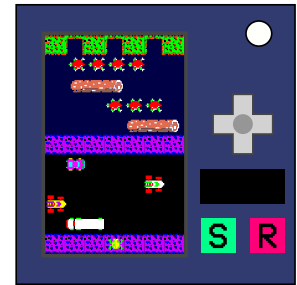# On the Subject of Ribbit Programming

Is this frogramming?

This module has a D-Pad (consisting of four directional arrows and a middle button), a small input display, a start and reset button, and a large game of "Frogger".

From bottom to top, you are shown:

- A thin patch of grass (the starting point) with a frog. You will be controlling this frog.
- A street with multiple cars. (Initially frozen in place.)
- A second thin patch of grass (the middle point).
- A river with logs and turtles. (Initially frozen in place.)
- A final thin patch of grass with gaps (the goal points).

The goal is to pre-program the movements of the frog such that it moves from the bottom patch of grass to one of the top goal points, whilst weaving through the traffic in the streets and jumping on the logs and turtles in the river.

The frog can be programmed to either move up, right, down, left, (with the directional buttons on the D-Pad) or wait (with the central button on the D-Pad). Moving the frog in any direction causes it to jump one unit in the given direction.

The length of one unit is illustrated along the top and right of the module via short lines. The patches of grass are each one unit wide, the street and river are each four units wide, and the goal points are one unit wide.

The cars, turtles, and frog are one unit long and one unit wide. The logs and groups of turtles are one unit wide and can be several units long. One of the street lanes has a lorry instead of a car, which is two units long.

If the frog jumps onto a log or turtle, the frog will move at the same rate as the log or turtle it is standing on.

While the program executes, the moving items (car, lorry, log or turtle) move sideways at constant rates (wrapping across the edges of the game), while the frog's own movements occur instantaneously at every integer time step. The first command occurs at time 1.

How can the program stop?

If any of the following occurrences happen, the module will strike and reset:

- The frog dies upon getting hit by a car or lorry.
    - Specifically, any amount of sub-units is shared between the frog and a car or lorry.
- The frog dies upon falling into the river.
    - Specifically, no amount of sub-units is shared between the frog and any log or turtle.
    - The frog can die simply by jumping directly into the river.
    - The frog can also die if it gets shoved off the log or turtle it is standing on. If the frog is standing on a log or turtle, and it reaches the edge of the screen, the log or turtle will wrap around, but the frog will not. This can cause the frog to get shoved off and fall into the river.
- The frog dies of boredom.
    - Specifically, if five steps in the program have occurred without the frog jumping to a new position. (The frog may be moving on a turtle or log at this time, but this isn't considered movement in this context.)
- The frog steps on the top patch of grass instead of a goal.
- The frog jumps outside of the game window (for example, jumping left from the leftmost column).
- There are no more instructions in the program. This will kill the frog immediately.

A skull and crossbones will appear at the location of the frog's death. Press the 'R' button to reset to the initial state of the module.

Direction and position of moving items

The logs in the river always move right.

The turtles in the river always move left.

The cars in the streets may move either direction. The cars move in the direction that they face.

All moving items start in whole-unit positions.

Determining the speeds of the cars

Street lanes are numbered 1-4, from bottom to top.

Take the battery count (adding or subtracting 4 until it falls into the range of 1-4). This lane's car moves <u>1 tenth</u> of a unit per step in the program.

For the first three characters in the serial number, convert any letter to its alphabetical position, and set any 0's equal to 10. The remaining three lanes' cars (from bottom to top) move a <u>tenth</u> of that many units per step in the program.

Determining the speeds of the logs and turtles

River lanes are numbered 1-4, from bottom to top.

Take the indicator count (adding or subtracting 4 until it falls into the range of 1-4). This lane's moving item moves <u>1 tenth</u> of a unit per step in the program.

For the last three characters in the serial number, convert any letter to its alphabetical position, and set any 0's equal to 10. The items in the remaining three river lanes (from bottom to top) move a <u>tenth</u> of that many units per step in the program.

How does the module solve?

Upon moving the frog to any one of the three goals at the top of the screen, the program stops, and the module disarms. The goals are located at X positions 1, 3, and 5, but the frog can safely jump to them with a maximum leeway of <u>5 tenths of a unit</u> to the left or right of the goal.