clang 3.2)
- ▶ only standard libraries
- ▶ communication via network (use sockets)
- ▶ no multi-threading
- ▶ no busy waiting, only idling when it's someone else's turn

strict requirements:

- ▶ no disqualifications after some point (indicated by exercises)
- ▶ better than trivial AI in the end

Technical Details I

- ▶ Code and reports must be provided on a 'stable' branch
- ▶ Makefiles/Antfiles must be provided for automated compilation
- ▶ your code needs to compile properly (Java 7, g++ 4.8, clang 3.2)
- ▶ only standard libraries
- ▶ communication via network (use sockets)
- ▶ no multi-threading
- ▶ no busy waiting, only idling when it's someone else's turn

strict requirements:

- ▶ no disqualifications after some point (indicated by exercises)
- ▶ better than trivial AI in the end

further questions: written specification, **ask us**

Technical Details II

- ▶ matches “continuously” run on cluster
- ▶ a webinterface will be made available to monitor the matches, and the performance of your client
- ▶ your code will be pulled and compiled automatically

Contents

- 1 Organisation
- 2 Goals of this lab
- 3 Marking of this lab
- 4 Reversi
- 5 Infrastructure
- 6 Assignment 1**

First assignment

1. create 3 different maps
2. create data structure to represent maps
3. write an algorithm to check for valid moves
4. create two different functions to rate a player's state
5. expand your algorithm to take special fields and stones into account
6. write a (short) **report**