

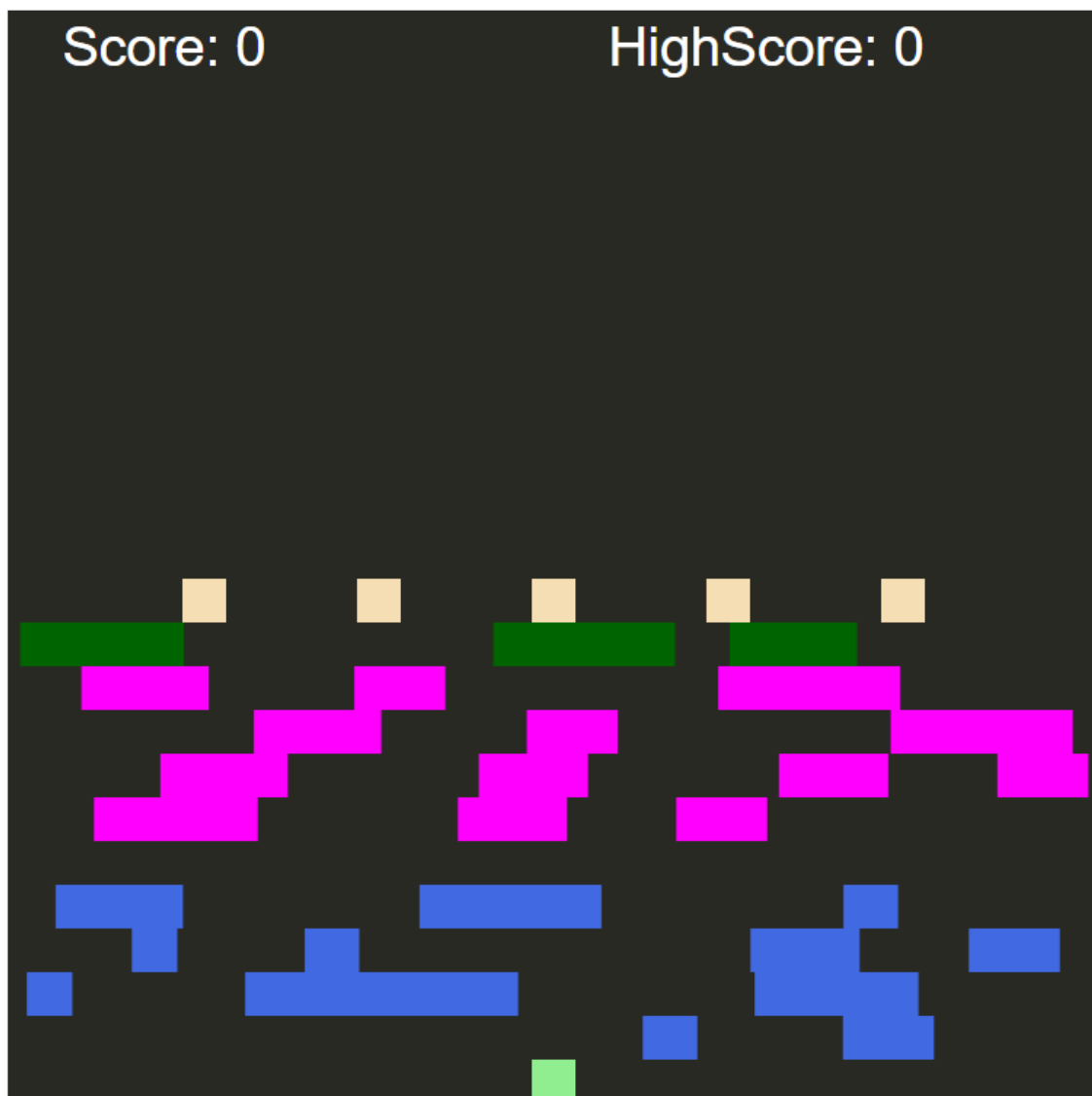
## FIT2102 Assignment 1

### Introduction

For my assignment I have created a game which need the player to control the object which is in light green colour enters the target goal which is in wheat colour and there are some obstacles like cars, logs and crocodile is in this game.

---

# Frogger



As the objects in above the light green colour object is the frog, which is controlled by the player, the blue colour object is the car which will collide with the frog object, the pink colour object is the log which lets the frog not drown in the water, the green colour object is the crocodile.

Name: Kang Hong Bo  
Student ID: 32684673

## Ground

For the first four row which has the car object the background is assumed to be the ground which the frog can stand on it and if the car collides the frog the frog will [respawn](#) at the first line. For this collision I created a body collide function which is reference from the asteroid code this function will check if the object is intersected by using filter.

## Safe Zone

The first row which is the start line and the line between the logs(pink) and car(blue) there will be a safe zone which can let the frog to stay.

## River

For the 5 row which has crocodile object and the log object the background will be the river which will let the frog drown if the frog is not on log or the back of the crocodile.

## Goal/Target

For the goal we use the same body collides which check if the frog collides with the target. Each game need players to

## Crocodile

For the crocodile is like the logs which will let the frog stand on the back of the crocodile but will be dead if the frog is in the head of the crocodile. For the crocodile in this game the direction of the object is moving is where the head of the crocodile is. The head of the crocodile is the same size with the frog.

## Game run

For each game run we will observe the input key from the user with the Keyboard Events and update the state of the game. If no key is observed the game will run with the tick to update the view of the moving object. I also create three classes excluding tick FB which is controlling the forward and backward of the frog, the LR control the left and right of the frog and the restart control the restart of the game.

Name: Kang Hong Bo  
Student ID: 32684673

## Game Over

To end the game the player must reach the 5-target goal to finish the game else the game will game over.

## Scoring

The score for this game is using the state that I added TOS which will count the time for the counting of the score, the player will get higher score if he gets to the target faster. For each target approached the state of the isScore will become true to update the score in the game else will not update anything for the score.

## Move Body

Having a velocity in the body of the object to control the movement of the object with positive velocity will let the object move to the left else will go to the right. For controlling the frog object is mentioned in [game run](#).

## Respawn

The respawn of the frog, which means that the frog is collided with by cars or drowned or eaten by the crocodile will be set to the starting point of the game.

## FRP design

For all the collision I am using map, filter and concat to control the state of the object to be updated. Then creating the body of the object is using a carried function which will create rectangles and using map to insert the arguments for the creates object. In the handling collision I made lots of functional relative which control the state to pass in the reduce state.

## Maintaining purity

For most of the handle collision function is a purity function since only specific output will be outputted and no side effect will produce from the code. This also make our game easier to add on function and features in future since is possible to use back the pure function.

Name: Kang Hong Bo  
Student ID: 32684673

## Restart game

For restarting the game, I included the observable which take the input of key R to set all the state to the initial state but remain the high score for the game output.

## High score

After each game the high score will save the highest score in the history and will not be gone while the game is continuous running. We can simply do this by having a guardian which see if the current score is higher or not compare to the high score which is recorded earlier.