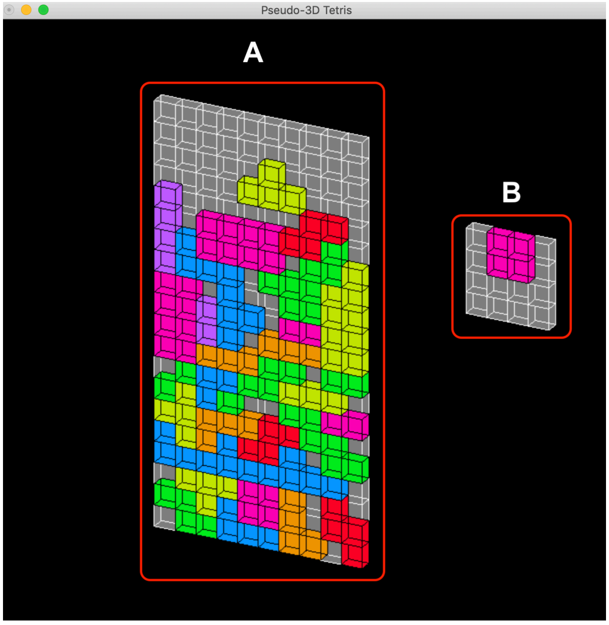
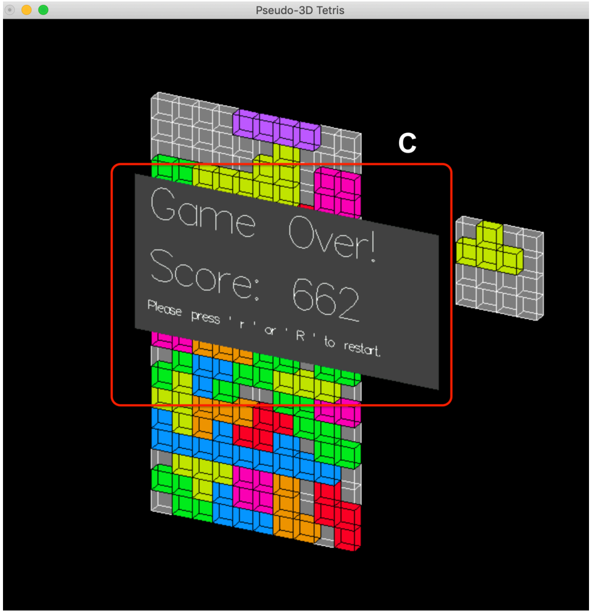
Features:

- **A** is the Tetris play board
- **B** is the Piece Look-ahead
- **C** is the Game over prompt





Game Over!
Score: 662
Please press ' r ' or ' R ' to restart.

- The game implement it's rotation according to Super Rotation System (SRS)
- The controllable Tetromino simply does not rotate when it kicks the wall or it's rotation will collide into already-landed Tetrominos.
- The game implement a naive method for line clearing (same as the classic Game Boy edition and most of other Tetris.)
- The game generates random tetrominos block with the time seed.

Main design aspects:

- Use different colours on the wire of unoccupied and existing Tetrominos. Hence clearer to distinguish it from the background play board.

- Avoiding colours that are too intense to soothe our eyes when playing the game.

- Draw cubes from left to right and bottom to top to make it looks more 3D.

How specific OpenGL / GLSL features are employed

- 3D Cube:
  - Created and store the cube into the list.
  - Draw the 3D solid cube first then uses *GL_LINE_LOOP* to add the wire to make it distinguishable.

- Tetris play board / Grid:
  - Since I am essentially calling 3D cubes, having it from left to right and bottom to top will make it looks more 3D, avoiding overlapping of 3D cubes.
  - Uses *glTranslatef* to make sure each elements are at the right place.

- Auto falling mechanism*:*
  - Assigns a difficulty value which the user could alter and affects the delay time when the function calls itself again through *glutTimerFunc().*

- Control Mechanism:
  - Uses *glutKeyboardFunc()* and *glutSpecialFunc()* to keyboard control. Also, *glTranslatef()* is used to move the controllable Tetromino and it was redrawn every time when the user moves it.

- GameOver Prompt:
  - The prompt is drawn using *GL_QUADS* and the score and reminding text were drawn after *glTranslatef().* Due to the fact that score user attained could have different digit and might alter the location of the reminder text, a mitigating mechanism has been implemented to make sure texts are not out of places.
  - Note that *glutStrokeCharacter()* is also used to draw the text in the prompt.

How to compile and run it

Simply run the make command and run the executable file.

If Symbolic link created:

```
-> make Tetris
-> ./Tetris
```

If Symbolic link not yet created:

```
-> make -f Makefile.linux Tetris
```

How to use it

The following keys will be used to control the game:

↑ : Rotate the controlling Tetromino according to SRS.

← : Move the controlling Tetromino one block to the left.

→ : Move the controlling Tetromino one block to the left.

↓ : Move the controlling Tetromino one block to the bottom.

“Spacebar”: Make the controlling Tetromino falls instantly to the bottom.

“R”: Restart the game and clear the Tetris board.

“r”: Restart the game and clear the Tetris board.

Difficulty Controlling keys:

“1” : Set the difficulty to 1.
“2” : Set the difficulty to 2.
“3” : Set the difficulty to 3.
“4” : Set the difficulty to 4.
“5” : Set the difficulty to 5.
“6” : Set the difficulty to 6.
“7” : Set the difficulty to 7.
“8” : Set the difficulty to 8.
“9” : Set the difficulty to 9.

(Increasing the difficulty increases the falling speed of the controllable Tetromino.)

Word Count: 516