Soliton Premier League 2015

Summary

Write a program that will play the game ‘Chain Reaction Reloaded’ with another computer program using the Soliton Premier League Dashboard.

What all are provided

A folder with all necessary tools, softwares \\Server\SPL Organizers\Question

1. “Battle Field” - This is the playing arena which will establish communication between both bots.
2. “Super Bot” - A Super Bot is provided with whom your code can play against. Note that he would make mistakes at times!. We are providing different levels of intelligence/modes to the super bot. Initially only the first mode is unlocked, once you win a mode , then the next mode will be unlocked.
3. “Simulator” - You can use this to play against any bots. This like Man Vs Bot.

The document is organized as follows

- How to play the game
- How to write a program
- Evaluation

How to Play the Game

Terminologies and Conventions

- Board - This refers the **Chain Reaction Reloaded** board

- Coin - These are the objects that you place on the board to play the game. First player would have red coins and second player would have green coins

- King - A special coin and each team would have one King. The objective is to find the opponent's King and eliminate using Chain Reaction (explained later) before opponent takes your King.

- Cells - These are the places where the coins and the King are placed. There are 49 cells arranged in 7x7 fashion. Each cell could either be empty, or have multiple coins (1, 2 or 3) of the same player depending on the maximum it can hold or can have a King

- Board Numbering - The cells are numbered as follows

1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	32	33	34	35
36	37	38	39	40	41	42
43	44	45	46	47	48	49

- Maximum capacity of a cell - A cell can only hold a specific number of coins depending on the location of cell. The maximum cell capacity of each cell is shown below.

1	2	2	2	2	2	1
2	3	3	3	3	3	2
2	3	3	3	3	3	2
2	3	3	3	3	3	2
2	3	3	3	3	3	2
2	3	3	3	3	3	2
1	2	2	2	2	2	1

- Owning a cell - A cell can be owned by a player if the cell contains at least one coin of the player. Cell owners can be changed in course of game using Chain Reaction. Player can place coins only in empty cell or the cells that they own

- Cell Burst and Chain Reaction - If a cell exceeds its maximum capacity it would burst and give one coin to the neighbouring cells (top, right, bottom and left) depending on cell location.
    - Cell 1 would burst when it has 2 coins. It would give one coin each to cell 2 and 8 and cell 1 would be empty. Similarly cell 7, 43, and 49 also would burst when there are two coins placed on it
    - Cell 2 to 6 would burst if it has 3 coins. Cell 2 would burst and give coins to cells 1, 3 and 9. Same case with the leftmost row, rightmost row and bottom most row excluding the corners.
    - Every other cell in the centre would burst if there are 4 coins and it would give to neighbours. Ex: Cell 9 would burst when there are four coins and would give one coin each to 2, 10, 16 and 8.
    - The neighbouring cells now would be owned by the current player and neighbour cell can ‘Burst’ again if its maximum capacity is reached. This is the chain reaction
    - If the neighbouring cell has the current player’s King, the coin is sucked by the King. If the neighbouring cell has opponent’s King, then the current player wins

- **King Swap** - A player can change the position of their King by swapping to another cell of theirs or an empty cell. Ex: Say for Player 1 King was in cell 5 and cell 40 is owned by Player 1 with 2 coins, Player can move the King to cell 40 and bring the contents to cell 40 to cell 5. The number of times you can move King is determined ahead of the game.

Playing the game

- The first player would be decided by the toss and has to play first.
- The first move of each player is to place the King. Player 1 would place the King in any of cell from 1 to 21 and player 2 would place in any of the cell from 29 to 49. The opponent's King will not be revealed to the players and is hidden.
- Now each player takes turn to either
    - Place their coin - Player can place their coin either in an empty cell or in their cell.
        - If a player places his coin on the hidden King of the opponent thinking that it is an empty cell, theN both the King's would be revealed from the next move of the current player.
        - Eg: Both the Kings are not revealed and Player2's King is in cell 40. Player 1 is playing her 5 the move. Player 1 tries to place the coin in cell 40 thinking that it is empty. Player 1 would get an error with a special error code. Then Player2 would his 5th move and he would play as usual. From Player 1's 6th move the both the Kings will be revealed to each other.
    - Swap the King - A player can change the position of their King by swapping to another cell of theirs or to an empty cell he has not used all swaps yet. Ex: Say for Player 1, King was in cell 5 and cell 40 is owned by Player 1 with 2 coins, Player can move the King to cell 40 and bring the contents to cell 40 to cell 5.
- After each move, the cells are checked for burst condition and chain reaction is triggered and the board is updated
    - During chain reaction if the bursted coin hits the current player's King, it is absorbed in by the player's King.
    - During chain reaction if the bursted coin hits the opponent's King, the current player wins!
- Players take turn and continue to play till one of them wins. The winning condition is determined as follows

Winning Conditions & Points

- The chain reaction that the player triggers goes and hits the opponent's King, the player who triggered the chain reaction would win.
- If a player owns every cell of the board except the opponent's King (i.e., the opponent cannot place a coin or swap the King), the player who owns all the cell wins
- If a player makes more than 2 wrong placements/moves or swaps. (Eg. Player tries to place his coin in opponents cell etc), the player loses and opponent would win
- Walkover/Abort - If a player walks over does not play his move within 3 seconds, the opponent is considered as the winner
- A tie is not possible and every winning match will earn the player 1 point.

Special Cases

- The number of times you can swap the King is set at the beginning of the game

- What if I swap my King to opponent's King when both are hidden
    - Its an error and a strategy to know the his secret without revealing your's

- In my first move, what if I place my King in a different cell than what is asked for?
    - Its an error and your King will be placed by the application either in cell 1 or cell 49 and would be revealed.

How to write your program

Before starting to read this section, ensure that you have understood how to play the game, how to use swaps and how the winner would be decided. You should have got a standardized team name by now.

Terminologies and Conventions

Read the terminologies section above.

- The first player would be decided by the toss and the color of the first player is red
- TeamName - You already have team name and a standardized team name (6-10 alphanumeric with underscore) is here.
- Folder for your team - C:\Soliton Premier League 2015\Battle Field\TeamName and would be called as *TeamFolder* from here on. Ensure that you use the team names as in this sheet
    - Example C:\Soliton Premier League 2015\Battle Field\ABC

Make this TeamFolder variable as a configurable one, read it from an ini file or make the user configure it. You should not have to rebuild you code if you have to change the path.

Writing the code

1. On start up of your application, it should create your team-folder. If the folder exist it should clear all the files in your team-folder. It should not modify or access other folders in the 'Battle Field' directory.
2. Initialize a variable i to 0
3. Increment i by 1
4. Wait in a loop for to wait for the file input_i .txt in your teamFolder. The loop should be as follows. Make sure that there is at least 10ms sleep in you while loop
    a. Write a file ready.txt in your team-folder. It could be empty (can have any content). Keep creating/overwriting the ready file in every iteration.
    b. Check if a file input_i.txt exist.
        i. If yes, wait for 200 milli seconds and go to step 5
        ii. If no, go to 4.c
    c. Check if a file Gameover.txt
        i. If yes, exit the application (For some reason the game is aborted). You can read to understand who won.
        ii. If no, go to 4.a and repeat steps
5. Read the contents of input_i.txt and get the following info
    a. It would have 56 lines.
    b. First line would tell what is expected

```
1  Expecting = K 29-49    What is expected. Here King position in 29-49
2  1 = A     // Player 1 name is A
3  2 = C     // Player 2 name is C    Use these info to know who is player1 and 2
4  PowerPlay1 = 2 //Read and ignore this. This would be 0
5  PowerPlay2 = 2 //Read and ignore this. This would be 0
6  KingMove1 = 6   // Number of King moves/swaps remaining for Player 1
7  KingMove2 = 6   // Number of King moves/swaps remaining for Player 2
```

    Expecting = <What> <Where>
    What could either be K or C. For K, where would be <startCell>-<endCell> to indicate where you can place the King
    c. 2nd and 3rd line would tell who first and second players are
    d. 4th and 5th line are unused (read and ignore)
    e. 6th and 7th line would tell how many king swaps are there for each player.
    f. From Line 8 would tell the board info in the following format
    <cellNumber> = <whoOwns>,<whatIsPresent>,<howMany>
    <cellNumber> will take values from 1-49
    <whoOwns> : 0 - No owner, 1 - Player 1, 2 - Player 2
    <whatIsPresent> : N - Empty, C - Coin, K - King
    <howMany> : 0 - 3

If both the Kings are hidden, for Player 1, his King will be shown like this. Say Player1 is King is in cell 5.

5 = 1, K, 0

and the same cell for player 2 will be shown like an empty cell as

5 = 0, N, 0

Once the Player 1 King is revealed, the King would be shown as 1, K, 0 for both the players.

6. Call your optimal Method to decide where to place - Your program should now decide where to place your coin. If the expected move is to place the King, you should return the position where you want to place your King in your side of board. If coin is expected, you can place your coin in an empty cell or in your own cell. You can also do a swap. **An Important point** - If you place the King in the wrong position in your first move, your King would be placed in the top-left cell (Cell 1) incase you are player1 or bottom-right corner (Cell 49) if you are second player. In addition, your King would be revealed. Opponent's King would not be revealed to you until you find it.
7. Write the board numbers and cell numbers to a file output_i.txt in your teamFolder in the following format. The angular brackets should not be written onto text file.
   For placing King - K <position>
   For placing Coin - C <position>
   For Swapping King - S <from>-<to>
   Example:
   - K 2
   - C 5
   - S 2-10

8. Wait in a while loop (Don't forget to sleep for at least 10ms every iteration)
   a. Check if a file acknowledge_i.txt exist.
      i. If yes, wait for 200 milli seconds and go to step 11 after exiting the while loop
      ii. If not, go to 8.a
9. Read the contents of acknowledge_i.txt. It would give a single error code. The meaning of errors code are explained below. Errorcode 0 implies that you have made a valid move and error code 100 means you placed on opponent's King.

   Error Codes = Information

   0 = No Error

   1 = Expected move is King, but King move is not given

   2 = Two lines is used while expecting for a King (It could also be an empty line)

   3 = Invalid King Cell value (not in the range expected) . Battle Field will automatically place the king in either cell 1 or cell 49 and make it visible

4 = Expected Move is Cell or Swap, but neither cell nor swap is found or found an extra line.

5 = Expected Move is cell or swap, More than 2 lines used for a turn (Not valid since power play is not there)

6 = Expected Move is cell or swap, but two lines are there in the output file

7 = Swap is used but no available swap for the player is 0

8 = Placed the coin on your own king

9 =Placed the coin on opponents cell.

10 = tried Swap, but the "from" cell is not king

11 = tried Swap, but the "to" cell is an king cell

12 = tried Swap, but the "from" is owned by Opponent

13 = tried Swap, but “to” is owned by opponent

16 = Placed on the opponent King cell (after finding)

20 = Time Out

100 = Found Opponent King

17 = Swap is used but the output format is wrong

10. Goto step 3 and wait for next input

Evaluation process

Toss

We will have toss between two teams competing. Whoever wins the toss will be the first player

Round 1

The registered 39 temas + our Random BOT (40 Team) would be split into 8 groups (Group A to Group H) with each group having 5 teams. Each team within a group will play against each other using SPL Dashboard. The winner would be decided by the winning conditions discussed above and points would be awarded. The top 2 teams within each group would be selected for next round.

The points table would look as follows. The ‘Moves’ is total moves played in every winning match.

Team Name	Played	Won	Moves	Points
Team 01	4	3	1	3
Team 02	4	.	.	2
Team 03	4	.	.	
Team 04	4			
Team 05	4			

In case of a tie in points, the following rules apply to resolve it

- Team with Maximum points would get the first preference
- Teams with lowest moves will be ranked higher
- If moves are also tie, the team that won when the 2 teams competed would get precedence
- If more than two teams are in tie in moves, a round robin among the teams will be conducted with changed settings

The sixteen teams from Round 1 would be split into 4 groups and round robin will be played in each group. Top two teams from each group would be selected.

	Round 1							
	Group 1	**Group 2**	**Group 3**	**Group 4**	**Group 5**	**Group 6**	**Group 7**	**Group 8**
Group 1st	A1	A2	A3	A4	A5	A6	A7	A8
Group 2nd	B1	B2	B3	B4	B5	B6	B7	B8

Round 2				
	Group 1	**Group 2**	**Group 3**	**Group 4**
Group Members	A1	B1	A5	B5
	B2	A2	B6	A6
	A3	B3	A7	B7
	B4	A4	B8	A8
Group 1st	C1	C2	C3	C4
Group 2nd	D1	D2	D3	D4

				Winner
Quarter Final 1	C1	vs	D3	E1
Quarter Final 2	D1	vs	C3	E2
Quarter Final 3	C2	vs	D4	E3
Quarter Final 4	D2	vs	C4	E4

				Winner	**Loser**
Semi Finals 1	E1	vs	E4	F1	G1
Semi Finals 2	E2	vs	E3	F2	G2

Third place play off	G1	vs	G2

Finals	F1	vs	F2					

Disclaimer :-)

All decisions made by organizing team is final. Do keep the fun and competitive spirit of the event high! Play Hard, Play Fair! Wish you all the very best and let the best code win!

Code Submission

First version should be submitted by 1900 Hrs IST. How to submit will be updated shortly

You can start submitting your final version of code as early as 5:00 AM Sept 27 2015. To be updated