Team number: 10
Team name: Chang Gang
Andrew Chang
Ivy Chang

Tetris Program by Andrew and Ivy Chang
Last edited 5/18/2018

What:
Our goal is to recreate a fun, classical, and fairly simple game - Tetris.
Target Audience:
This game would be especially interesting to children who would appreciate it for its simplicity. Besides being straightforward and easy to control, it is also very entertaining and addictive.
Primary Features:
Tetris is about manipulating the "tetrominoes" (geometric shapes made of 4 square blocks) by rotating them and moving them sideways so that they fall on the bottom of the grid to create rows. Each time a whole row is made, the row disappears. The objective of the game is to not let the pieces stack up to the top of the grid. If that happens, the player loses and the game is over.

Instructions:
Keyboard commands:

- ❖ Right Arrow Key: Moves block right
- ❖ Left Arrow Key: Moves block left
- ❖ Down Key: Drops block
- ❖ Up Key: Rotates block
- ❖ Space bar- Drops block all the way to the top of the pile/ bottom of grid

Class List:

- Block (abstract): Contains basic methods and common properties that all blocks have (x and y coordinates, tile size) and determines which methods the subclasses of it must have.
- The subclasses below all create the shape of blocks using 4 tiles, and are inherited from the block class.
    - I-shape
    - S-shape
    - ReverseS-shape
    - L-shape
    - J-shape
    - T-shape
    - Square shape

- Action: Connects keyboard commands(inputs) to the grid which makes the blocks move
- TetrisVisual: (GUI Components) Draws a grid object (which contains all the tiles and their colors) and draws the score in the bottom
- Grid: The “brain” of the game- uses arrays to keep track of every tile and sets each tile’s color. Also contains methods to clear rows, keep score, and end the game when pieces reach the top.

Responsibility:

Assignments-

README: Both
UML Diagram: Ivy

Classes-

Blocks: Ivy
IBlock: Ivy
SBlock: Andrew
ReverseSBlock: Ivy
LBlock: Andrew
JBlock: Ivy
TBlock: Andrew
SquareBlock: Ivy
TetrisVisual: Andrew
Grid: Both
Action: Andrew

Credits:

Github- Sharing code

Presentation Credits (ex. Pictures used in presentation) are at end of presentation

Known bugs:

- In the split second right after the current block hits the pile of blocks at the bottom of the grid and right before the timer in the Action class calls on its actionPerformed method to realize that the block has hit something, the timer in the tetrisVisual class is still able to call on its own actionPerformed method which takes in commands from the Action class. This allows the user to continue to rotate the block during that split second. The reason this happens is because the timer in the Action class has a rate of 1000 milliseconds (1 second) while the timer in the tetrisVisual class runs at a rate of 10 milliseconds.
- The blocks overlap each other when the pile is only 1-2 lines away from reaching the top of the grid.