FCP: Tetris

By Andrew and Ivy Chang

Team Overview

Team Name: Chang Gang

Team Number: Team 10

Team Member Names: Andrew and Ivy Chang

Class Responsibilities:

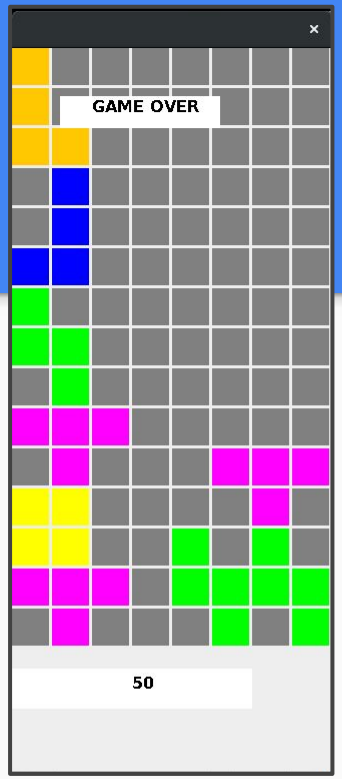
Andrew - Action, TetrisVisual, (S, L, and T Block)

Ivy - UML Diagram, Block, (I, ReverseS, J, and Square Block)

Both - Grid, ReadMe, Presentation

Explaining the Project Overview

- Our program is basically a very simple version of the popular tetris game.
- Blocks fall down from the top of a grid and you try to order them into rows.
- Each time you fill up a row it disappears and you earn points.
- Controls:
  - Up arrow key- rotate (clockwise)
  - Down arrow key- move down 1 more tile
  - Left arrow key- move left
  - Right arrow key- move right
  - Spacebar- send block straight to the bottom
- You can also earn points by pressing the spacebar.
- The game ends when the pile of blocks reaches the top.

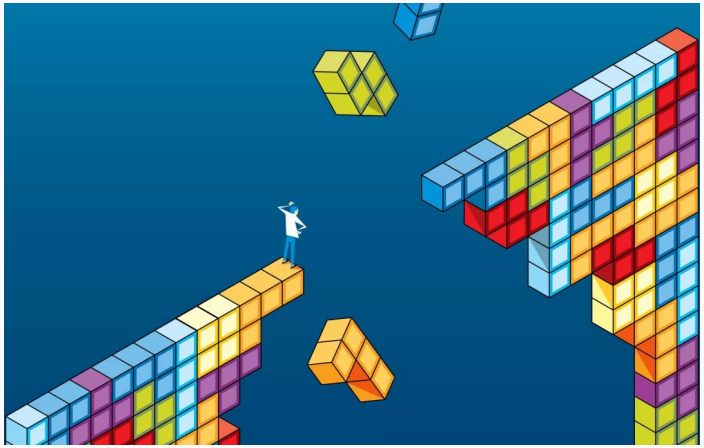


A possible outcome of Tetris

The Problems/Challenges

In the initial stages:

- We didn't know how to share code besides copy-pasting on a Google Docs, so it was hard to get the latest versions of both of our code
- We didn't use two-dimensional arrays so we had problems drawing new blocks at the top of the screen (unsure how to implement that way)



Afterwards:

- We had some issues with boundary checks

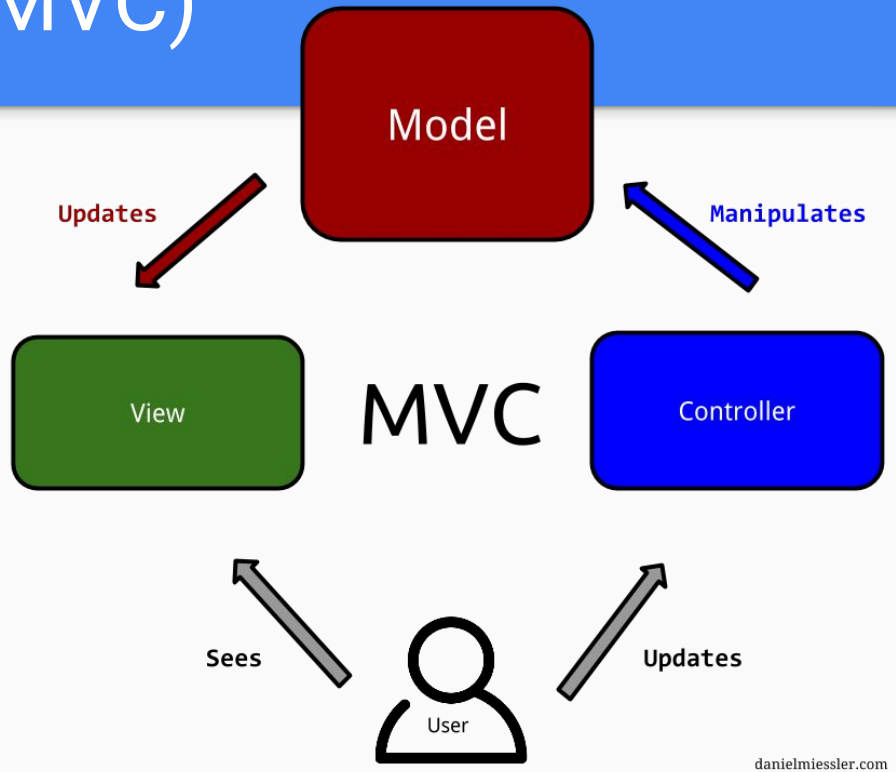
Reviewing the Design (MVC)

-Model(Data): Maintains the data based on commands

-View(GUI): Displays the model, “interface” users interact with

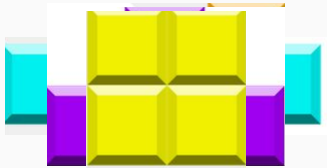
-Control(Commands): Manages the model and view

<u>Our Implementation</u>



Implementation

- 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
  - 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.



Some ideas if we had more time...

- Create an instructors/help button in the corner
- Create a pause button that would pause the game
- Make a replay button after the game has ended so there is no need to rerun the program
- Create an initial menu screen that displays the title, some options, and potentially instructions
- Make the game two-player!

!Thanks for Listening!

Questions? Ask away!!

P.S. Did you catch that we spelled TETRIS!
With the first letter of each slide after the title?

Credits

MVC Picture: https://danielmiessler.com/study/mvc/