Gomoku AI protocol

Introduction

This document describes communication between a brain (an artificial intelligence) and a go-moku manager (a program which manages and controls a tournament). The manager creates two pipes. The first pipe is used to send commands from the manager to the brain. The second pipe is used to send replies from the brain to the manager. The brain uses standard input-output functions (scanf and printf in C, readln and writeln in Pascal,...) therefore it can be written in any programming language. The brain must be a console application, not Windows GUI. Be careful, some run-time libraries buffer the console output. For example, it is necessary to call fflush(stdout) after printf in C programs. You can also use low-level functions ReadFile and WriteFile.

Each line contains exactly one command (there is only one exception). The manager puts bytes CR LF (0x0d, 0x0a) at the end of lines. The brain can send lines that are ended with CR LF, or just LF, or CR. The manager ignores lines that are empty. It must not crash if a line is too long, but it can silently cut very long lines.

If the brain has only one thread, it is very important not to read from input when the brain is required to think or respond to a command. It would lead to deadlock (both brain and manager are waiting). Manager will terminate the brain in this situation after the time for a turn is out. The brain can have two threads to avoid that problem. The first thread reads commands from input and the second thread thinks and writes responds to output. It is necessary to use some synchronization objects (events, locks or semaphores). The number of threads is not important for a tournament. One thread is sufficient for a tournament. But two threads are useful for human players. For example, someone may want to change time limits while the brain already started to think. The thinking can also be easily canceled at any time without terminating the brain.

The brain is required to process commands START, BEGIN, INFO, BOARD, TURN, END. The brain can ignore INFO commands which are not needed. The reply for every other command is UNKNOWN (Due to backward compatibility and possibility to extend the protocol).

Brain's name and temporary files

The name of the brain can contain only characters A-Z, a-z, 0-9, dash, underscore, dot. The name is required to begin with prefix "pbrain-".
Example:
pbrain-swine.exe
pbrain-pisq5.exe

Only the executable file is required to begin with prefix "pbrain-". The name of ZIP file does not have this prefix. There can be both 32-bit and 64-bit exe in the ZIP file, the 64-bit exe file name must have substring 64. For example, MyGomo.ZIP can contain files pbrain-MyGomo.exe and pbrain-MyGomo64.exe. The manager launches pbrain-MyGomo.exe on 32-bit OS and pbrain-MyGomo64.exe on 64-bit OS.

Working directory is set by the manager. It doesn't have to be the directory where the brain's executable file is saved. The brain must specify full path to all data files which it uses. It can obtain the path from function GetModuleFileName or it can look at the beginning of its command line which can be discovered from the main function parameters. The manager must put name of brain's exe file on the command line in such a form so that the brain can open the file.

The brain can create a folder in the current directory to store its temporary files. The name of the folder must be the same as the name of the brain. The maximal allowed size of the folder will be announced on Gomocup web page (it is now 20MB). The manager can delete all temporary files when the manager exits or after a tournament finished. Command INFO folder is used to determine folder where persistent files can be saved.

Mandatory commands

START [size]

When the brain receives this command, it initializes itself and creates an empty board, but doesn't make any move yet. The parameter is size of the board. The brain must be able to play on board of size 20, because this size will be used in Gomocup tournaments. It is recommended but not required to support other board sizes. If the brain doesn't like the size, it responds ERROR. There can be a message after the ERROR word. The manager can try other sizes or it can display an error message to a user. The brain responds OK if it has been initialized successfully.

```
Example:
 The manager sends:
  START 20
 The brain answers:
  OK - everything is good
  ERROR message - unsupported size or other error
```

TURN [X],[Y]

The parameters are coordinate of the opponent's move. All coordinates are numbered from zero.

```
Expected answer:
 two comma-separated numbers - coordinates of the brain's move

Example:
 The manager sends:
  TURN 10,10
 The brain answers:
  11,10
```

BEGIN

This command is send by the manager to one of the players (brains) at the beginning of a match. This means that the brain is expected to play (open the match) on the empty playing board. After that the other brain obtains the TURN command with the first opponent's move. The BEGIN command is not used when automatic openings are enabled, because in that case both brains receive BOARD commands instead.
Expected answer:
 two numbers separated by comma - coordinates of the brain's move

```
Example:
 The manager sends:
  BEGIN
 The brain answers:
  10,10
```

BOARD

This command imposes entirely new playing field. It is suitable for continuation of an opened match or for undo/redo user commands. The BOARD command is usually send after START, RESTART or RECTSTART command when the board is empty. If there is any open match, the manager sends RESTART command before the BOARD command.

After this command the data forming the playing field are send. Every line is in the form:

[X],[Y],[field]

where [X] and [Y] are coordinates and [field] is either number 1 (own stone) or number 2 (opponent's stone) or number 3 (only if continuous game is enabled, stone is part of winning line or is forbidden according to renju rules).

If game rule is renju, then the manager must send these lines in the same order as moves were made. If game rule is Gomoku, then the manager may send moves in any order and the brain must somehow cope with it. Data are ended by **DONE** command. Then the brain is expected to answer such as to TURN or BEGIN command.

```
Example:
 The manager sends:
  BOARD
  10,10,1
  10,11,2
  11,11,1
  9,10,2
  DONE
 The brain answers:
  9,9
```

INFO [key] [value]

The manager sends information to the brain. The brain can ignore it. However, the brain will lose if it exceeds the limits. The brain must cope with situations when the manager doesn't send all information which is mentioned in this document. Most of this information is sent at the beginning of a match. The time limits will not be changed in the middle of a match during a tournament. It is recommended to react on commands at any time, because the human opponent can change these values even when the brain is thinking.
The key can be:
timeout_turn - time limit for each move (milliseconds, 0=play as fast as possible)
timeout_match - time limit of a whole match (milliseconds, 0=no limit)
max_memory - memory limit (bytes, 0=no limit)
time_left - remaining time limit of a whole match (milliseconds)
game_type - 0=opponent is human, 1=opponent is brain, 2=tournament, 3=network tournament
rule - bitmask or sum of 1=exactly five in a row win, 2=continuous game, 4=renju, 8=caro
evaluate - coordinates X,Y representing current position of the mouse cursor
folder - folder for persistent files

Information about time and memory limits is sent before the first move (after or before START command). Info time_left is sent before every move (before commands TURN, BEGIN, BOARD and SWAP2BOARD). The remaining time can be negative when the brain runs out of time. Remaining time is equal to 2147483647 if the time for a whole match is unlimited. The manager is required to send info time_left if the time is limited, so that the brain can ignore info timeout_match and only rely on info time_left.

Time for a match is measured from creating a process to the end of a game (but not during opponent's turn). Time for a turn includes processing of all commands except initialization (commands START, RECTSTART, RESTART). Turn limit equal to zero means that the brain should play as fast as possible (eg count only a static evaluation and don't search possible moves).

INFO folder is used to determine a folder for files that are permanent. Because this folder is common for all brains and maybe other applications, the brain must create its own subfolder which name must be the same as the name of the brain. If the manager does not send INFO folder, then the brain cannot store permanent files.

Only debug versions should respond to INFO evaluate. For example, it can print evaluation of the square to some window. It cannot be written to the standard output. Release versions should just ignore INFO evaluate.

How should the brain behave when obtains unknown INFO command ?
- Ignore it, it is probably not important. If it was important, it is not in an INFO command form.

How should behave the brain obtaining the unachievable INFO command?
(for example too small memory limit)
- The brain should wait with the output of the problem until the manager sends the first command not having an INFO form (TURN, BOARD or BEGIN). The manager does not read messages from the brain when sending INFO command.

Example:

```
 INFO timeout_match 300000
 INFO timeout_turn 10000
 INFO max_memory 83886080
```

Expected answer: none

END

When the brain obtains this command, it must terminate as soon as possible. The manager waits until the brain is finished. If the time of termination is too long (e.g. 1 second), the brain will be terminated by the manager. The brain should not write anything to output after the END command. However, the manager should not close the pipe until the brain is ended.
Expected answer: none
The brain should delete its temporary files.

ABOUT

The brain is expected to send some information about itself on one line. Each info must be written as keyword, equals sign, text value in quotation marks. Recommended keywords are name, version, author, country, www, email. Values should be separated by commas that can be followed by spaces. The manager can use this info, but must cope with old brains that used to send only human-readable text.
Example:

```
 The manager sends:
  ABOUT
 The brain answers:
  name="SomeBrain", version="1.0", author="Nymand", country="USA"
```

Optional commands

The extended commands published in this section are not required to be implemented in Gomocup tournament, however it can be useful especially for human players.

RECTSTART [width],[height]

This command is similar to START, but the board is rectangular. Parameters are two comma-separated numbers. Width corresponds to coordinate X, height belongs to coordinate Y. The manager must use START command if the board is squared.
Example:

The manager sends:
 RECTSTART 30,20
The brain answers:
 OK - parameters are good
 ERROR message - rectangular board is not supported or other error

RESTART

This command is used after the match is finished or aborted. This command has no parameters. The board size remains unchanged. The brain releases previous board and other structures, creates a new empty board and prepares itself for a new match. Then the brain answers OK and further communication continues as after START command. If the brain answers UNKNOWN, the manager sends END and executes the brain again.
Example:
The manager sends:
 RESTART
The brain answers:
 OK

TAKEBACK [X],[Y]

This command is used to undo last move. The brain removes stone which has coordinate [X,Y] and then answers OK.
Example:
The manager sends:
 TAKEBACK 9,10
The brain answers:
 OK

PLAY [X],[Y]

It is used by the manager as a respond to SUGGEST command only. It imposes move [X],[Y] to the brain.
Expected answer: two comma-separated numbers which should be the same as PLAY parameters. If the brain doesn't like the coordinates sent by the manager, it can answer some other coordinates (but it is not recommended).
Example:
The brain has sent:
 SUGGEST 10,10
The manager sends:
 PLAY 12,10
The brain moves onto 12,10 and answers:
 12,10

SWAP2BOARD

This command is used to deal with the opening stage of **Swap2** rule. It is sent once or twice to the AI between command START and command BOARD. Specifically, it has three cases, and we show examples for each of them.

Case 1. The manager asks for the first three stones.

The manager sends:

```
 SWAP2BOARD
 DONE
```

The AI answers:

```
 7,7 8,7 9,9
```

-

Case 2. The manager sends the coordinates of the first three stones and asks for the choice of options.

The manager sends:

```
 SWAP2BOARD
 7,7
 8,7
 9,9
 DONE
```

The AI answers:

SWAP - if the AI decides to swap (option 1)

8,8 - output the coordinate of the 4th move if the AI decides to stay with its color (option 2)

8,8 8,6 - output the coordinates of the 4th and 5th stones if the AI decides to put two stones and let the opponent choose the color (option 3)

-

Case 3. The manager sends the coordinates of the first five stones and asks for the choice of options.

The manager sends:

```
 SWAP2BOARD
 7,7
 8,7
 9,9
 8,8
 8,6
 DONE
```

The AI answers:

SWAP - if the AI decides to swap (option 1)

6,8 - output the coordinate of the 6th move if the AI decides to stay with its color (option 2)

-

After the opening stage, the stones on the board will be treated as an opening for a standard match. For example, following the above example in Case 3, assuming the AI chooses option 2, the manager will send the following messages to the other AI:

```
 BOARD
7,7,1
```

```
8,7,2
9,9,1
8,8,2
8,6,1
6,8,2
DONE
```

-

As another example, following Case 2's example, assuming the AI chooses option 1, the manager will send the following messages to the other AI:

```
 BOARD
7,7,2
8,7,1
9,9,2
DONE
```

-

Commands that are sent by the brain

The brain can use these commands if it needs them. The manager is required to know them.

UNKNOWN [error message]

The brain sends this as a respond to a command that is unknown or not yet implemented. That means the brain must not exit after receiving some strange line from the manager. The parameter after UNKNOWN keyword is message that can be displayed by the manager to a user. If the manager has sent some optional command which the brain does not implement, then the manager is required to try some mandatory command.

ERROR [error message]

The brain sends this when it receives some known command, but is not able to cope with it. For example the memory limit is too small or the board is too large. Parameter is a short message that the manager writes to a log window or to a log file. The manager can also change some game options and try action again.

MESSAGE [message]

The message for a user. The manager can write it to some log window or to a log file. The brain is expected to send messages just before respond to some command. There must not be a new line character inside a message. Multi-line text can be sent as a sequence of two or more MESSAGE commands.

It is recommended to send only English messages. If the brain chooses another language, it should detect code page that is used on the PC (Win32 function GetACP()) and should not send messages that cannot be displayed in that code page.

DEBUG [message]

It is similar to MESSAGE command, but it is used for debugging information that is useful only for the author of the brain. These messages will not be visible to public in Gomocup tournament.
Example:

```
 The manager sends:
  TURN 10,15
 The brain answers:
  DEBUG The most promising move now is [10,14] alfa=10025 beta=8641
  DEBUG The most promising move now is [11,14] alfa=10125 beta=8641
  MESSAGE I will be the winner
  10,16
```

SUGGEST [X],[Y]

The brain can answer SUGGEST [X],[Y] instead of [X],[Y] and does not change it's internal state. The manager has a possibility to ignore brain's suggestion and force another move to the brain. Expected manager's answer is PLAY or END. In Gomocup tournament the manager always answers the move the brain has sent in the SUGGEST command. Most brains do not use SUGGEST command.

Version history

- 2023-03-20
  INFO rule 8 - Caro
- 2022-12-20
  SWAP2BOARD command
- 2016-02-07
  coordinates in BOARD command must be in correct order if rule is renju
  both 64-bit and 32-bit exe can be in ZIP
- 2016-02-02
  INFO rule 4 - renju
- 2006-03-11
  INFO rule 2 - option continuous game, also added value 3 to the BOARD command
- 2005-12-19
  INFO folder - for permanent files
  ABOUT - format changed to keyword="value"
  INFO timeout_turn 0 means as fast as possible
- 2005-06-26
  TAKEBACK - used for undo
- 2005-06-03
  INFO rule - option exactly five in a row
- 2005-05-19
  INFO game_type
- 2005-04-21
  RECTSTART - rectangular board
  RESTART - clears the board
  BOARD - it is mandatory command