FIT2102 Programming Paradigms Assignment 1
Name: Lim Tzeyi Student ID: 30512867

To achieve Functional Reactive Programming, we have to avoid using mutable variables.

In order to achieve this, the initial part of my code. I decided to create 4 classes of constants: PaddleConstants, ScoreDisplayConstants, ShowGameOverConstants and BallConstants. The variables in each of the class are set to readonly so they are immutable and the decision to group each of the related variables together is to make the code more readable and easier to modify in the future for other programmers. When they want to change certain values objects related to those classes, instead of going line by line and changing which is tedious and the person modifying it might miss out one of the values causing the program to perform unexpectedly.

The second part of the code is that I used types for Ball, Paddle and State. This allows me to define a set of properties for the objects I want in my initial state for my program. To make it immutable the properties are set to readonly and can only accessed by using the state for the object. This is to be used as a second type checking to make sure the passed in variable type matches the specified type for the object if not the program may crash.

The third part of the code is again declaring a new constant class where the maxScore, tickRate and pongObservable$(the whole game's observable). These are very important variables as if these variables are mutable, the whole gameplay may change when the tickRate and maxScore gets modified causing the game to not run as intended.

The fourth part of the code is declaring initialState. This array uses type state so the properties of initialState must conform to the types declared in state. As mentioned, this is used as a safety net to help us check if we are passing in the correct data types to the objects.

The fifth part of the code is the part where I used scan to update the initialState by using pongMechanics function. A scan is used here so that we can reduce the values into the initialState allowing us to subscribe to the updateView function which updates our canvas with objects moving.

The sixth part of the code is the pongMechanics function. I have decided to put the score checking, collision detection and ball movement in this function as I think they all belong to the whole game's mechanics and should be placed together. To access and change the state of the it is done by letting the function accept the initialState and modify the state which in turn is passed into updateView function through scan. This causes the objects to move around in the svg.