CS246 Final Project, Demo

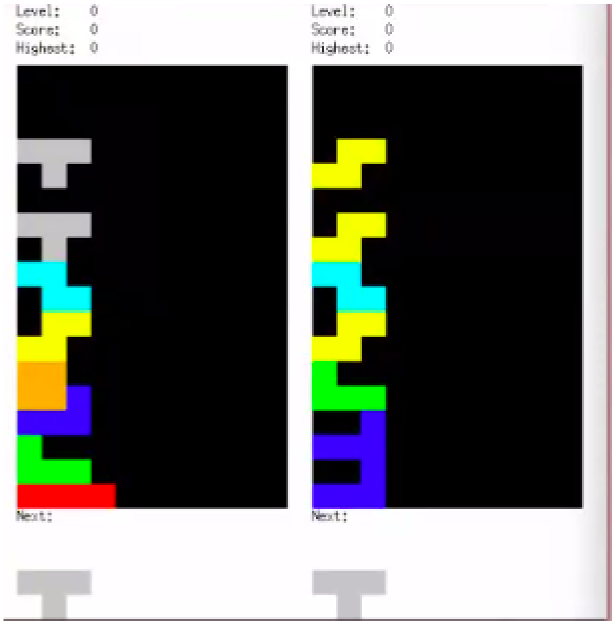
Written by Haoqi Zhang
Shengkang Zhang

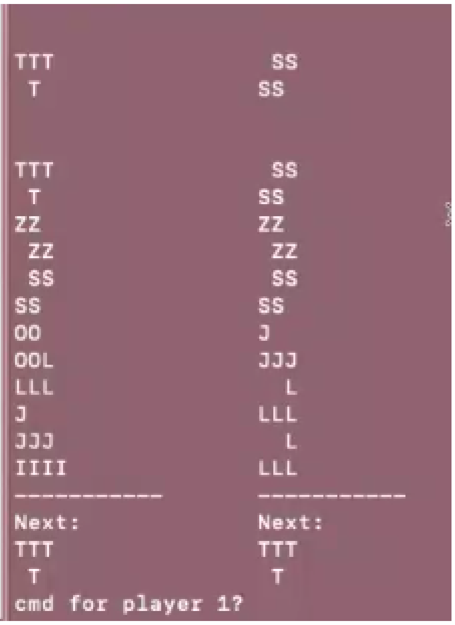
Index Page

View Model of Game

We have both text display and graphical display of the game. For both displays, they have two game boards of two players, and each board has 15 rows (additional 3 rows to show the block at initial position, so there are a total of 18 rows), and 11 columns, also showing the current score, current level (and highest score) of the game.
Here are the images of the two displays.

(graphical display) (text display)



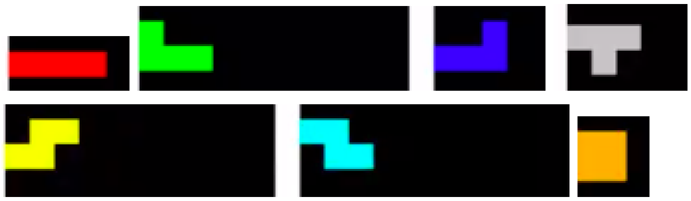


```
TTT                SS
 T                SS


TTT                SS
 T                SS
ZZ                ZZ
 ZZ                ZZ
 SS                SS
SS                SS
OO                J
OOL               JJJ
LLL                 L
J                 LLL
JJJ                 L
IIII              LLL
-----------       -----------
Next:             Next:
TTT               TTT
 T                 T
cmd for player 1?
```

Rules of Game

There are 7 different kinds of blocks, which are “I”, “J”, “L”, “T”, “S”, “Z”, “O”, and the images are given here.



The current block can move in the direction of left, right, or down and rotate clockwise or counterclockwise, just make sure the path of it when it changes does not filled by any other blocks and do not go out of boundary of the board, when the bottom of current block touches other block (or drop), the move stops, the block cannot move as an individual.

The score is calculated from 0, if any row and blocks are cleared, the calculation method is given below,
currScore += (number of lines cleared + level) * (number of lines cleared + level)
// if any blocks are cleared, then do below, otherwise, do not do below
currScore += (number of blocks cleared + level) * (number of blocks cleared + level)

When quitting the game, the player with the highest score will be the winner, and will be displayed with the highest score.

Available Command

Here are the available commands in the game:

- left
- right
- down
- drop
- clockwise
- counterclockwise
- levelup
- leveldown
- nonrandom file (not implement in out program)
- random (not implement in out program)
- sequence file
- I, J, L, etc
- restart
- blind
- heavy
- force

Here are the effects of the command given above:

- left moves the current block one cell to the left. If this is not possible (left edge of the board, or block in the way), the command has no effect.

- right as above, but to the right.

- down as above, but one cell downward.

- clockwise rotates the block 90 degrees clockwise, as described earlier. If the rotation cannot be accomplished without coming into contact with existing blocks, the command has no effect.

- counterclockwise as above, but counterclockwise.

- drop drops the current block. It is (in one step) moved downward as far as possible until it comes into contact with either the bottom of the board or a block. This command also triggers the next block to appear. Even if a block is already as far down as it can go (as a result of executing the down command), it still needs to be dropped in order to get the next block.

- levelup Increases the difficulty level of the game by one. The block showing as next still comes next, but subsequent blocks are generated using the new level. If there is no higher level, this command has no effect.

- leveldown Decreases the difficulty level of the game by one. The block showing as next still comes next, but subsequent blocks are generated using the new level. If there is no lower level, this command has no effect.

- norandom file Relevant only during levels 3 and 4, this command makes these levels non-random, instead taking input from the sequence file, starting from the beginning. This is to facilitate testing.

- random Relevant only during levels 3 and 4, this command restores randomness in these levels.

- sequence file Executes the sequence of commands found in file. This is to facilitate the construction of test cases.

- I, J, L, etc. Useful during testing, these commands replace the current undropped block with the stated block. Heaviness is determined by the level number. Note that, for heavy blocks, these commands do not cause a downward move.

- restart Clears the board and starts a new game.

- blind The player's board, from columns 3-9, and from rows 3-12, is covered with question marks (?), until the player drops a block; then the display reverts to normal.

- heavy Every time a player moves a block left or right, the block automatically falls by two rows, after the horizontal move. If it is not possible for the block to drop two rows, it is considered to be dropped, and the turn ends.

- force Change the opponent's current block to be one of the player's choosing. If the block cannot be placed in its initial position, the opponent loses. (E.g., force Z)

Available Level

The available levels and their effects for Biquadris are listed below:

- Level 0: (Make sure you at least get this one working!) Takes its blocks in sequence from the files sequence1.txt (for player 1) and sequence2.txt (for player 2) (samples are provided), or from other files, whose names are supplied on the command line. If you get to the end of one of these files, and the game hasn't ended yet, begin reading the file again from the beginning. This level is non-random, and can be used to test with a predetermined set of blocks. Make sure that sequence[1-2].txt, and any other sequence files you intend to use with your project, are submitted to Marmoset along with your code.
- Level 1: The block selector will randomly choose a block with probabilities skewed such that S and Z blocks are selected with probability each, and the other blocks are selected with probability each.
- Level 2: All blocks are selected with equal probability.
- Level 3: The block selector will randomly choose a block with probabilities skewed such that S and Z blocks are selected with probability each, and the other blocks are selected with probability each. Moreover, blocks generated in level 3 are "heavy": every command to move or rotate the block will be followed immediately and automatically by a downward move of one row (if possible).
- Level 4: In addition to the rules of Level 3, in Level 4 there is an external constructive force: every time you place 5 (and also 10, 15, etc.) blocks without clearing at least one row, a 1x1 block (indicated by * in text, and by the colour brown in graphics) is dropped onto your game board in the centre column. Once dropped, it acts like any other block: if it completes a row, the row disappears. So if you do not act quickly, these blocks will work to eventually split your screen in two, making the game difficult to play. (not implement in out program)

Command Line Arguments

Here are the available command line arguments for Biquadris:

- -text
- -seed xxx
- -scriptfile1 xxx
- -scriptfile2 xxx
- -startlevel n

- -text runs the program in text-only mode. No graphics are displayed. The default behaviour (no -text) is to show both text and graphics.
- -seed xxx sets the random number generator's seed to xxx. If you don't set the seed, you always get the same random sequence every time you run the program. It's good for testing, but not much fun.
- -scriptfile1 xxx Uses xxx instead of sequence1.txt as a source of blocks for level 0, for player 1.
- -scriptfile2 xxx Uses xxx instead of sequence2.txt as a source of blocks for level 0, for player 2.
- -startlevel n Starts the game at level n. The game starts in level 0 if this option is not supplied.

Sample Test Cases

1. left, right, down, drop
   Input file: sequence1.txt sequence2.txt
   Command: 6left, 7right, 10down, drop

2. rotation
   Input file: sequence1.txt sequence2.txt
   Command: cw, cw, cw, cw, drop, ccw, ccw, ccw, ccw, drop

3. clear the rows (one or more)

   Only 1 row:
   Input file: sequence1.txt sequence2.txt
   Command: drop, drop, drop, 3right, drop, drop, 6right, drop, drop, drop, drop, 9right, drop
   Multiple row and blocks:
   Input file: sequence3.txt sequence2.txt (that is -scriptfile1 sequence3.txt)
   Command: drop, drop, drop, drop, 4right, drop, drop, 6right, drop, drop, 6right, drop, drop, clockwise, 10right, drop

4. special action

   Blind:
   Input file: sequence3.txt sequence2.txt
   Command: drop, drop, drop, drop, 4right, drop, drop, 6right, drop, drop, 6right, drop, drop, clockwise, 10right, drop, blind, 4right, drop, drop, drop

   Heavy:
   Input file: sequence3.txt sequence2.txt
   Command: drop, drop, drop, drop, 4right, drop, drop, 6right, drop, drop, 6right, drop, drop, clockwise, 10right, drop, heavy, 4right, drop, drop, drop

   Force:
   Input file: sequence3.txt sequence2.txt
   Command: drop, drop, drop, drop, 4right, drop, drop, 6right, drop, drop, 6right, drop, drop, clockwise, 10right, drop, force, T, 4right, drop, drop, drop