Final Project: Frogger

EE379 Spring 2019

Objective

In this project, you will implement the classic Frogger arcade game using C on the LandTiger board. Your game will be displayed on the GLCD display and you will control the game using the joystick on the board.

Deliverables

Assigned: 4/11/19 (remember that there *will* be a Lab 5; 1-week lab, week of 4/15).

Demo:

- Fully working Frogger arcade game on the LandTiger board, meeting all requirements outlined below.

Demo due: By 5PM, Friday, 5/10/19

Design Document due: 11:59PM on Wednesday, 5/15/19. Submit all code that you wrote, plus a detailed design document including a schematic/block diagram and API documentation of your code. The final project document should be like a more detailed version of your standard lab reports.

Lab Contents

Part A: Frogger Game Overview

Frogger is a classic arcade game from 1981, developed by Konami in Japan and distributed worldwide by Sega. Frogger is widely regarded as one of the classic arcade games and holds a definitive place as one of the great video games.

Before starting the project, you should play some version of the Frogger game. There are many online versions of the game. One classic version can be played at

http://www.free80sarcade.com/frogger.php

Frogger is played by controlling frogs from the bottom of the screen to their homes at the top of the screen one at a time. First, the frogs cross a busy road with cars and trucks moving in alternating directions. Next, the frogs jump across a river with logs and turtles floating in opposite directions. Five homes are located at the top of the screen; one frog must land into each home to win the game.

If the frog gets hit by a car/truck, misses the log/turtle and falls into the water, or scrolls off the screen while riding on the rivier, the player loses a life. The player starts with four lives. The player loses by losing all lives.

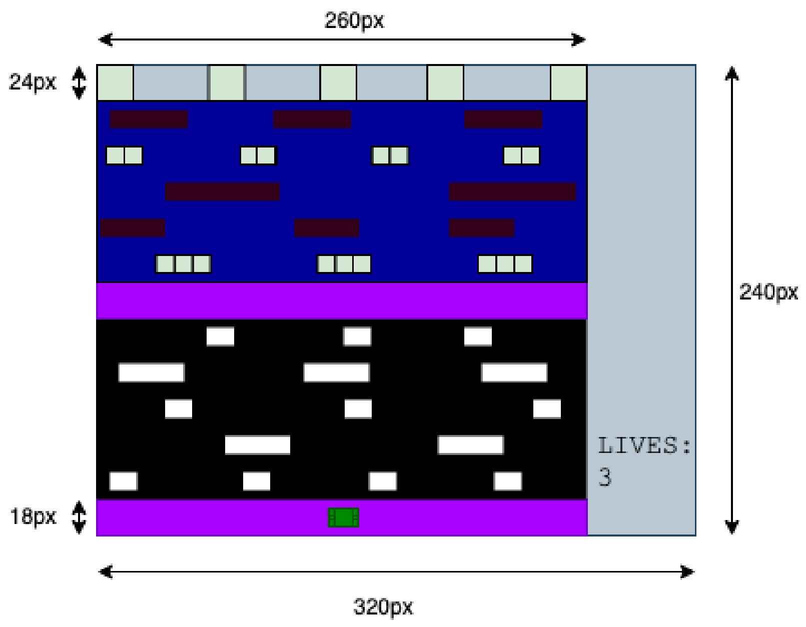
Part B: Frogger Game Requirements

Your Frogger game must meet all of the following requirements:

- Frogs should be displayed with a sprite that is more complicated than a simple square/rectangle (see below for an idea)
- The frog should be controllable with the joystick on the board and able to move left, right, up, and down.
- The frog should start on a safe area on the bottom of the screen.
- Five rows of cars/trucks should appear on a road that occupies the bottom part of the display. There should be two rows of trucks and three rows of cars, and the trucks should be bigger than the cars (which rows are cars and which are trucks is up to you) They should move in alternating directions.
- Cars/trucks may be drawn as rectangles. Their pattern may be simple and repeating. They all may move at the same speed.
- A safe zone should be displayed half way up the screen. Frogs can land anywhere on the safe zone.
- Five rows of turtles/logs should be displayed on a river that occupies the top part of the display. There should be two rows of logs and three rows of turtles and they should move in alternating directions. The pattern may be simple and repeating, and they all may move at the same speed.
- Five homes should be shown at the top of the screen. The frogs should only be able to land in these homes. Missing the home does not have to result in the frog dying. Once a frog lands in the home, a new frog should appear at the bottom of the screen.
- Frogs getting hit by cars/trucks, jumping into the water, or riding a log/turtle off screen should result in a loss of life and a new frog starting at the bottom of the screen.
- The number of lives must be displayed on the screen, starting with four.
- The text “You Win” should appear if five frogs are navigated to the five homes. “You Lose” should appear if the player loses all lives.

Part C: Example Diagram

The diagram shown below is NOT a requirement, but just an example of how you might construct your game board. I encourage you to be creative with colors, shapes, sizes, etc. as long as the requirements are met.



Part D: Extra Credit Options

Do NOT attempt any of these until ALL of the above requirements are satisfied. You can't get extra credit unless all of the basic requirements are met. The maximum score on the final project is 150/100.

- Actual sprites: Rather than using simple rectangles for the frogs, cars, trucks, logs, and turtles, create actual sprites that look similar to the real objects in the Frogger game. [up to 25 points].
- Keeping score: In addition to remaining lives, also display the score. The score should go up whenever frogs make successful moves towards the top of the screen. See the actual gameplay for scoring amounts. [15 pts]
- Multiple levels: When one level is cleared, start a new level in which the game speeds up. The level number should be displayed on screen. [15 points]
- Control using the keyboard: Use a PC keyboard to send UART commands to the board to move the frog instead of the joystick. Any keys are acceptable (w/a/s/d or arrow keys, for example) [30 points]
- Pause/resume option: Use a pushbutton to pause/resume the game. Display the text "PAUSED" on the center of the screen when paused. [10 points]
- Animations: any kind of animation when the frog dies, jumping animations, etc. [Up to 35 points]
- Varying speed of cars/trucks/turtles/logs: In the actual Frogger game the speeds of the objects in each row differ. Create at least three different moving speeds [15 points]
- General style/other cool stuff: If you think of any other features that aren't mentioned here, feel free to mention them to Dr. Fritz. This is your chance to be creative! Also, you might get extra points for great style or other cool features we didn't think about! [various points]

Part E: Submission Instructions

Submit a Zip file containing all .c files you wrote plus a PDF design document. Your document should include a schematic/system block diagram as well as API documentation for all of your code.