FIT2102 Assignment 1 2023
Name:Austin Sing Jet Wong
Student id: 32933975

I have been able to implement a tetris method with pure functional programming to some extent. Here are the details of how I did it.

State that changed by observable is maintained as immutable through combination of multiple function and which it allow that only new item is created which each stream will modify the state immutably which state that are automatically update is the Tick, which It could not be controlled by us while other than that it is controlled by the player.

So basically the map of the tetris is basically like a Matrix, in which only discrete position exists in tetris battle for each block in the view, which could help detect the collision by the interaction between the blocks where all of the movement is controlled by that mapping matrix. When the block is presented, it will occupy a space in matrices, which when the block moves to remain immutable, each state, the matrix is created to match the previous match as it is an important piece of the object which determines the view and the logic of the block. Even changing a value in a matrix, a new Matrix is created to ensure it is pure. The elimination of the row will change the position of block which indicates by id in the code, which it needs to move a relative distance depends on how much row is removed from its below, the score is also determined by its. by having an array of information, it could help create the matrix in a straight manner with saving static block, the game ends on when the array represents the highest point of the tetris battle is occupied by the static block.

The way Shape is determined by the relative geometric and its colour from the origin block (0,0), so that it could help changing the direction of the course according to each block respectively ,which it generate by the function with one by one for the block that which it got the unique id which it is easy to keep track for the view to be manipulated, which it helped a lot in terms of efficiency which the id is based on the total block created until game is restart. The game restart is governed by its own event which, it is reusing the state from the initial state while creating a new state with some carry-on from the previous game such as higher score.

The rotation of the block is simply twerk the vector and returns its with the new vector, which using advantage of occupied matrix, we could rotate it while all the conditional is fulfilled that it can rotate in the direction given each of the space is not occupied with each block has its respective vector that are directed from the centre of the shape for the correct mathematical calculation , which it is kinda limit in the way, as there is many situation where the block is surrounded and there is no way to rotate 90 degree, which here comes in the feature of the rotation, which it is brute force approach

So,basically, a List of respective vectors from the number of combinations -3 to 3 is created which is the maximum range that can be obtained from the number of the block in the user Block, which I merge and map it to create each possible combination. So while checking the

rotation, it will brute force each of the combinations by adding to each perspective block if one of them fails, it will not rotate , then choose another to see whether it could or not, until we get the first value that it could be rotated

The view is basically using the id to edit, create and destroy and that to remove the side effect sometimes is needed to delete and then recreate, which each state will changes something, so id is able to correctly identified what to change in the View Port and it is Basically the only way to visualise what has happened for the game