

Assignment 1 Report

Summary the Game

The game was developed following the instructions of the assignments. Besides that,

The bot paddler is following the y position of the ball, which makes the winning of the user impossible. So there are gaps that have 4 times the radius of the ball which the bot cannot step in.

KEYBOARD COMMANDS OF game:

ArrowUp -> move paddle to top

ArrowDown -> move paddle to bottom

Space -> start or restart the game(restart after finish of a game)

M -> mute or unmute the Game

I -> adjust the size of paddle

O -> adjust the speed of paddle

P -> adjust the size of the ball

Summary the Code

I use 3 observable stream in the game:

- 1) The user control of the paddle, the reasons it separates it from other keyboard settings is previously I was attempting to develop a multiplayer mode as the extension of the game, but eventually it's unfeasible because the rxjs won't handle two keyboard events simultaneously.
- 2) The settings and the switch of the game, I map all the setting commands and game starting commands to one observable stream.
- 3) For each game, one new observable stream is used to power the game state machine.

Why Functional Programming for this game?

Because it's asynchronous, in the whole there are many instructions and changing of state, the instructions and processes are able to flow through the observable stream. So I don't need to explicitly schedule the processes and instructions, to make the code cleaner and more understandable.

Besides that, I also avoid imperative code and minimize and organize the number functions which have side effects to make sure the code is easier to maintain.

I also structure my code with Object Oriented Design, which provides encapsulation of data and organizes functions with similar business logic, which lets the other parts of code easier to find and access the data and functions, and the OOP structure itself also provides documentation to make code easier to understand.

In my code there is two important classes:

- 1) State : to store the data and control the state of the whole game, powered by an observable stream. In the whole game the observable stream moves the ball through the canvas, if the canvas touches the boundary, the stream will call the function in State to calculate the next motion, record the new result, reset the canvas etc.
- 2) Setting : a function to store the configuration and handle the changes of settings, even if the user changes the setting between the gameplay, the other functions in the system are able to access the latest data from these classes.
- 3) RNG(modified from lab code) : to generate a ballMotion object with random number
- 4) BallMotion : store and calculate the x y velocity of the ball

Extra Feature

My extra features are providing user real time controls. There are sound effects and visual effects when bouncing the ball from the paddle or top/bottom barrier, the user able to mute the sound effects at any time stage of the game. Besides that, users are also able to change the speed and size of ball, speed of paddle at any stage of the game.

