NORTHWESTERN UNIVERSITY

Super Duper Tetris Fun Time

A Java Game by Group Delta

Xiying Deng, Boda Du, Cyrus Forbes, Kenny Franco,
Aaron Goldblum, and Kanut Harichanwong

June 9, 2014

1 PROJECT OVERVIEW

For team Delta's project this quarter, we created a Tetris game that two people can play simultaneously on the same computer in both a cooperatively and competitively mode. One person uses the arrow keys and another uses "WASD" to control the two boards. The cooperative mode has the two players working together to get the high score. This includes the same line must be complete on both boards for it to disappear and the two players can switch their respective held blocks to help each other out. In the competitive mode, the players will be trying to stay alive the longest while the opponent works against this. By spending the score accumulated from removing lines, a player can drop pieces or create an unfilled line on their opponent?s board. The high score, or quick start version, is easier since that just includes two stand alone games without much interaction besides sharing a screen.
For this project, we first dealt with creating a working Tetris game before creating two simultaneous ones. We used a grid as the board with squares colored in as pieces. We thought this method would be easier than creating objects that we have to manipulate in space. Also by doing this, we will not have to disassemble pre-designed objects when a row is eliminated. Our original thought was to use threading to allow the two players to play simultaneously but after testing we determined that a single program would work with minimal discrepancies. This project was a lot of work, but Group Delta was up for the challenge.

This a Super Duper Tetris, there are three models for players to chose, quick start, competitive, cooperative.

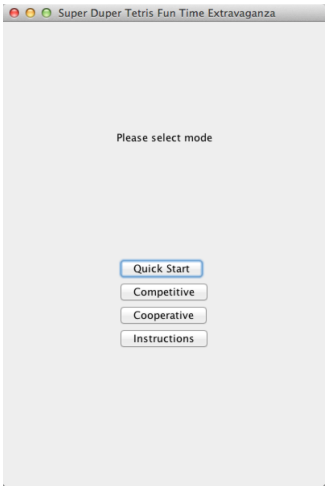

Figure 1.1: The homepage of the game.

1.1 Quick Start

This mode is just normal Tetris. Outlast Opponent to win. The blocks of two players are different.

Player One Controls: On Left Side

W = rotate piece
A = move piece left
D = move piece right
S = move piece down
LEFT SHIFT = save current piece / swaps held piece

Player Two Controls: On Right Side

Player Two Controls: On Right Side
UP = rotate piece
LEFT = move piece left
RIGHT = move piece right
DOWN = move piece down
RIGHT SHIFT = save current piece / swaps held piece

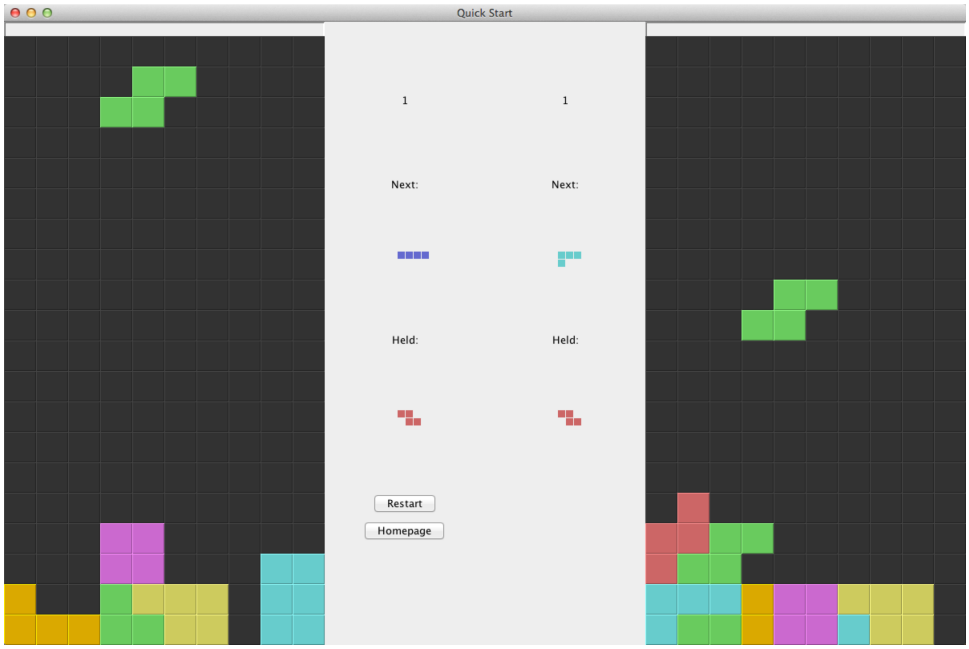

Figure 1.2: The Quick Start model of the game.

1.2 Competitive

This mode allows you to mess with your opponents game. Simply rack up points to either send a line up or make the opponents piece drop for two points each. Score is based on how many lines you have completed. There are twenty pieces per round and first to go through all the pieces is saved from getting a line pushed up to their board. The blocks get faster as rounds progress.

Player One Controls: On Left Side

W = rotate piece
A = move piece left
D = move piece right
S = move piece down
LEFT SHIFT = save current piece / swaps held piece
1 = send line to opponent
2 = drops opponent's current piece

Player Two Controls: On Right Side

Player Two Controls: On Right Side
UP = rotate piece

LEFT = move piece left
RIGHT = move piece right
DOWN = move piece down
RIGHT SHIFT = save current piece / swaps held piece
, = send line to opponent
. = drops opponent's current piece

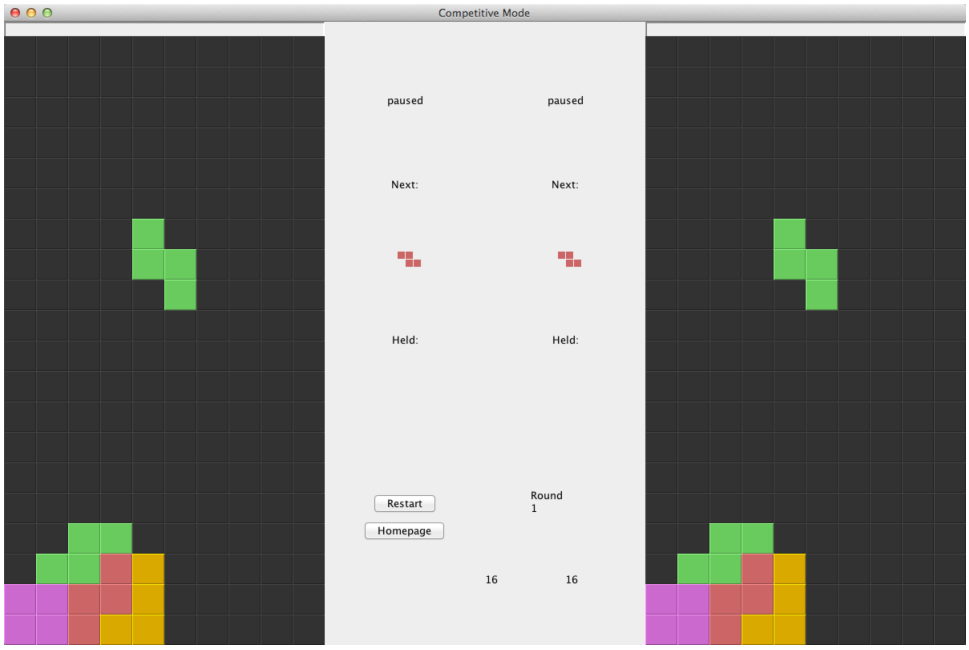

Figure 1.3: The Competitive model of the game.

1.3 Cooperative

This mode requires team work. Lines will not dissappear unless both players have the same lines completed.

Player One Controls: On Left Side

W = rotate piece
A = move piece left
D = move piece right
S = move piece down
LEFT SHIFT = save current piece / swaps held piece
ENTER = Switch held pieces between players

Player Two Controls: On Right Side

Player Two Controls: On Right Side
UP = rotate piece
LEFT = move piece left
RIGHT = move piece right
DOWN = move piece down
RIGHT SHIFT = save current piece / swaps held piece
ENTER = Switch held pieces between players

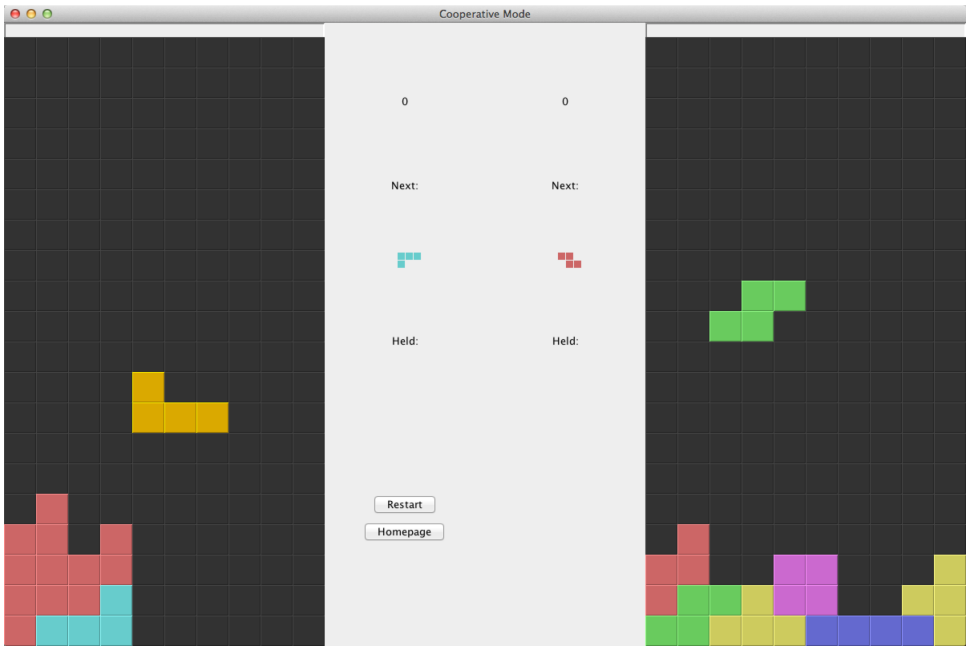

Figure 1.4: One player has one full line couldn't make that line disappear in cooperative model.

1.4 INSTRUCTIONS

If you want know how to play this game, you can use the "Instructions" button to get the instructions.

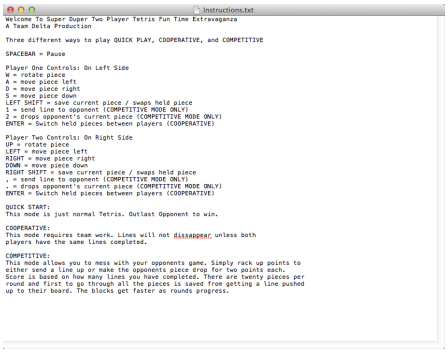

Figure 1.5: The instructions of this game.

2 Class Overview

2.1 SwingControlDemo

Overview

SwingControlDemo is the screen that first appears when the game is launched. It allows the user to choose between Cooperative mode and Competitive mode. It also provides the user with instructions for the game.

Important Methods

Constructor:
Calls prepareGUI() method.

private void prepareGUI():
Prepares the GUI that allows the user to choose between Cooperative and Competitive. It creates a new JFrame, sets the JFrame parameters, and creates all the labels and the Competitive/Cooperative/Instructions buttons.

public void actionPerformed(ActionEvent e):
When either the Competitive, Cooperative, or Instructions button is pressed, this method is called. Based on which button was pressed, this method creates a new Competitive or Cooperative object, or prints the instructions for the game.

2.2 Tetris

Overview

Tetris is the superclass to the Competitive and Cooperative classes. It provides the basic methods and properties that both Competitive and Cooperative share. The Tetris class controls the entire game. It contains two Board objects (the Board class will be explained later in the report) and renders the score in between the two Board objects.

Important Methods

Constructor:
Creates a createpiece object, which holds the data for the next tetris piece. Creates a new JFrame that holds the two Board objects and the score. Separates the JFrame into 3 columns using GridLayout: first is player 1?s board, second is the score, and third is player 2?s board. Creates all the labels and buttons for the score panel. Finally, starts the two Board objects using board.start() and board2.start().

public void keyPressed(KeyEvent e):
Controls actions when a key is pressed. Calls methods that move/rotate pieces, hold pieces, and all other keyboard actions.

2.3 COMPETITIVE

Overview

A subclass of Tetris. Overrides key methods so that the game can run in competitive mode.

Important Methods

Constructor:
First calls the superclass constructor, then changes the title of the JFrame to ?Competitive Mode?.

public void actionPerformed(ActionEvent e):
When the Restart or Homepage button is pressed, this method is called. If Restart is pressed, disposes the current JFrame and creates a new Competitive object. If Homepage is pressed, disposes the current JFrame and creates a new SwingConrolDemo object, taking the user back to the start page.

public void keyPressed(KeyEvent e):
Overrides certain key presses since certain keys have different actions in Cooperative and Competitive.

2.4 CREATEPIECE

Overview

Holds an list of Shape objects in order. Used by the Board object so that both player 1 and player 2?s boards have the same order of pieces .

Important Methods Constructor:
Creates an array of length 21 and populates the array with 21 random Shapes.

2.5 BOARD

Overview

The meat of the program. The Board objects control the flow of the each players tetris game, including moving/rotating pieces, checking for full rows, drawing the board, and controlling each player?s score.

Important Methods

Constructor:
When a Board object is created, the parent Tetris object, the player number, the JPanel, and the mode are passed in. The constructor then populates the properties of the Board object using these values. Creates objects for current piece, next piece, and held piece. Creates a timer that the game runs on - every tick calls the actionPerformed() method.

public void actionPerformed(ActionEvent e):
Called every time the timer ticks. Checks if a piece is currently falling - if so, it advances the piece down one line. If the piece is done falling, it creates a new piece.

public void start():
Starts the timer and begins the game.

public void pause():
Pauses the game if the game is running. Unpauses the game if the game is paused.

public void stop():
Ends the game by stopping the timer and clearing the board.

public void paint(Graphics g):
When called, this method first clears the board, then draws the board. It loops through every piece in the board and calls the pieces paint function, which draws the piece at the given coordinates.

public void drawLose(Graphics g):
This method is called when the specific player loses. It displays the lose screen over the players board.

public void drawWin(Graphics g):
This method is called when the specific player wins. It displays the win screen over the players board.

boolean tryMove(Shape newPiece, int newX, int newY):
Attempts to move the Shape to a new location. Returns true if the new location is unoccupied or a valid location. Returns false if the new location is occupied or an invalid location.

void oneLineDown():
Calls the tryMove() method on the current piece and tries to move it down one line.

private void clearBoard():
Removes all pieces from the board.

private void pieceDropped():
Drops the current piece to the bottom and sets falling to false so that the newPiece() method will be called.

private void newPiece():
Spawns a new piece at the top of the board. The new piece?s shape is taken from the list of pieces generated at the beginning of the game in the Tetris object.

private void removeFullLines():
Check each row to see if it is full. If the mode is Competitive and the row is full, remove it. If the row is full and the mode is Cooperative, check to see if the equivalent row in the other players Board is full and if they are both full, remove them both. When the row is removed, drop every piece above it by one block.

public void hold():
Swaps the current piece with the hold piece. If the hold piece doesn?t exist, add the current piece to the hold piece and call the newPiece() method.

private void drawSquare(Graphics g, int x, int y, Tetrominoes shape):
Used in the paint() method. Pass in a x location, y location, and the shape to draw, and this method will draw the shape at the given location.

2.6 JAVADOC

We have published our java doc on the Internet, you could get the java doc on

https://duboda.github.io/214