Student ID: 1222089

CS324 Computer Graphics Coursework Report

Project: Tetris

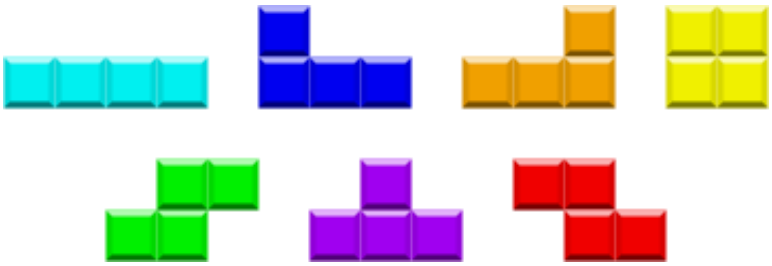
1. Introduction

1.1 History

Tetris is a classic video game created by Alexey Pajitnov in 1984[1]. The game was originally developed to test the capabilities of new hardware however it went on to sell over 170 million copies as of 2010[2].

1.2 Basic Rules

Tetris involves a board with the height of 20 tiles and the width of 10 tiles. From the top of this board drop random pieces of the following shapes named Tetriminos[3]:



A tetrimino falls one tile at a regular interval. Every time a line is filled with tiles it is erased. Once a tetrimino does not fit on the board when placed, the game ends.

A tetrimino can be manipulated in the following ways: left and right rotation, “soft drop” (moves one tile down on key press), “hard drop” (the piece falls all the way down the Y axis and locked) and left and right movement of the piece. Pieces can also be stored and swapped for the previously stored tetrimino or replaced by the next random tetrimino if this operation had not been performed before. This operation however can only be performed once per tetrimino.

1.3 Rules used in this project

There have been many versions of Tetris over the years.
Rather than following an existing Tetris guideline the project was developed from scratch using the a combination of rules from various versions. The focus of the project

[1] http://en.wikipedia.org/wiki/Tetris
[2] http://web.archive.org/web/20131107133844/http://top100.ign.com/2007/ign_top_game_2.html
[3] http://tetris.wikia.com/wiki/Tetromino

was on using OpenGL for graphics rendering. The following rules were used in addition to the ones mentioned in section 1.2:

- The player can see the next piece that will follow.
- There are 20 levels of speed with which a tetrimino can fall. These speeds are hard coded using the Gameboy version of the game as reference[4]
- The player can define the start level as well as the end level through the settings menu (default is 0 and 20).
- The level is incremented every 10 lines cleared.
- Pieces are randomized using C++ library function using the time for seed.
- If after a rotation the piece does not fit on the board a brute force algorithm attempts to reposition the piece. If the piece does not fit on the board the game ends.
- Scoring is defined as follows:
    - For each tile of a “hard drop”: 2 points
    - For each tile of a “soft drop”: 1 point
    - Using a single piece:
        - 1 Line cleared: 40 points
        - 2 Lines cleared: 100 points
        - 3 Lines cleared: 300 points
        - 4 Lines cleared: 1200 points
- Finally a ghost piece will show the spot where a tetrimino will lock using soft drop, hard drop or simply letting it free fall.

2. Features

- Tetris game mechanics as outlined in the previous section
    - Soft Drop + Hard Drop
    - Left + Right rotation,
    - Line clearing
    - Ability to "Hold" a Piece
    - Scoring system
    - 20 Levels of Difficulty
    - Piece Look ahead
    - Ghost Piece
- 2D menu complete with control instructions and mouse controlled settings.
- FPS counter (disabled by default)
- Multiple rendering Options
    - 3D
    - ASCII (text)
    - 2D Vector
- Enable/Disable “Ghost Piece”
- Can turn textures on /off for 2D vector render

[4] http://tetris.wikia.com/wiki/Tetris_%28Game_Boy%29

3. Design and Implementation

The code is split among the following files:

- tetrisGL.cpp:
  Contains the main function along with subroutine declarations, and utility classes (Menu Item, Piece(tetrimino) and LinkedNode). All global variables are also declared in this file along with their initializations in appropriate functions.

  Finally this file contains the callbacks passed to GLUT for display.

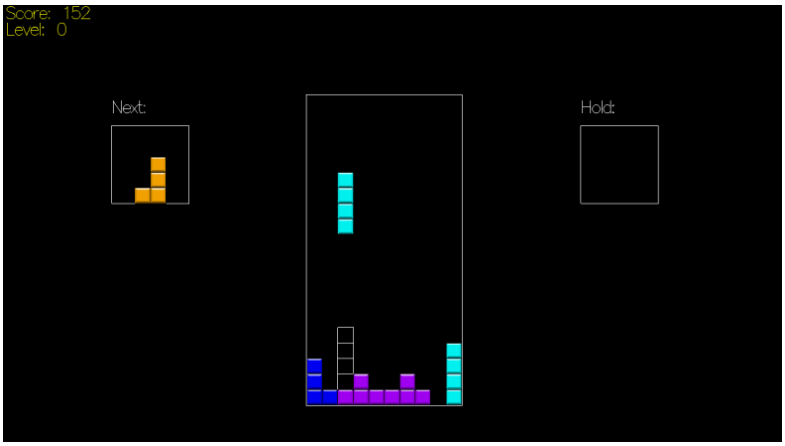
- util.h:
  Contains utility functions such as image loading (for textures) and dynamic memory manipulation. The function cleanup which releases program allocated memory is called whenever the user exits the program using a keystroke or by pressing Exit within the game menu.

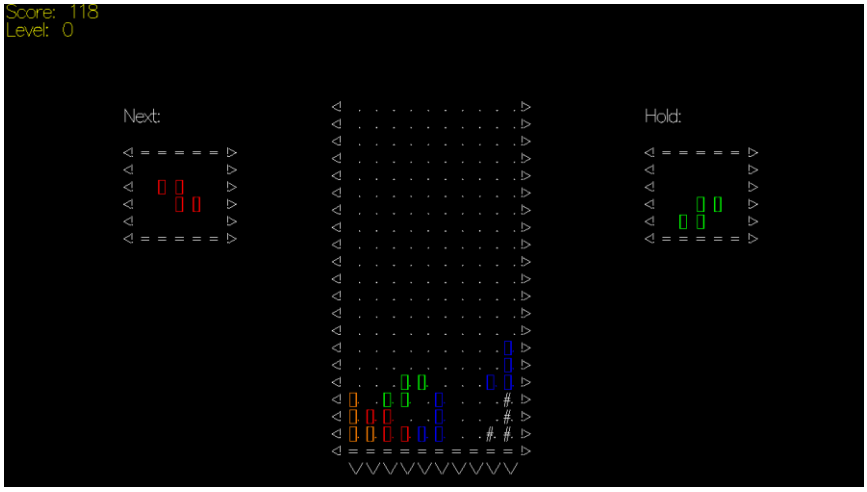
- input.h:
  Contains functions for mouse and keyboard input. The functions were designed in such a way as to map keys to functions and keep the amount of processing to a minimum in this file.

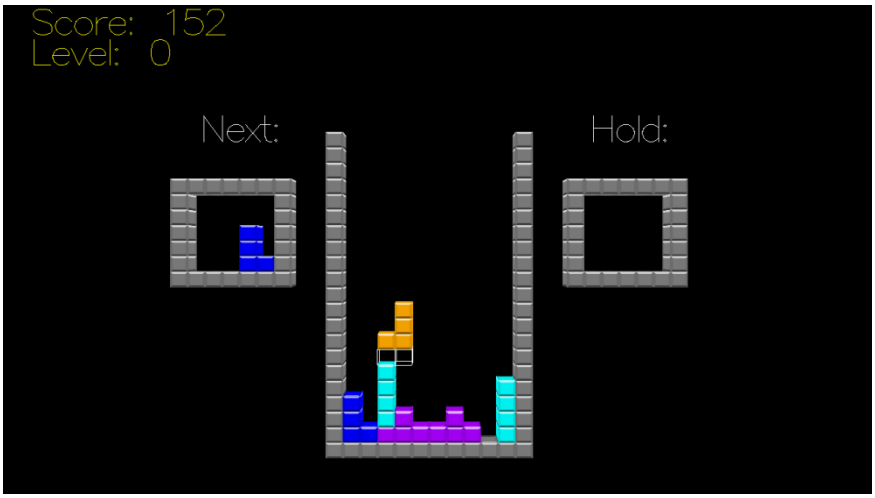
- control.h:
  Game control interface. Contains functions which modify the settings such as textures and rendering options. Also contains Menu actions like exit or restart game.

- game.h:
  This file contains all the game logic functions such as piece rotation or random tetrimino generator.

- draw.h:
  All the drawing functions are contained here. “draw.h” includes simple primitive drawing (squares and triangle) as well as functions for the different rendering options such as 3D, 2D Vector and ASCII.







4. Compiling and Running

To compile the program simply extract the zip file, navigate to the directory containing the files tetrisGL.cpp and the Makefile using bash and type the command make.

To run the program type *./tetrisGL* in bash, maintaining the same directory.

5. References

- Control Instructions page was constructed using this picture for the keyboard buttons: http://attosoft.info/blog/keyboard-keys/
- Tetriminos textures: http://upload.wikimedia.org/wikipedia/commons/3/39/Tetrominoes_IJLO_STZ_Worlds.svg
- Conversion from number to text: http://www.cplusplus.com/articles/D9j2Nwbp/

Word Count: 819