



MONASH
University

ASSIGNMENT 1

Programming Paradigms

Haider Ali

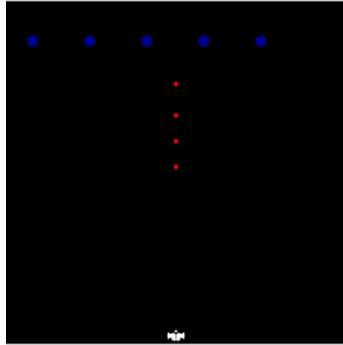
Student ID: 31371337

FIT2102

Due Date: 10th September, 2021

Introduction:

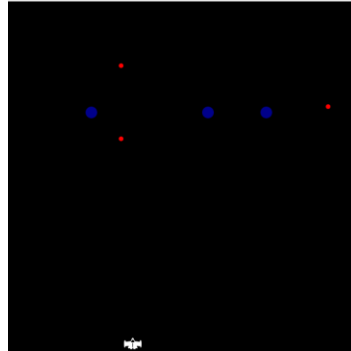
Space Invaders



Controls:

Turn Left ←
Turn Right →
Fire Space
Restart R

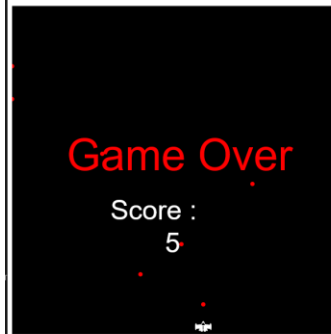
Space Invaders



Controls:

Turn Left ←
Turn Right →
Fire Space
Restart R

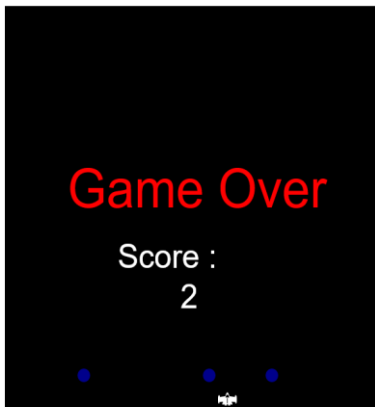
Space Invaders



Controls:

Turn Left ←
Turn Right →
Fire Space
Restart R

Space Invaders



Controls:

Turn Left ←
Turn Right →
Fire Space
Restart R

This is a basic implementation of the Space Invaders 1978 arcade game written in Typescript using Functional Reactive Programming. RxJS observable streams are used to make the program as pure and functional as it can be.

Gameplay:

The player moves a ship using the arrow keys and shoots bullets at the aliens (a round circle) using the spacebar. The player needs to destroy all the aliens before they reach the bottom of the screen or collide with the ship.

FRP used:

- Declaring constants and readonly variables so that data cannot be mutated by impure functions. Arrow functions to capture variables from enclosing scope and returning new values instead of changing them.
- Using map, filter, scan to maintain states instead of using the imperative approach. Scan has been used for handling our user input stream and filter filters the keypresses.
- Observable streams are used to handle user inputs and time elapsed. Subscribe function called on game state to observe whenever a change occurs in the observable stream (for example, user moves ship)
- A game state that is set to an initial value and changes according to the user input. As there are no impure functions (such as assigning a value in the functions) its easier to debug and there are less chances of bugs. Each function returns a new value instead of directly changing the argument passed in.
- Update view is the only impure function as it is directly making changes to the values of the attributes that are being displayed on the screen.

- Reduce State function is used to determine whenever there is a game state transition. Reduce state handles appropriate calls according to the type of game transition (Move, shoot or tick).
- handleCollisions : for handling the bullets colliding with aliens, ship colliding with alien, whether any aliens are left and for updating the state to get the new score.
- Block of code that are easy to understand and change later without massively affecting any other behavior.