Culminating Proposal - Tetris

The proposed culminating of interest is: Tetris. This is a common well known game that has blocks dropping from the top of the playing field to the bottom of the playing field. The playing pieces are made up of four blocks that are connected. This results in 7 playable pieces within the game. The player can move the piece horizontally or vertically. As the game progresses, rows at the bottom of the playing field will be filled up. When they are, they will be cleared and all the blocks will shift down the number of rows cleared. Consequently, as the game progresses, difficulty progresses too, which is the speed that the active block is falling down the playing field. The game is over when the active piece cannot move down from the top of the screen or cannot be created.

Goals of this culminating project:

1. Create custom data types:

'Serves as a point location.

1. PointRec:

* X as Integer
* Y as Integer

'Record of properties of game pieces active in the game.

1. GamePiece:

* Shape as Integer 'Shape of piece out of the available 7.
* Color as ColorConstants 'Color of piece.
* Position as Integer 'Current rotational position of piece.
* Center Block as PointRec 'Location of the center block of piece.
* Piece(3) as PointRec 'Relative to center block of piece.

1. GameBoard:

* Color as ColorConstants

1. HighScores:

* Name as String
* Score as Integer

1. Set aside memory for the active piece:

* Data Type: GamePiece
* Purpose: It will contain the information of the current active piece that is dropping down the playing field.

1. Set aside memory for the playing field:

* Data Type: GameBoard
* Record Array Two Dimensional (x,y)
* Purpose: It will act as a grid and contain the color of each cell in the playing field.

1. Create the active piece:
   1. Randomly select the shape of the playing piece out of the available 7.
   2. Set the fields of the active piece with the appropriate values specific to the selected shape.

'They all start in their first rotational position and the 3 blocks are relative to the center block of   
 the piece. The center block will start in the center of the playing field.

* Available Shapes:
* z: red block 1 (-1,-1) block 2 ( 0,-1) block 3 ( 1, 0)
* s: green block 1 (-1, 0) block 2 ( 0,-1) block 3 ( 1,-1)
* t: purple block 1 (-1, 0) block 2 ( 0, 1) block 3 ( 1, 0)
* o: yellow block 1 (-1,-1) block 2 ( 0,-1) block 3 (-1, 0)
* l: white block 1 (-1, 0) block 2 (-1, 1) block 3 ( 1, 0)
* j: orange block 1 (-1, 0) block 2 ( 1, 0) block 3 ( 1, 1)
* i: blue block 1 (-2, 0) block 2 (-1, 0) block 3 ( 1, 0)

1. Create the playing field:

* Set the color of each cell to the background color of the playing field (i.e. gray)

1. Draw the playing field:

* Display the playing field by drawing each cell out into a picture box.

1. Begin the game loop:

'Various tasks for proper functioning of game.

* 1. Create a new active piece if there is no active piece in the playing field, if there is an existing active piece in   
      the playing field then drop it down by one row.
  2. Check if the active piece can move down by a row, if not do:
* Check if any of the blocks in the piece filled in a row in the playing field, if there are filled rows, clear them and drop the blocks above down by the number of rows cleared.
* If rows were cleared, do:
* Add the cleared rows to the number of lines cleared. When a certain number of lines are cleared, increase the difficulty by increasing the speed at which the pieces drop at.
* For every line cleared, add 1 to the total score.
* Check if the active piece is at the top, if it is, then the game is over.
  1. Input the current active piece and remove the previous instance of it from the playing field.
  2. Delay the loop so that there is a "speed" effect. 'As the game continues, the delay will become shorter.

1. Keyboard Events:
   1. Up Arrow Key:
      * Allows the player to rotate the active piece on the playing field.
      * If the rotated active piece is within the playing field, then allow it to rotate, otherwise, do not rotate.
      * If the rotated active piece is placed only in empty cells in the playing field, then allow it to rotate, otherwise, do not rotate.
   2. Down Arrow Key:

* Allows the player to make the active piece drop at a faster speed in the playing field.
  1. Left Arrow Key:
* Allows the player to move the active piece to the left of the playing field.
* Move the piece left, if its final position is within the playing field and is in empty cells on the playing field, otherwise do not move left.
  1. Right Arrow Key:
* Allows the player to move the active piece to the right of the playing field.
* Move the piece right, if its final position is within the playing field and is in empty cells on the playing field, otherwise do not move right.
  1. Space Key:
* Allows the player to instantly drop the active piece to its final cells in the playing field.
* The final cells are only in the columns in which the active piece is currently located in, therefore the final cells would be the cells that the active piece would drop to if it continued falling straight down.
  1. P key:
* Allows the player to pause the game.

1. Game Over:

* When the game is over, if the player's score is within the top 10 range of scores from previous played games, then allow the player to add their name and score to the list.

1. High Scores:

* The high scores will only record the top 10 scores.
* They will be retrieved from a high score file when the application starts.
* They will be stored in a high score file when the application closes.

1. Menu:
   1. Game

* Start A New Game
* Pause Game
* End Game
  1. About
* Show High Scores 'This will display the top 10 high scores.
* Help 'This will display the rules and instructions on how to play the game.
* About 'This will be a copy right.