

# CS324 COMPUTER GRAPHICS COURSEWORK

## - TETRIS

Iesa Wazeer  
University of Warwick DCS  
u1831114  
Coventry, United Kingdom  
Iesa.Wazeer@Warwick.ac.uk

### I. FEATURES

#### A. Piece Movement

All the basic movement, such as "drop", "move left", "move right" was included immediately and then extended to work with a "soft drop" and "hard drop". A soft drop is when the Tetromino falls faster whereas a hard drop is when it is instantly placed.

#### B. Rotation

Clockwise and Counterclockwise piece rotation was included and we extended the code to make them work with "Wall kicks" as well. A wall kick occurs when a player tries to rotate a piece and the rotated position is obstructed by other pieces. The game will attempt to "kick" the position into an alternative position nearby [3].

#### C. Hold & Next Piece

To better mimic the game of Tetris we included the ability to hold a piece and to see what the next piece would be.

#### D. Levels and Scoring

We created a scoring system which mimics the original Tetris game where the score is based on the number of lines cleared and the current difficulty level [4]. The equation we used was:

$$40 * (n + 1)100 * (n + 1)300 * (n + 1)1200 * (n + 1)$$

where n is the difficulty level starting from 0 and the level increases every 10 lines cleared.

#### E. Pause and Reset

Finally, I added the ability to pause the game and reset the game board through keyboard buttons. The game can only be reset once the game is over, which is when no more pieces can be placed. The player can pause whenever they want, as long as the game isn't over.

### II. MAIN DESIGN ASPECTS

We split each component of the code into different classes, making our code Object Oriented. This decision was inspired by examples of Tetris online such as [1] and [2].

- The Square class display an individual Square which we defined as 30 pixels.
- These Squares were used in the Tetrominos class which display, move and rotate a Tetromino.
- The Board class was used to draw the Board, check if any piece has collisions with the board, store the piece, keep track of lines cleared and update score and difficulty accordingly.
- The Game class brought everything together, where all the classes work together to create a game. This is where I created the keyboard logic, displayed everything and the reset function for when the game ends.
- The Tetris class is where we initialise glut, create the shaders and call a Game instance.

The PieceInfo and Shader\_utility header files were used to modularise the code more, where the data in PieceInfo is called inside Tetrominos and the functions in Shader\_utility are called in Tetris

### III. OPENGGL AND GLSL FEATURES

#### A. OpenGL

OpenGL was used throughout the project as specified in the labs. We used glut to initialise a window, display on it, accept keyboard commands, run an idle function etc. In addition to using the standard functions provided in the glut library, I used an extension library called freeglut which was useful for displaying text on the screen. This was used to show the level, score and all other text which was drawn on the window.

#### B. GLSL

GLSL was used to make the pieces have a pseudo-3d look as per the specification given in the coursework sheet. We created a fragment shader which made the top and left 10 % brighter and to make the bottom and right 10% darker. Shaders were used when drawing blocks but not when writing text on to the screen. This meant that we had to constantly switch from using our shader to not using it. We used GLSL 1.2 when creating our shaders.

#### IV. COMPILING AND RUNNING

Using the example in the labs, I created my own makefile which compiles and creates a runnable program name "Tetris". To create the program use, once in the main directory run the command:

```
$ make
```

to run the file run:

```
$ ./Tetris
```

and to remove the Tetris file run the command:

```
$ make clean
```

#### V. HOW TO USE

Below is a list of commands used to play the game:

- Move left: 'a' key
- Move right: 'd' key
- Soft drop: activates while holding 's' key
- Hard drop: 'spacebar' key
- Hold piece: 'c' key
- Rotate clockwise: 'e' key
- Rotate counter-clockwise: 'q' key
- Pause: 'tab' key
- Restart (only when the game ends): 'k' key
- Exit program: 'x' key

Word Count (excluding title, references and word count text): 746 words

#### REFERENCES

- [1] <https://github.com/xhacker/tetris-opengl>
- [2] <https://levelup.gitconnected.com/writing-tetris-in-python-2a16bddb5318>
- [3] <https://harddrop.com/wiki/SRS>
- [4] <https://tetris.fandom.com/wiki/Scoring>
- [5] <https://moodle.warwick.ac.uk/course/view.php?id=39423>