

## Frogger Game Design Decisions

**Purity:** The game manages the state and maintains purity by constructing an initial state, and scanning the initial state and reducing state parameters, the state transducer, which subscribes everything to the update function for updating the observable view shown in SVG screen. In all the functions, they return a copy of current state with new changes of the attributes and continuously pass it to the next function till a round of scan ends. Also, the type and interface attributes are set as read-only to avoid the change for breaking the purity.

The FRP style programming is following the cycle of:

View – sees -> User – uses -> Controller – manipulates -> Model – Updates -> View

**View:** From the designing aspect, the first consideration is that how are the graphics looking like for user. The method for the visualization of Observables is by giving the shape, color, size and name for each body in the game to identify them. I used `document.getElementById` and `document.createElementNS` to get the body, and used `setAttribute` to set the view for each body of a type. In particular, the only frog is set as a circle, and others are set as rectangles with different colors, position, size. The moveable bodies can move with a given velocity in a single row. Then collisions will show a reaction on the SVG for directly telling to the player, which is either game over or the frog actions. The Observable aims to text in front of screen, it needs to perform 2 kinds of text showing, first is the information that is fixed on the top of the screen, the second is like Game Over which shows with conditions.

**User and Controller:** When the player can see the Observable in the SVG screen, the frog, a controllable observable, should be interacting with player. I set up five key events that listen for pressing the corresponding key, which are arrow keys for moving the frog in four directions and key R for restarting the game.

**Model:** In this game, each item of a type should have a property, like the controllable(frog), moveable(car, plant, crocodiles), desatation(home), fatal(car, crocodiles ,fly ). The game is running and updating by a state, which describes the current situation, such as the frog's position and score. A RNG random number generator is included in state for choosing a random unreached home to appear a fly and become fatal and the fly appears and disappears depends on the elapsed time. Furthermore, I decided the time unit or interval in this game, since the movement of each item should be independent, an appropriate distance moving per interval and the length of interval are vital for this time-based game

### Highlights:

**Restart:** a restart functionality that can reset the game by pressing key R in any time. When it listens the event of key R down, an event of restart is recognized in reduce state which returns the copy of initialState that gives all the initial settings with type state. It can be easily re-used in each stage, but keeps some attributes by changing the copy of initialState.

Highest score: the game is able to keep track of the high score across previous rounds without refreshing the page. There are attributes named score and highest score in the type of state and initial value of 0 in the initial state. The score increases 5 for each forward step of the frog, and increase 50 when the frog reach a unreached home, and this score will inherit to next level, and clean to 0 when frog die. While the highest score is comparing itself and current score and changed to the greater one, which keeps at gameover and restart by {...initalstate, highestscore: state. highestscore } .

Row design: In order to make Smooth and usable game play, the fixed distance of movement is set, I decided to give 25px as a row, thus, all the body should only has the height with 25px and the frog only move vertically with 25px as well. This keeps the uniformity of the game.

3 distinct objects: the game have many objects that behave differently, which are car/ plant, crocodile, fly.

Crocodile: a game body appear in river only, and combined by two objects, body and head. Simply, the body of crocodile behave similarly with plant that can contain frog, abd head body of crocodile behave similarly with car that can collide with frog.

Home: the fly may appear and take up the home location, which become fatal when the frog approaches. The appearing of fly is based on time, it appears at start of game and every 20, and disappears at every 10 seconds. The color of home normally is blue, and a frog reached home is green, and when a fly appear, it is filled in red.