Things not implemented:

- Graphical output only shows the blocks moving around (left, down, right, and drop). It does not accurately clear a row or show the correct level / score.

Run-Time Commands:

*****NOTE: all commands below check for invalid input*****

x(right / righ / rig / ri) OR x(left / lef) OR x(down / dow / do)

- moves the current (undropped) block 'x' times in direction specified
- default x = 1
- if block reaches the end of the board (in dir. specified), or another block it stops moving.

x(clockwise / clockwis / … / cl) OR x(counterclockwise / counterclockwis / … / co)

- rotates the current (undropped) block 'x' times in the direction specified.
- default = 1
- if block can't be rotated, due to being at the edge or being blocked by a block, it stays the same.

x(drop / dro / dr)

- drops 'x' number of blocks in the current players board
- If players board is full (ie. a new block can't spawn) ---> Player loses, can't play anymore

Rest of the commands implemented as stated in requirements.

Bonus Implementations:

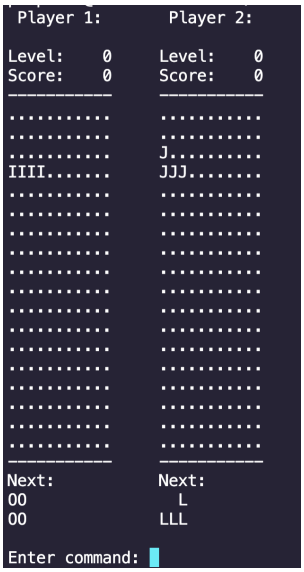
- No memory leaks + smart pointers
  Our program leaks no memory (except the small amount that is due to graphics).

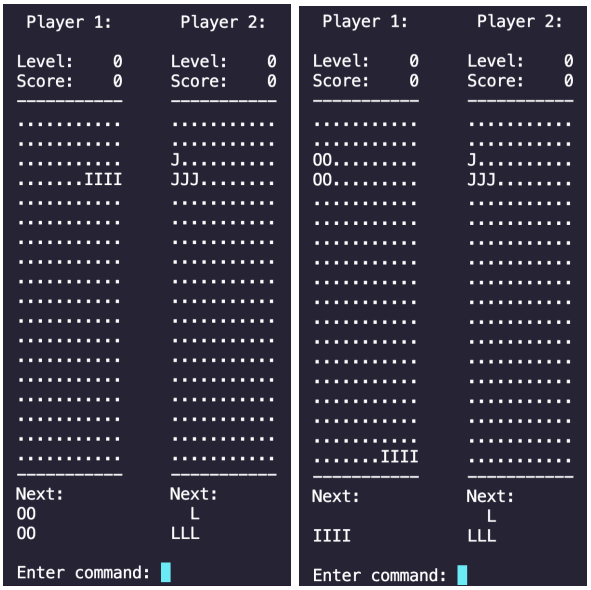
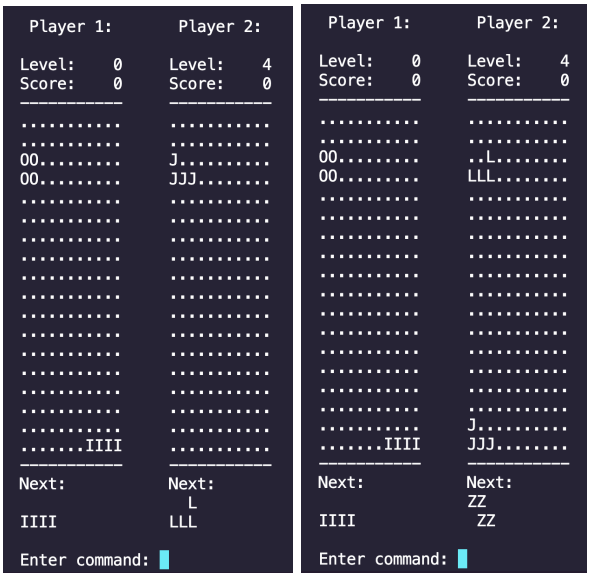
- Run time command: Quit
  Allows users to quit the program at any given time, after it displays the scores at that time.

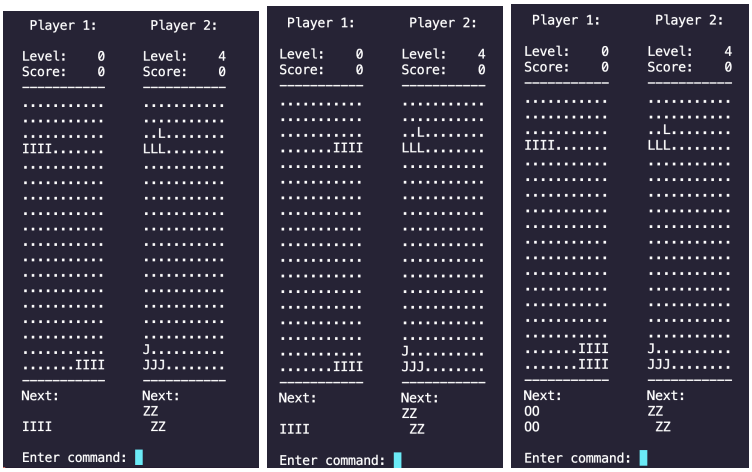
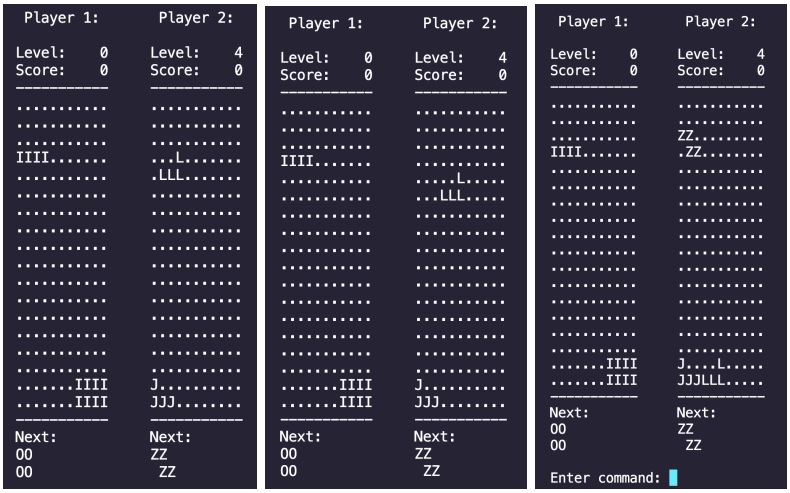
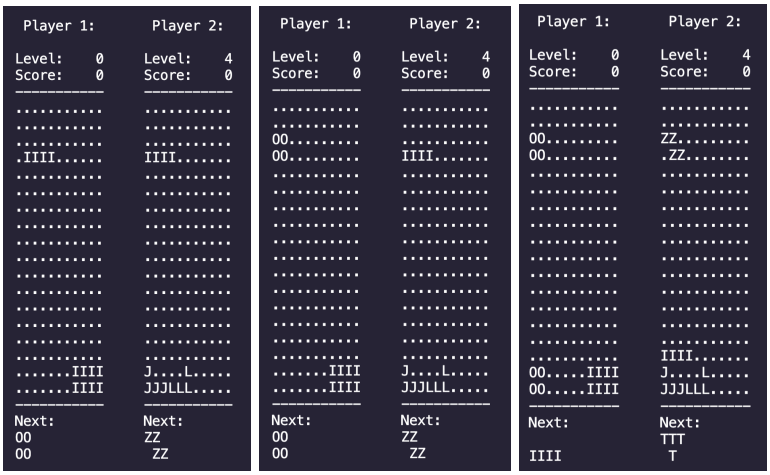
- Run time command: Rename
  Implementation of Q4 from project guideline, allowing users to change the command to call any of the run time commands.
  - Cannot rename any of the commands to 'rename'
  - Renamed commands must be entered in full (substrings will be invalid input)

Sample Games:

- Since graphical output isn't fully implemented, output shown here is text one. However, some features can be seen implemented on the graphical input as well!

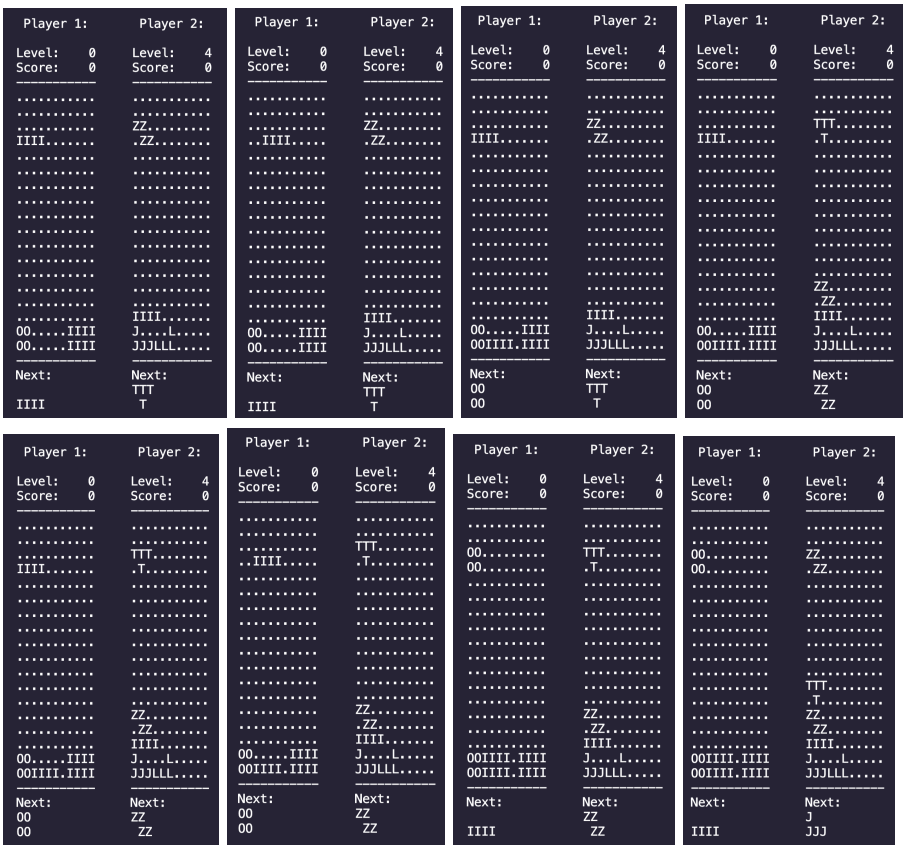
Game 1 command line:	`./biquadris –scriptfile1 sequence1.txt –scriptfile2 sequence2.txt`
Required file(s):	sequence1.txt sequence2.txt where: sequence1.txt has: I O sequence2.txt has: J L All commands listed in game1 can be found in order in game1.txt
Game starts	 <u>Now:</u> Player 1's turn

Entered: *10ri* then *dro*	 Moves until end of board is reached, then dropped it. <u>Now:</u> Player 2's turn
Entered: *4levelu* then *dr*	 Changed the level and dropped player 2's block. <u>Now:</u> Player 1's turn

Entered: *I* then *10ri* then *drop*	 Now: Player 2's turn
Entered: *ri* then *2ri* then *dro*	 Now: Player 1's turn
Entered: *O* then *dro* then *dro*	 Turn switches to player 2 after 2nd screenshot. Now: Player 1's turn

Entered

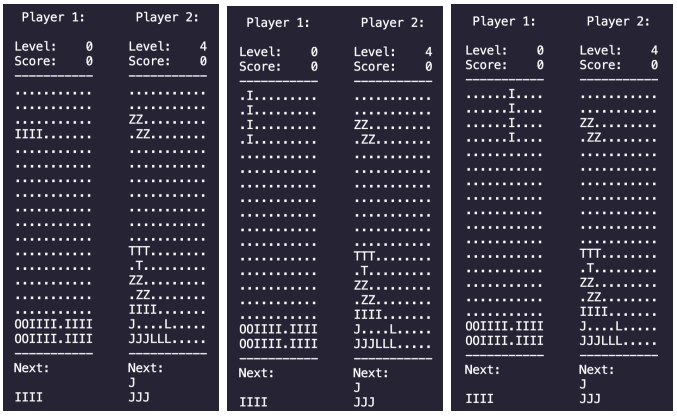
I then *2ri* then *dro* then *dro* x2

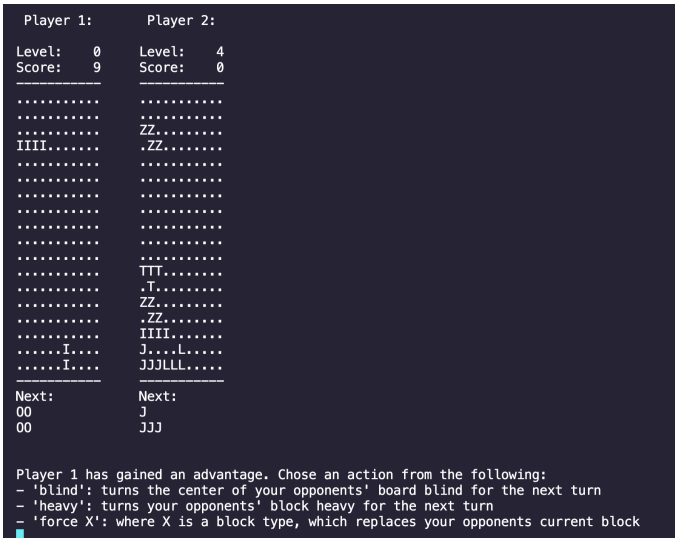
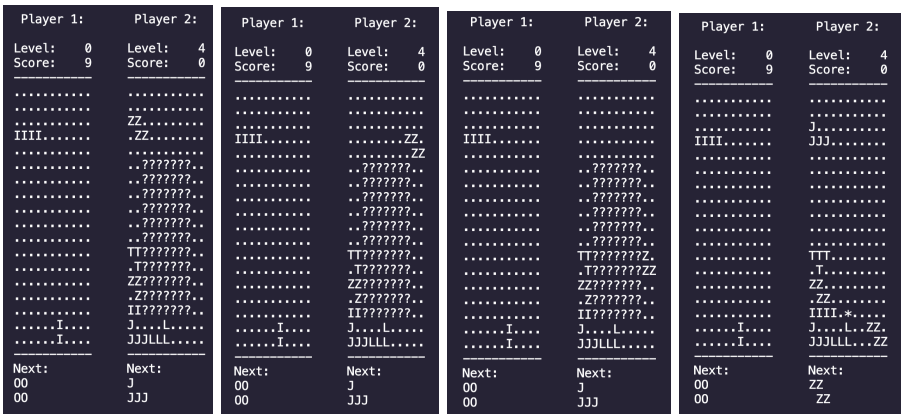
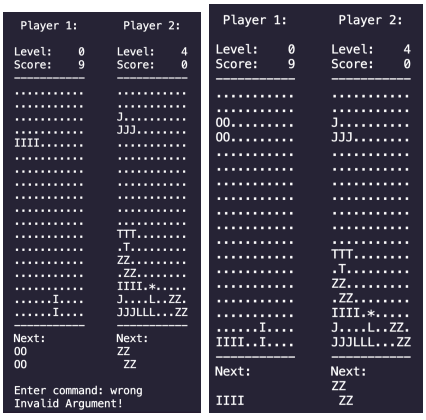


Now: Player 1's turn

Entered:

I then *cl* then *5ri* then *dro*



	 Player 1 cleared 2 rows thus they can choose a special disadvantage for player 2. Now: Player 2's turn (after player 1 chooses ability)
Entered: *blind* then *10ri* then *7dow* then *dr*	 Blind turns off after the current block is dropped. A star is dropped in the middle for player 2, as their in level 4. Also their block is heavy. Note: Both heavy and force are implemented properly!! Now: Player 1's turn
Entered: *wrong* then *dr* then *5dr*	

<table>
<tr>
<td></td>
<td>
<pre>
 Player 1:   Player 2:

Level: 0   Level: 4
Score: 9   Score: 0
----------   ----------
....
....
OO........   ....
OO........   IIII......
....   OO........
....   OO........
....   ZZ........
....   .ZZ.......
....   ZZ........
....   JZZ.......
....   JJJ.......
....   TTT.......
....   .T........
....   ZZ........
....   .ZZ..*....
....   IIII.*....
......I...   J....L..ZZ.
IIII..I...   JJJLLL...ZZ
----------   ----------
Next: Next:
                ZZ
IIII   ZZ

Player 2 has filled up their board. Player 1 can keep playing or call 'quit' to end the game.
Enter command:
</pre>
Since Player 2's board is full, they are out of the game. Player 1 can keep playing but since they are already ahead in the game, they choose to quit (bonus feature that displays winner + doesn't start new one).
</td>
</tr>
<tr>
<td>Entered:

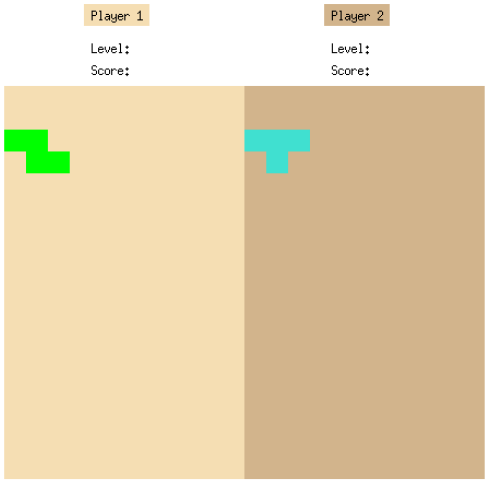
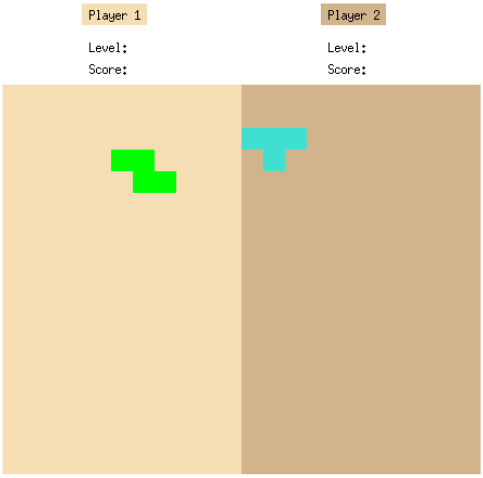
quit</td>
<td>
<pre>
Enter command: quit

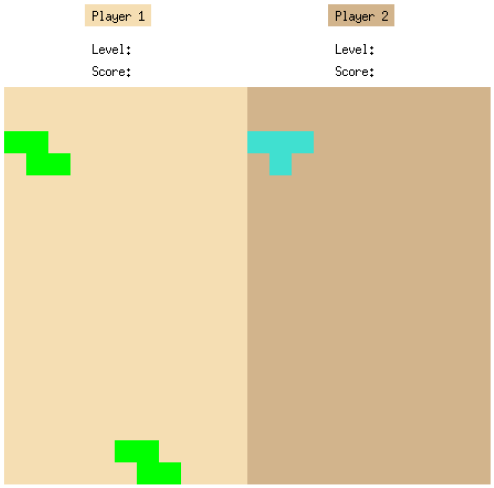
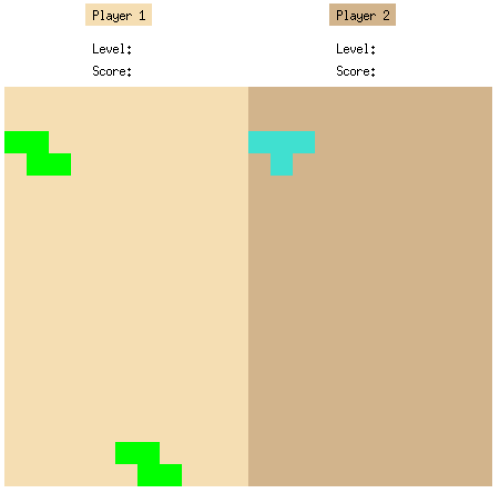
Final Scores
Player 1:   9
Player 2:   0

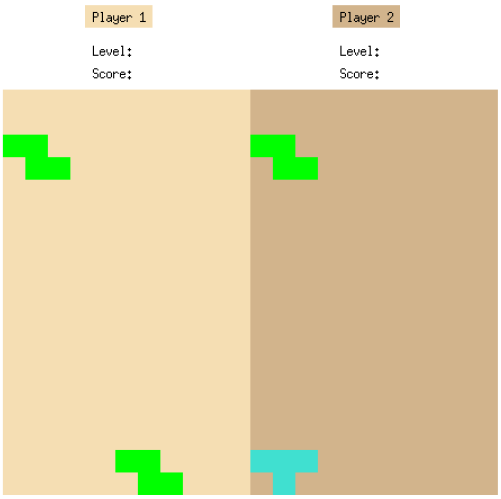
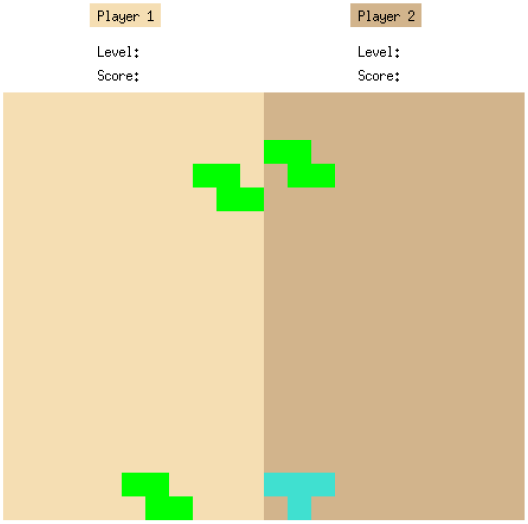
Player 1 wins!

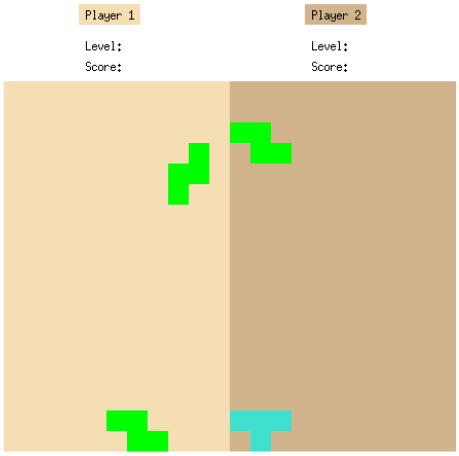
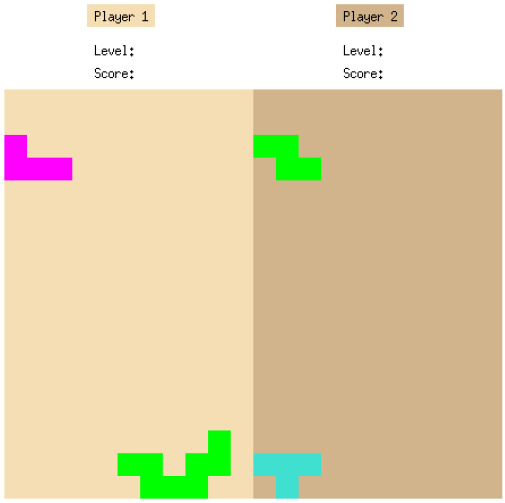
Highscore is: 9
Congratulations player1 for setting a high score

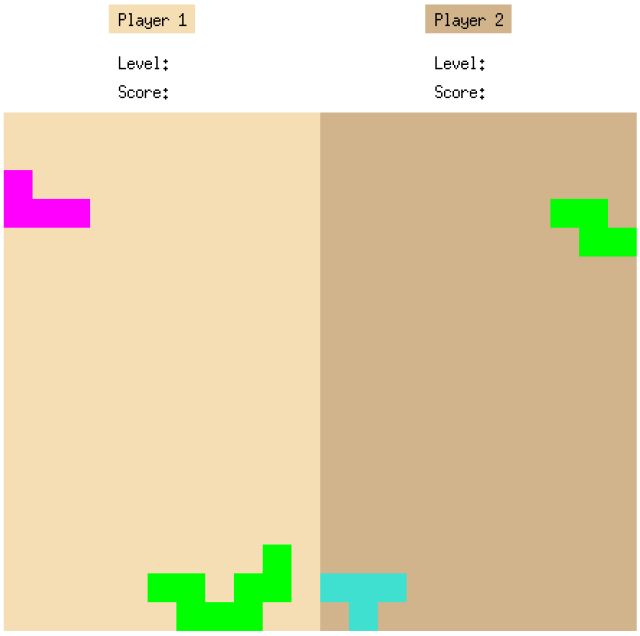
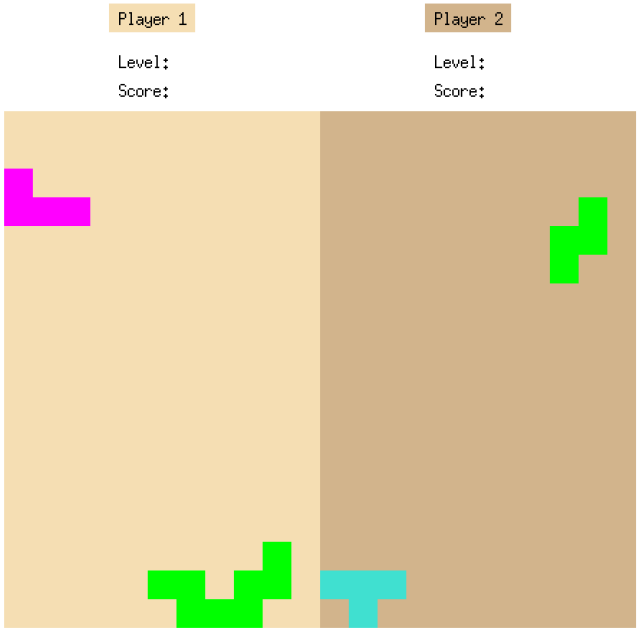
</pre>
Since player 1 chose to quit, rather then restart, the game displays final scores, winner and highscore then ends!
</td>
</tr>
<tr>
<td>Valgrind:</td>
<td>
With graphics:
<pre>
valgrind ./biquadris -scriptfile1 sequence1.txt -scriptfile2 sequence2.txt < game1.txt
</pre>
<pre>
==192297==
==192297== HEAP SUMMARY:
==192297==   in use at exit: 0 bytes in 0 blocks
==192297==   total heap usage: 33,266 allocs, 33,266 frees, 1,353,562 bytes allocated
==192297==
==192297== All heap blocks were freed -- no leaks are possible
==192297==
==192297== For lists of detected and suppressed errors, rerun with: -s
</pre>
Without graphics (-text option in command line):
<pre>
valgrind ./biquadris -scriptfile1 sequence1.txt -scriptfile2 sequence2.txt -text < game1.txt
</pre>
<pre>
==198064==
==198064== HEAP SUMMARY:
==198064==   in use at exit: 0 bytes in 0 blocks
==198064==   total heap usage: 29,430 allocs, 29,430 frees, 1,188,730 bytes allocated
==198064==
==198064== All heap blocks were freed -- no leaks are possible
==198064==
==198064== For lists of detected and suppressed errors, rerun with: -s
</pre>
</td>
</tr>
</table>

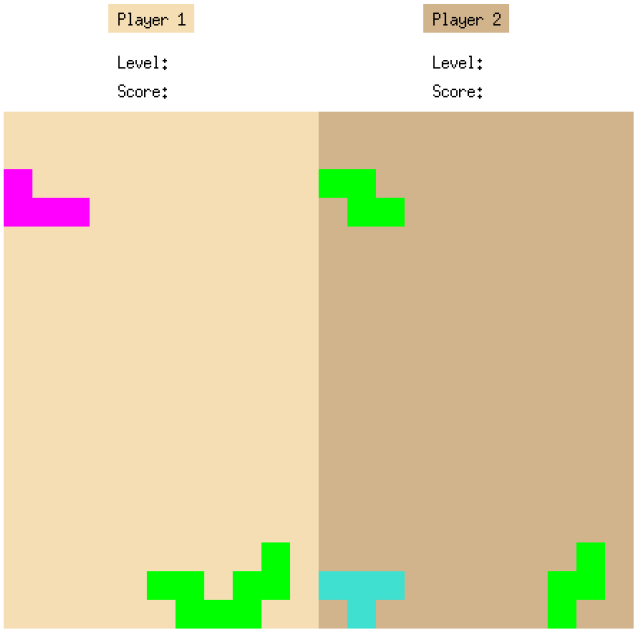
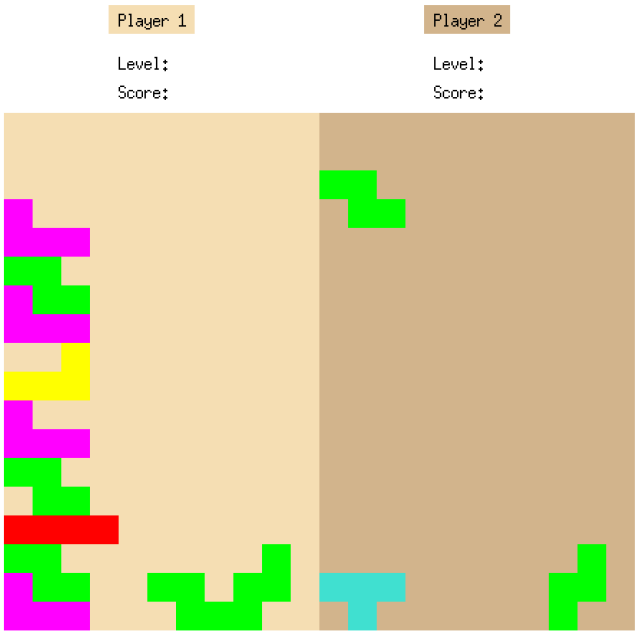
Game 2 command line:	`./biquadris -startlevel 4`
Game Starts	
Entered: *5ri*	

Entered: *dro*	
Entered: *2left*	

Entered: *dro*	
Entered: *8ri*	

Entered: *clo*	
Entered: *dro*	

Entered: *10ri*	
Entered: *count*	

Entered: *dro*	
Entered: *10dr*	

Entered:

10dr then *N*

```
 Player 1:       Player 2:

Level:    4     Level:    4
Score:    0     Score:    0
-----------     -----------
...........     ...........
...........     ...........
...........     ...........
J..........     IIII.......
JJJ........     .SS........
ZZ.........     SS.........
JZZ........     .SS........
JJJ........     SS.........
..L........     ZZ.........
LLL........     .ZZ........
J..........     ZZ.........
JJJ........     .ZZ........
ZZ.........     OO.........
.ZZ........     OO.........
IIII.*.....     ZZ.........
ZZ...*...Z.     .ZZ......Z.
JZZ..ZZ.ZZ.     TTT..*..ZZ.
JJJ...ZZZ..     .T...*..Z..
-----------     -----------
Next:           Next:
  L
LLL             IIII

Player 2 has filled up their board. Player 1 can keep playing or call 'quit' to
end the game.
---------------
Final Scores
Player 1:     0
Player 2:     0

It's a tie!
---------------
Highscore is: 0
---------------

Would you like to play again? (Y / N)
N
```

Valgrind:

Graphics:

```
valgrind ./biquadris -startlevel 4 < game2.txt
```

```
==23360== HEAP SUMMARY:
==23360==     in use at exit: 0 bytes in 0 blocks
==23360==   total heap usage: 29,692 allocs, 29,692 frees, 1,204,686 bytes allocated
==23360==
==23360== All heap blocks were freed -- no leaks are possible
==23360==
==23360== For lists of detected and suppressed errors, rerun with: -s
```

Without graphics:

```
valgrind ./biquadris -startlevel 4 -text < game2.txt
```

```
==27128==
==27128== HEAP SUMMARY:
==27128==     in use at exit: 0 bytes in 0 blocks
==27128==   total heap usage: 27,144 allocs, 27,144 frees, 1,071,214 bytes allocated
==27128==
==27128== All heap blocks were freed -- no leaks are possible
==27128==
==27128== For lists of detected and suppressed errors, rerun with: -s
```