Game plan:
We are creating a Tetris game, based on the version on the Nintendo Entertainment System (NES).

Note: Final project might be different compared to the plan listed here.

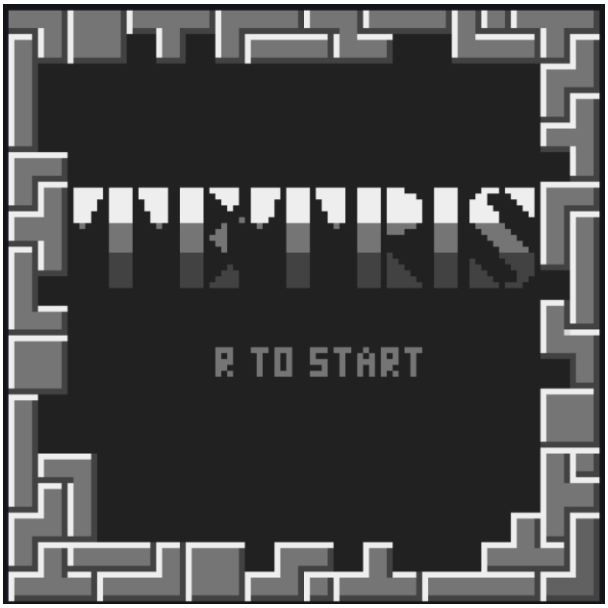
- **Game Objective:**
  - Clear horizontal lines on the playing grid by arranging falling **tetromino** (blocks / shapes) into complete rows.
  - The game continues until the player is unable to place a new shape into the grid, causing the game to end when the blocks reach the top.
- **Rules:**
  - No pause between the shapes; once the shape lands a new one starts falling immediately.
  - There will be seven different shapes (I, O T, S, Z, J & L) made up of four blocks. They can also be rotated 90 degrees clockwise or counterclockwise.
  - You can move the shapes left, right, rotate, or drop the shape quickly to the bottom.
  - Points are earned for clearing lines.
  - The player must fit the falling pieces into the grid, clearing horizontal lines as they go.
  - The game will end when the shapes fill up the grid and reaching the top as a result of failing to form complete lines and clearing them.
- **Title Screen:**
  - There will be a main “TETRIS” logo placed at the center of the screen in bold, large, text. The color may be brightly colored or monotone.
  - The background will be dark to help the title of the game stand out more.
  - There will be a start game button labeled as “START!” below the title and will be in bold text but not as large as the logo.
  - The High Scores will be below the start button showing “HIGH SCORES” and it will display the top 3 or 4 scores in the game. It will be bolded as well but the scores may be not bolded but will stand out to where you can see them. --> The High Score will be saved into a text file and be read when the game starts.
  - There will be an exit/quit button right next to start that will be labeled as “QUIT!” with the same fonts and size as the start button.
- **Game Screen:**
  - The shape will be a vertical grid (10x20).
  - The background will be a black dark background allowing you to see the blocks better.
  - The blocks will be small square units that form the play area. Each block within the grid is typically a square unit, and the shapes will fall into these spaces
  - The lines will be as the shapes fill the rows; those rows become full (leaving no empty spaces). When this happens the filled line disappears, and the rows above it fall downward to make space for the new shapes above.
  - The appearance of the falling shapes will be made up of four blocks. They are going to be brightly colored and easily distinguishable from one another. Possible color for the shapes: light blue, dark blue, orange, yellow, green, purple, and red.

    - For movement the player can move the falling shape left, right, or rotate it as it falls, trying to fit it into the gaps below.
    - There will also be a preview of the next shape that will be displayed on the left side, with a smaller rectangle (maybe 5X10?) (Planned but not implemented)
    - For the score, it will be located at the top left of the screen and it will track the player's points based on the number of lines cleared.
    - The score increased by a set number of points for each line cleared and the multiplier grows based on combo moves (like clearing multiple lines at once)
    - On the left of the screen will be a line counter, it will count the total number of lines the player has cleared.
    - When the game ends there will be a "Game over!" message that will appear at the middle of the screen. It will show the final score and either compare them to the high score or replace their score as the highest score if it was higher than the previous score.
    - Control display: it will show the arrow key mappings to help guide the player
- **Game Over Screen:**
    - The text will be in bold, large test, using an eye-catching font saying "GAME OVER!" at the middle of the screen.
    - Just below the "Game over" message, your final score will be displayed in a smaller but still prominent font.
- **High Score Screen:**
    - There will be a high score screen (if applicable) to show whether you have beaten the best score or not.
    - There will be two more buttons at the bottom saying either "RESTART" or "EXIT" to return to the main menu. The player will have the option to use the designated keys to either RESTART or EXIT.
- **Key Features:**
    - The game will consist of seven unique shapes made up of four square blocks each.
    - The playing field is a rectangular grid (typically about 10 blocks wide and 20 blocks tall).
    - The players must arrange the blocks within the grid to form complete horizontal lines.
    - When the horizontal line of blocks is filled completely, it will disappear, and any blocks above will fall down. Clearing multiple lines at once (like two or more) will earn more points
    - The blocks can be rotated 90 degrees clockwise or counterclockwise. Players can rotate and position the shapes to fit them into gaps.
    - The game will end when the stack of blocks reaches the top of the screen, meaning there's no room for a new block to fall.
    - The location and orientation of the falling blocks will be randomized.
    - The way that players will move the blocks will be to use their arrow keys.

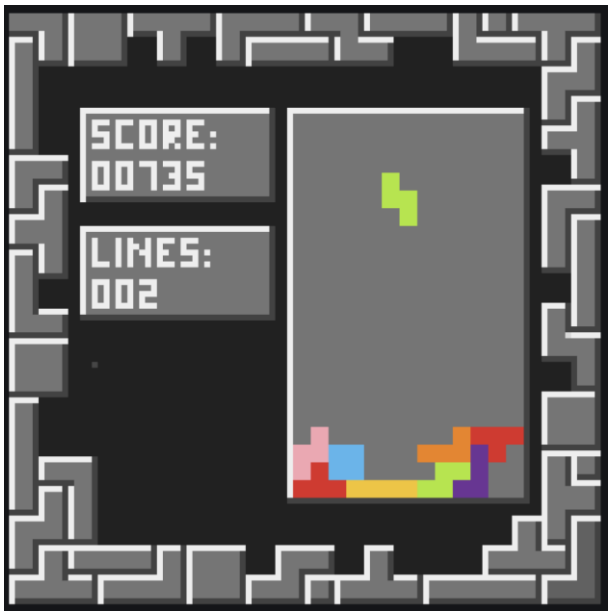
- **Our approach to making Tetris:**
    - Making a Class for the Tetromino (Blocks / Shapes)
        - Using this Class to make the multiple Tetromino in the game (creating instances)
            - Logic:
                - Shapes -> We could use Enumerate to create a list of the shapes (I, O T, S, Z, J & L)
                - Rotation of the Tetromino
                - Location tracker for this Tetromino?
                - Assign colors to the Tetromino
    - Game Interface
        - Created in JavaFX
            - Starting Screen, Play Grid, Game Over, and Highscore
            - Making an empty grid of 10x20 that serves as the game board
                - This screen will have line, score counter, and the "next shape display" on the side
            - Collision detection for the falling Tetromino and the border of the game grid
    - Game Loop
        - Some sort of frame (fps) mechanic for the Tetromino to fall and refresh game screen
        - Handling player input
        - A spawn mechanic for new Tetromino once the lowest one has collided with the grid or another Tetromino
    - Additional logic to implement:
        - Gravity
            - Increasing speed --> perhaps based on the number of lines cleared? (10, 20, 30, 40, etc.)
        - Line clearing logic for a full horizontal line
            - Perhaps a player can get additional points for clearing multiple lines at once
        - Keyboard input
            - Left and right arrows to move left and right
            - A and D to turn clockwise or counterclockwise
            - Down arrow to accelerate the drop
            - R (or Space) to start and restart
        - Low priority (might not be included in the final product):
            - Sound effects?
            - Background music?
    - Test Cases
        - Implementing appropriate Tests to check for appropriate user input

- If they input something incorrectly, then they will be asked to input the correct response instead of the program crashing

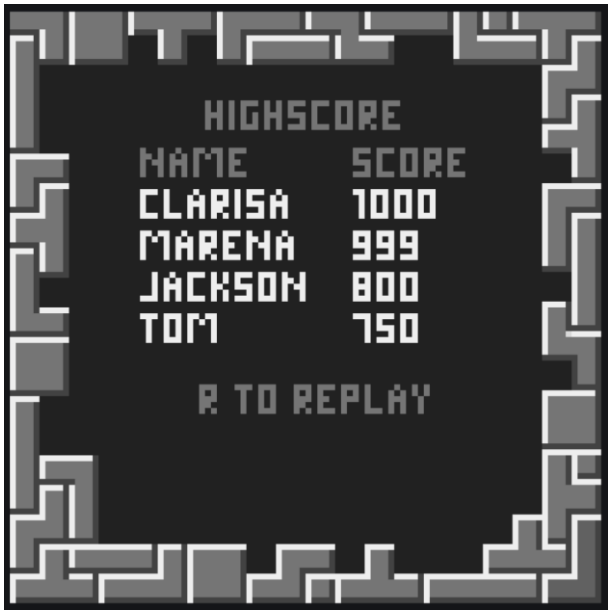
Sample Illustration for game (made by Clarisa):



Start Screen



Gameplay Screen



High Score Screen