2011 Fall CS202 Problem Solving
Final Project

Battle Tetris

What to do…

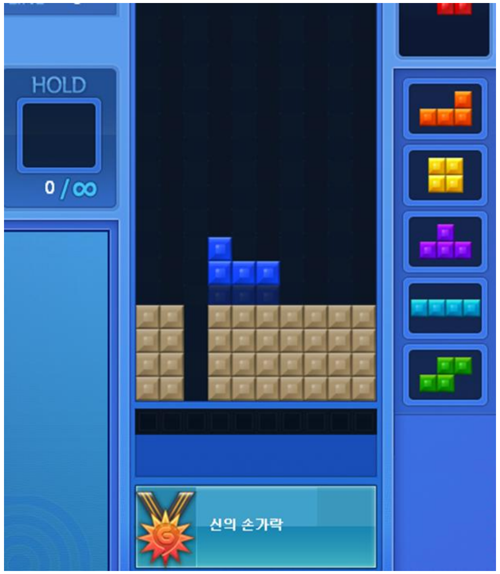
Project name: Battle Tetris

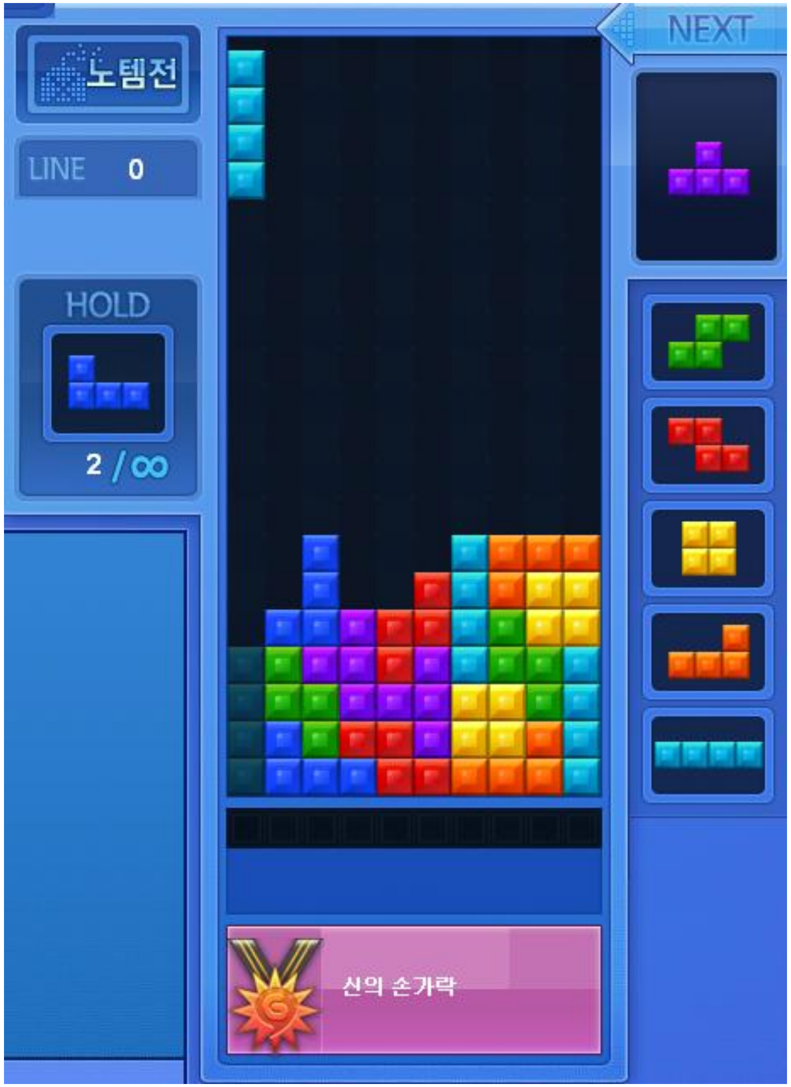
- Motivated by HanGame Tetris
    - http://tetris.hangame.com

- You should make AI algorithm for Tetris
    - Input
        - You and other person's current state of board (board, targetBoard)
        - Kind of current and next block(piece) (nextBlock)
        - And other context information (nowCombo, holdBlockNum, isOnHold, numOfGarbageLines, holePositionX)
    - Output
        - Where and how to drop the block (posX, turn)
        - Whether you will store(or exchange) the block or not (useHold)
        - Message to show

Tetris

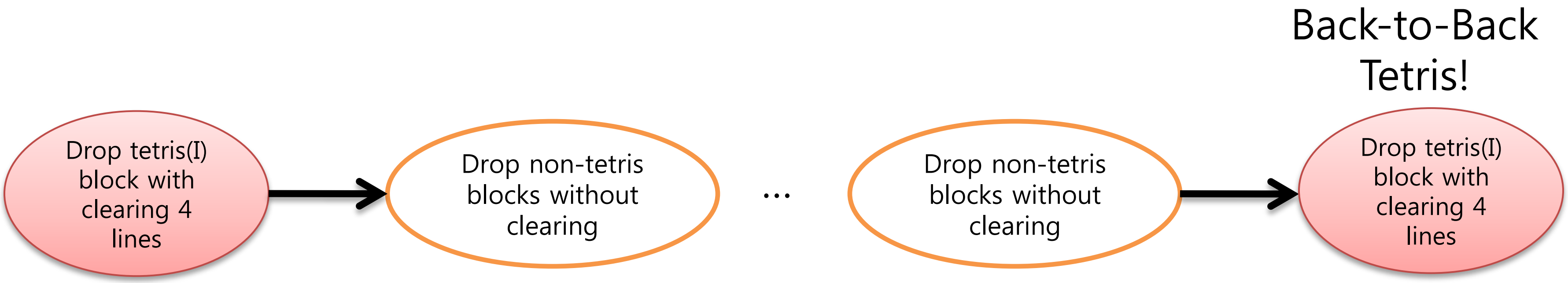
- If you delete(clear) 4 lines at once(tetris), the other party will be given 4 garbage lines
- If you delete 3 lines at once
  -> 2 garbage lines
- If you delete 2 lines at once
  -> 1 garbage line





Back-to-Back Tetris

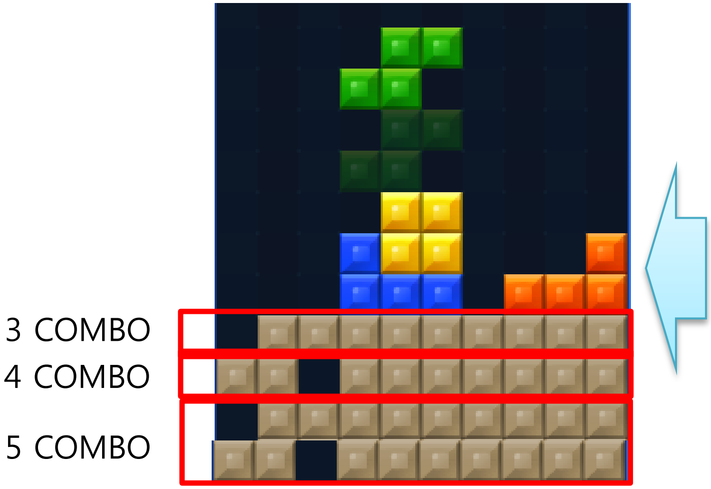
- A back-to-back tetris is, as the name states, a tetris (clearing 4 lines at a time) that's been cleared in a chain of consecutive tetrises.
  - the other party will be given 5 garbage lines


Back-to-Back Tetris!
Drop tetris(I) block with clearing 4 lines
Drop non-tetris blocks without clearing
…
Drop non-tetris blocks without clearing
Drop tetris(I) block with clearing 4 lines

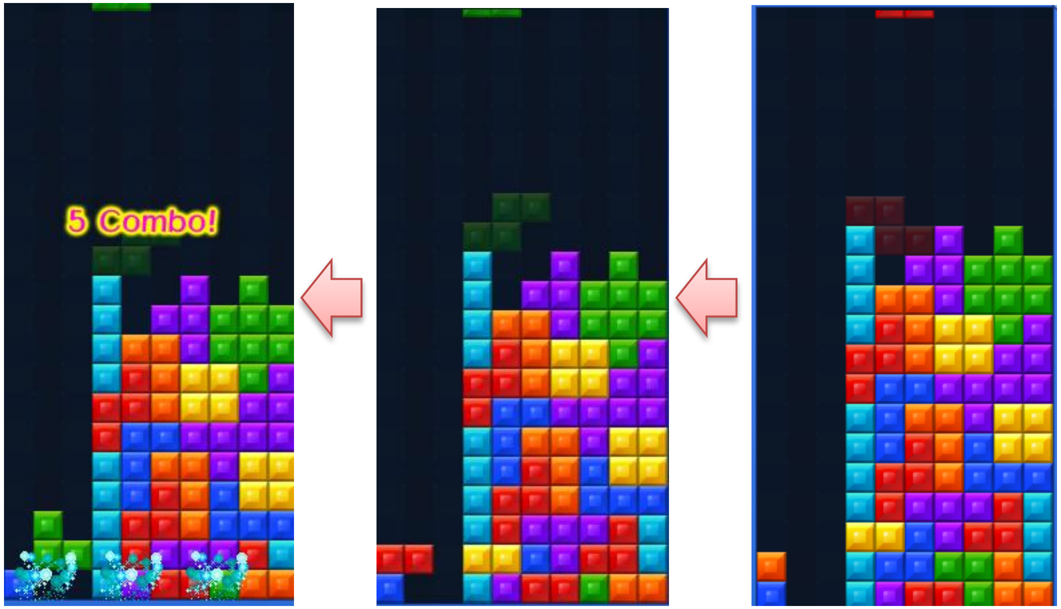
*http://answers.yahoo.com/question/index?qid=20080622152014AABZA48

Combo System

- After consecutive line deletion, the other party will be given..
  - 3~4 Combo : 1 line
  - 5~6 Combo : 2 lines
  - 7~8 Combo : 3 lines
  - 9~11 Combo : 4 lines



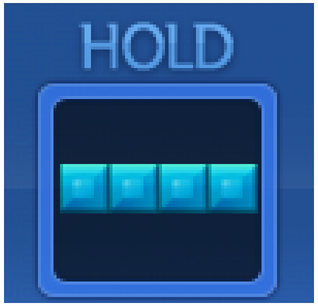
Enemy's board



My board

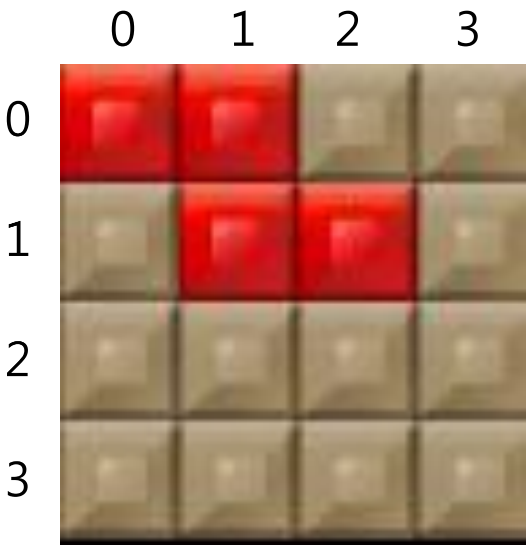
Holding Block

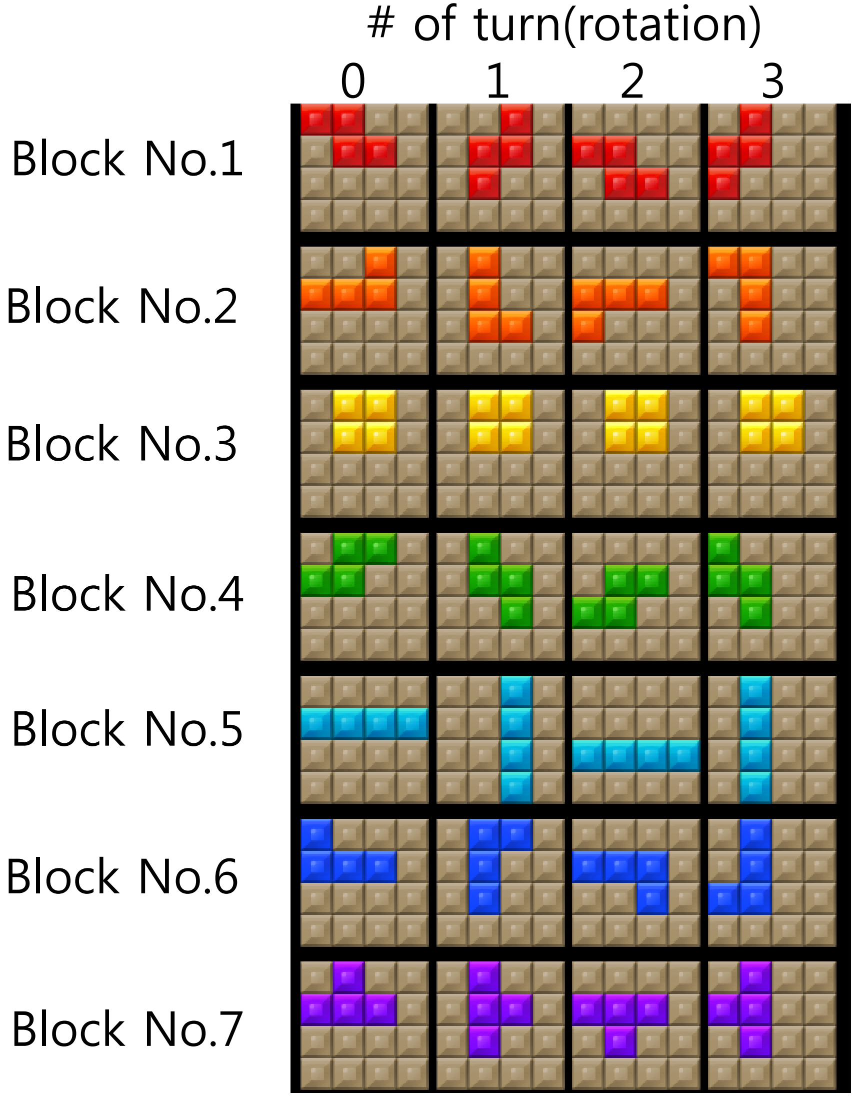
- You can store or exchange a block
  - If repository is empty
    - Current block is stored
  - Else
    - Current block is exchanged for the block in repository



- You can't use 'hold' consecutively without dropping the block

Block Information





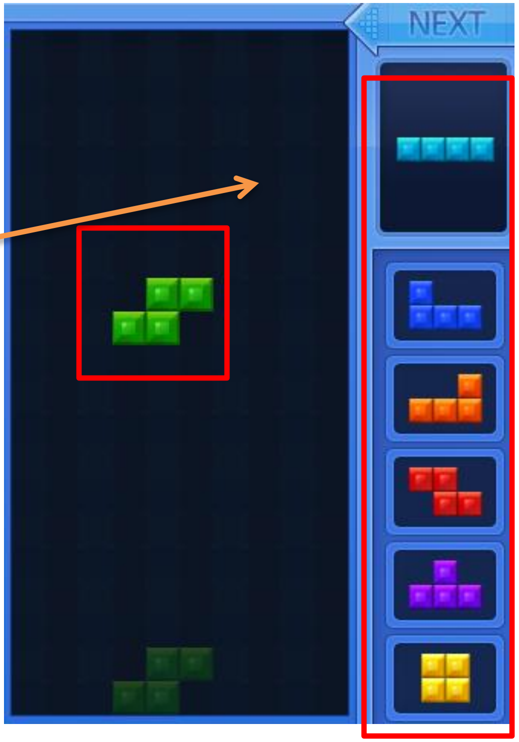
7-piece Bag System

Block count	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Block Kind No.	1	5	4	3	6	2	7	7	5	4	1	2	3	6	1	6	4
	Random order							Random order									

- Game server generates next blocks that will be used in game

- Block no. will not be overlapped in 1-7, 8-14, 15-21…

- Each block will be shown within 13 consecutive drops



Game & Decision Information

- **Game Information(you will be given)**

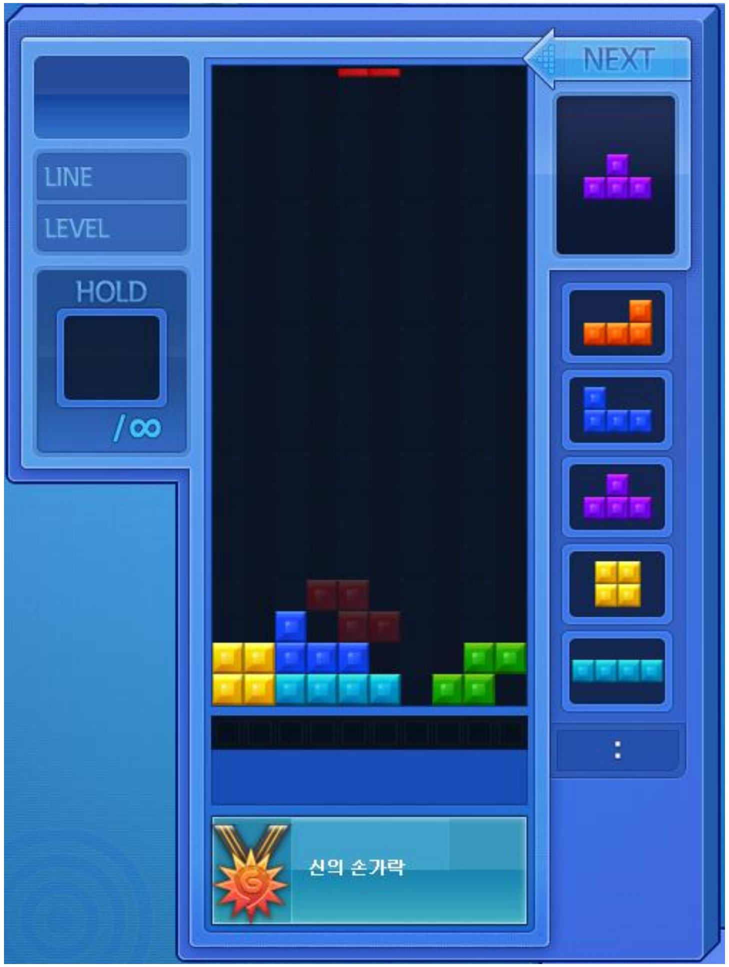
```
 board;                    //My game board
 targetBoard;              //Enemy's game board
 nextBlock[7];             //Block info.(index 0 is current block)
 nowCombo;                 // # of current combo
 holdBlockNum;             // # of current hold block(empty==0)
 isOnHold;                 // Whether 'hold' is used in previous decision.
                           //You can use 'useHold' on decision if this value is false(0)

 numOfGarbageLines;        // # of garbage lines to be pushed
 holePositionX[5];         // The x-position of the hole in garbage lines
```

- **Decision Information(you should send)**

```
posX;          // X position of dropping block(Pivot Position)
 turn;         // # of rotation
useHold;       // If useHold value is true(1), use 'hold'
message;       //Message to show (do not exceed 128 character)
```

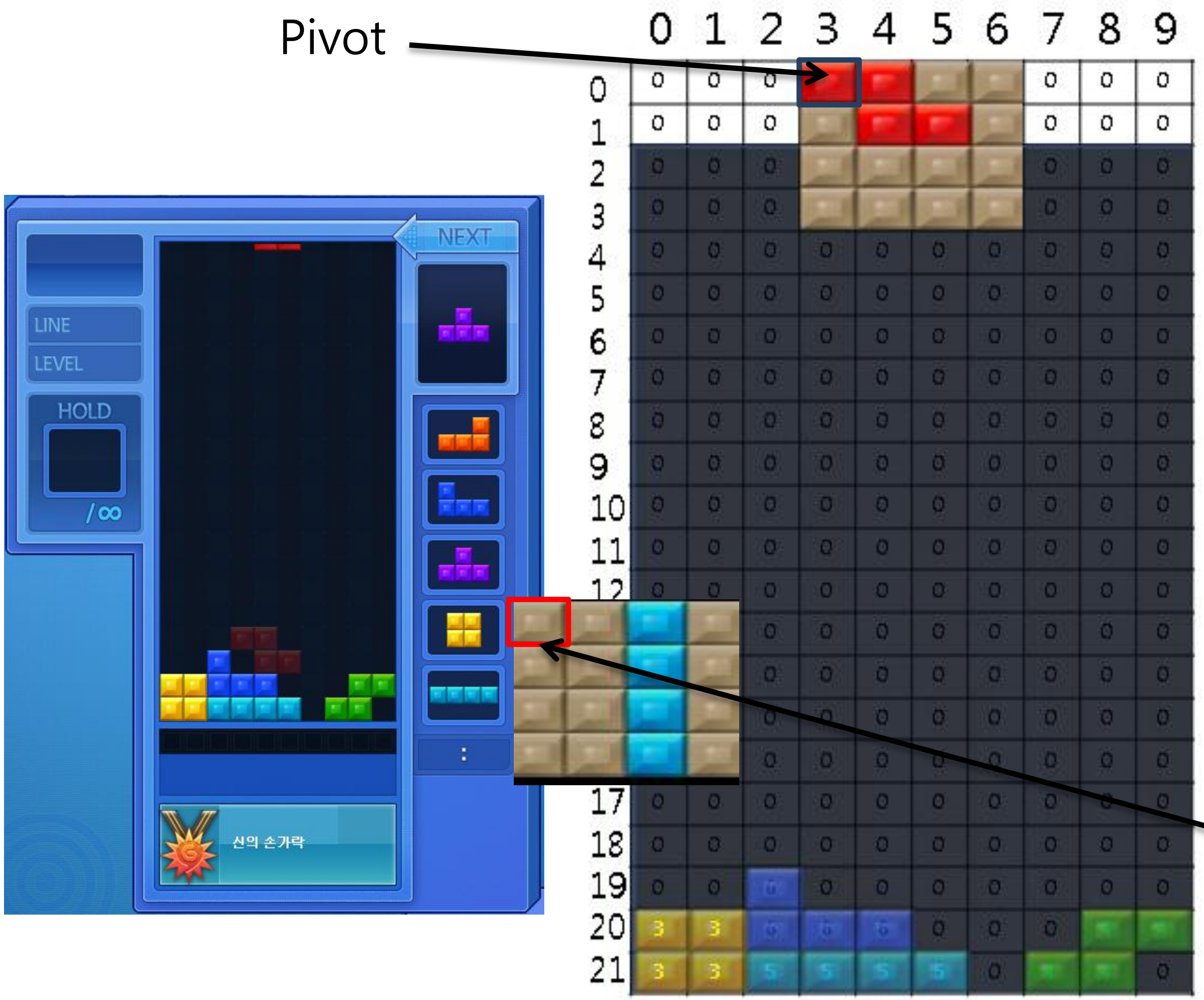
Board Presentation



	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0
19	0	0	6	0	0	0	0	0	0	0
20	3	3	6	6	6	0	0	0	4	4
21	3	3	5	5	5	5	0	4	4	0

Will not be shown

Board Presentation



Pivot

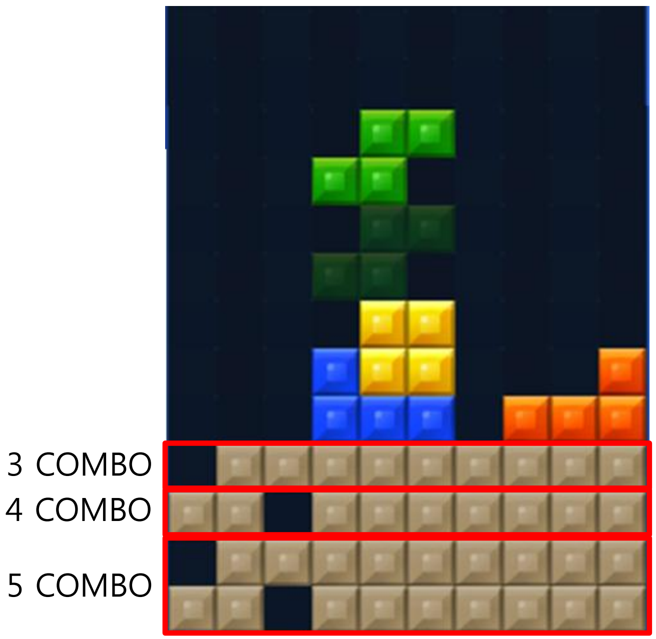
Current Block No : 1
Pos X : 3
Pos Y : 0
Turn : 0

Example)
board[20][0]= 3
board[20][8]= 4
nextBlock[]={1,7,2,6,7,3,5}

*X position of pivot can be minus value(e.g -2)

Garbage Block Info.

- Garbage line information that will be received after your decision is in ‘numOfGarbageLines’ and ‘holePositionX’



Example)

For 3 COMBO

numOfGarbageLines:1

holePositionX[4]={0,0,0,0,0}

For 4 COMBO

numOfGarbageLines:1

holePositionX[4]={2,0,0,0,0}

For 5 COMBO

numOfGarbageLines:1

holePositionX[4]={0, 2,0,0,0}

Win & Lose Condition

- You will win
  - The other party is died

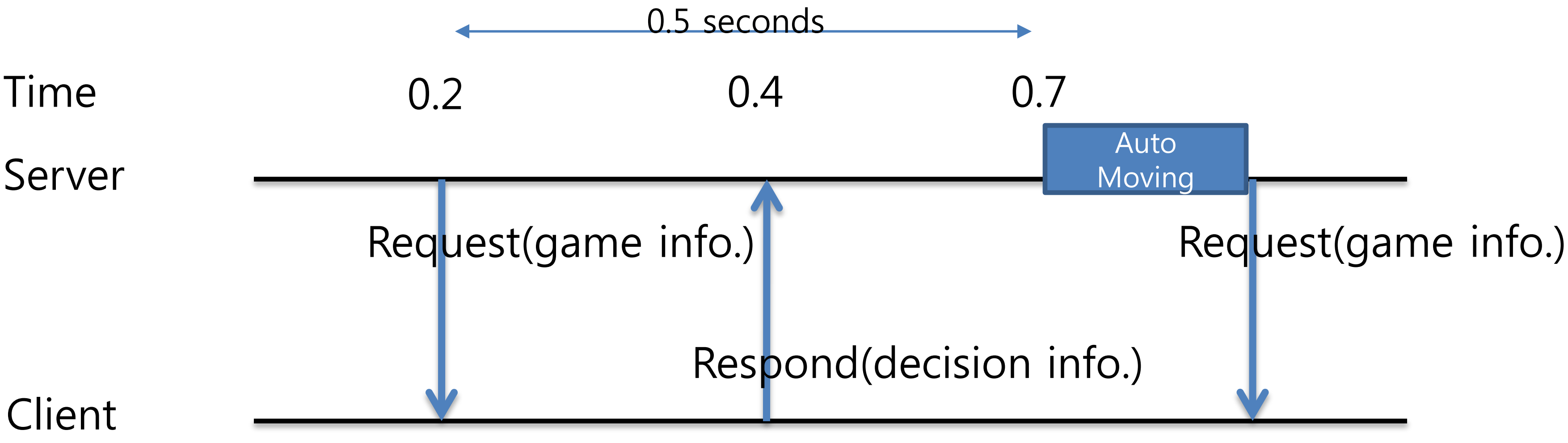
- You will lose
  - If the block that you chose is crashed the current blocks in your board or walls
  - Consecutive use of ‘hold’

- Draw condition
  - If two players are died together within 0.5 seconds

Additional Information

- Server will react your decision after '***Respond Time Limit***" seconds from the time that the server sent the game info.

Example) If Respond Time Limit is 0.5



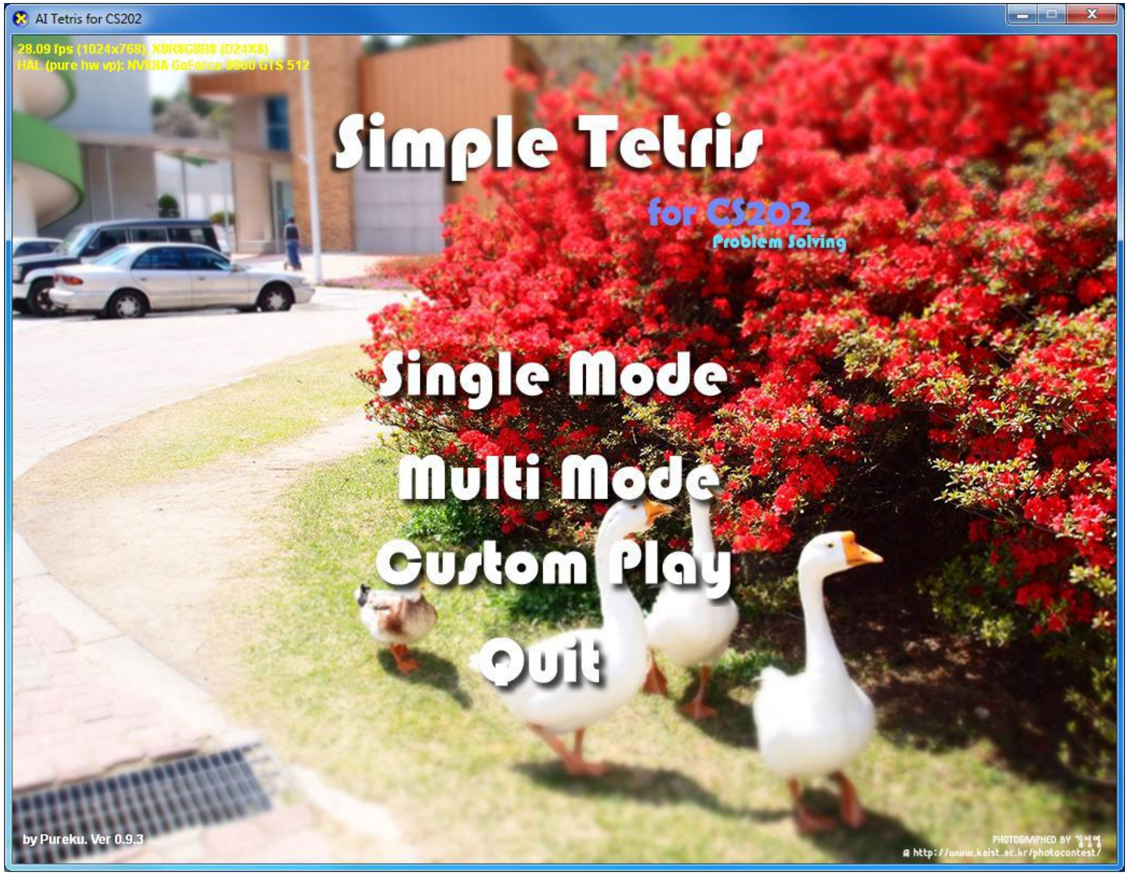
Additional Information

- For **AI vs AI** or **AI vs nobody**
  - Garbage blocks will be pushed after some time to prevent tedious play

Seconds	**60**	**70**	**80**	**90**	**95**	**100**	**105**	**110**	**115**	**120**	**125**	**130**	**135**	**140**	**142**	**145**	**147**	**150**	**153**	**155**
Lines	1	1	1	2	2	2	2	2	2	2	2	3	3	3	3	3	3	4	4	4

Game Images

- Title Screen



AI Tetris for CS202

```
28.09 fps (1024x768), X8R8G8B8 (D24X8)
HAL (pure hw vp): NVIDIA GeForce 8800 GTS 512
```

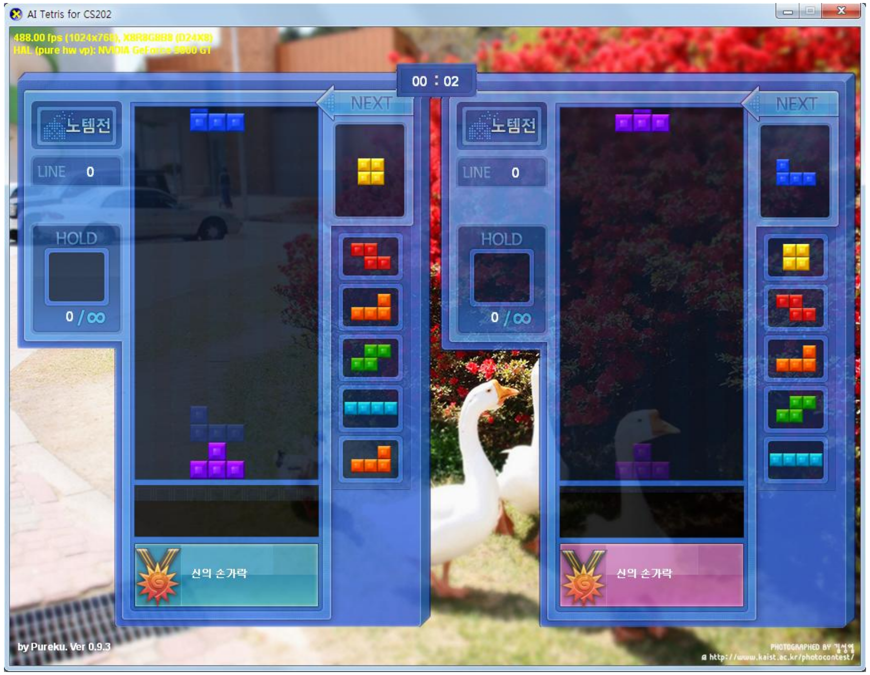
by Pureku. Ver 0.9.3

PHOTOGRAPHED BY 김성열
@ http://www.kaist.ac.kr/photocontest/

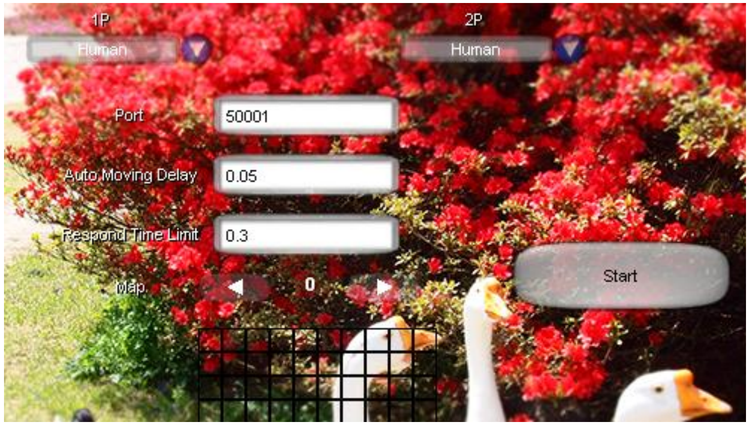
Control keys

You can control block by keyboard in 'Single Mode' or 'Multi Mode'

- **Single Mode** or '**Person vs AI**'
    - Move & Rotate: Arrow keys
    - Hard drop: Space key
    - Hold: Shift key
- **Multi Mode**
    - 1P
        - Move& Rotate: 'R','D','F','G'
        - Hard drop: Left shift key
        - Hold: 'Z'
    - 2P
        - Move& Rotate: Arrow keys
        - Hard drop: Right shift key
        - Hold: '/'



Custom Play



1P: Human

2P: Human

Port: 50001

Auto Moving Delay: 0.05

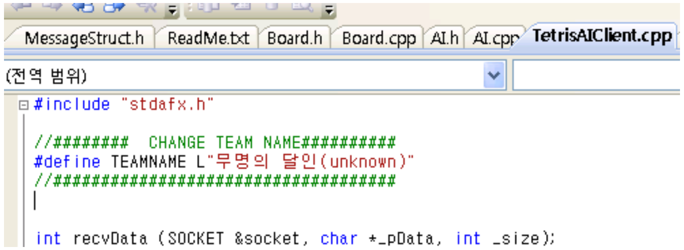
Respond Time Limit: 0.3

Map: 0

- You can connect AI client after setting and clicking ‘start’ button

AI Client

- C++ Client
  - Implemented by using visual studio 2010
  - Set team name in ‘TetrisAIClient.cpp’ file



```
#include "stdafx.h"

//########  CHANGE TEAM NAME##########
#define TEAMNAME L"무명의 달인(unknown)"
//####################################


int recvData (SOCKET &socket, char *_pData, int _size);
```

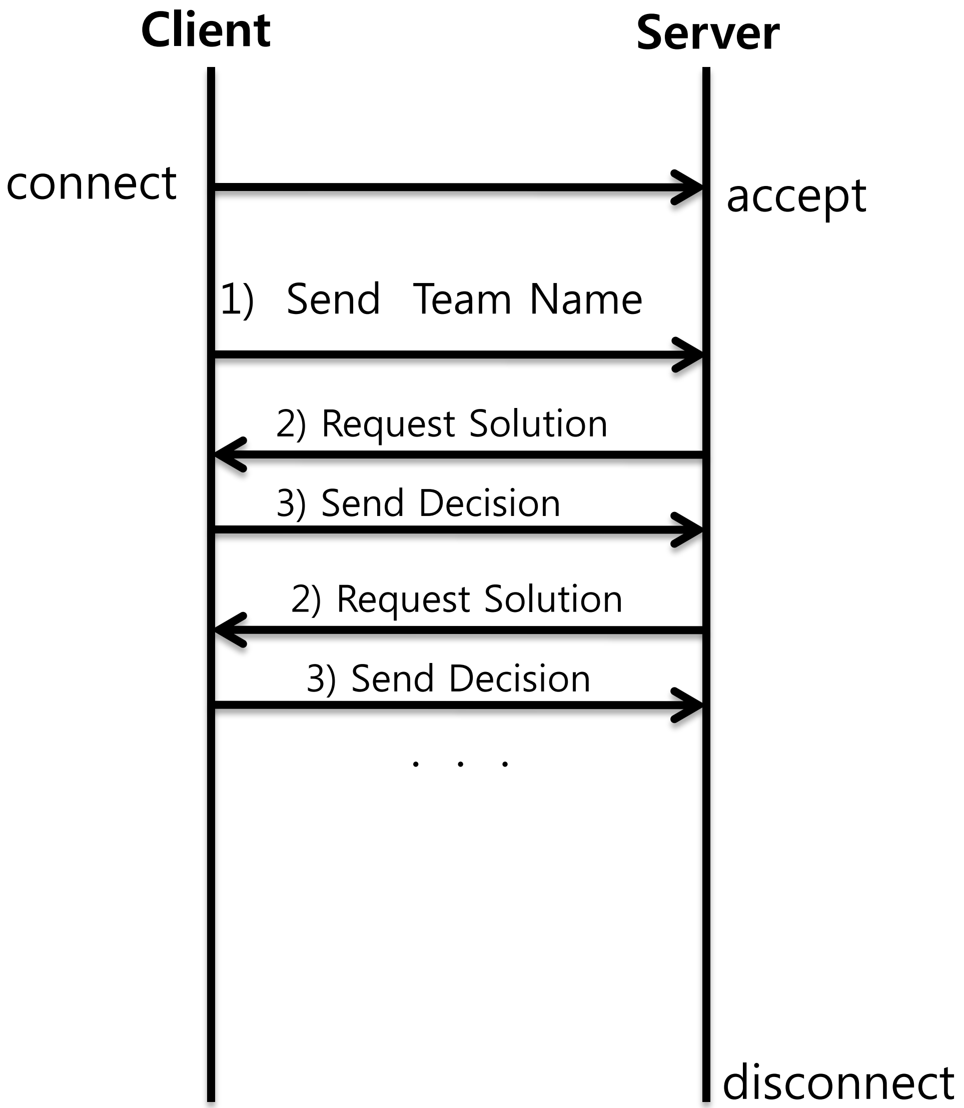
  - Implement ‘AI::decision(Game_Info gameInfo)’ method
- Java Client
  - Implemented by using eclipse(you can import project file)
  - Set team name by changing field named ‘TEAM_NAME’ in AI.java
  - Implement ‘AI.decision(GameInfo gameInfo)’ method

Sample heuristic algorithm is implemented in ‘decision’ method to show how to implement AI

2011 Fall CS202 Problem Solving
Final Project

AI Tetris Protocol Specification

Protocol Overview



Client

Server

connect

accept

1) Send Team Name

2) Request Solution

3) Send Decision

2) Request Solution

3) Send Decision

. . .

disconnect

1) Send Team Name

- Team Name [256 bytes]
  - Should be encoded with UTF-16 Little-endian
  - End with NULL(0) character

2) Request Solution

- Game information [456 Bytes]

```
 {
        BYTE board[22][10];      //My game board(BYTE is unsigned char in c)
        BYTE targetBoard[22][10]; //Enemy’s game board
        BYTE nextBlock[7];            //Block info.(index 0 is current block)
        BYTE nowCombo;              // # of current combo
        BYTE holdBlockNum;        // # of current hold block(empty==0)
        BYTE isOnHold;               // You can’t use ‘hold’ if true
        BYTE numOfGarbageLines;    // # of garbage lines to be pushed
        BYTE holePositionX[5];  // The x-position of the hole in garbage
                                          lines
}
```

3)Give Decision

- Decision information [264 Bytes]

```
{
        int posX;            // X position of dropping block
        BYTE turn;           // # of rotation
        BYTE useHold;   // Use ‘hold’ if useHold is not 0
        BYTE message[256]   //Message to show
        BYTE reserved[2]; //For aligning struct(don’t use)
}
```

  - int is little-endian byte order
  - Message should be encoded with UTF-16 Little-endian