# The Classic Arcade Game - ‘Frogger’

## Product description

We built the classic arcade game - ‘Frogger’. The objective of the game was to help our character (the frog) cross the bottom of the screen to the top by dodging the obstacles (cars and trucks represented by colored rectangular pixels) which ran across the screen horizontally at different rates. The game started with the vehicles moving at the minimum pace. However, as levels progressed, the speed of the vehicles increased which essentially made the game difficult for the user.

## Components Used:

The following components were used in order to complete building the product:

1) We used an Adafruit 32x32 RGB LED matrix panel (model number : CRESCENT ENTERPRISES - 267) to display the game contents.

2) We had a CFAL1602C (OLED) display for displaying and keeping track of the player's progress and score which was configured using Serial Protocol Interface.

3) Push Buttons were used to avail movement to the frog in different directions. The controls were programmed using General Purpose I/O peripheral to include upwards, left-side, and right-side movement. Along with the main controls, there was a reset button to go back to a default screen.

4) We used UE-Mega Boom Speaker (model number: 984-001504) configured using Digital-Analog Conversion to play Tetris theme soundtrack.

## Challenges Encountered

As a part of building this project we faced several challenges that were addressed by following a stepwise approach and understanding the working mechanisms of the various components we were working on.

The challenges encountered and approach adopted to solve the issues are as follows :

## Challenge encountered to get the LED Matrix display functional

The RGB LED Matrix was challenging to work with, and it was one of the major roadblocks for the project. We first understood the sequence that we needed to take for enabling different ports for the LEDs to be enabled, and for that we learnt more about shift registers, and how the output-enable (OE) and latch worked together. However, after doing it row by row, we couldn’t enable pixel by pixel, and the two halves of the matrix were mirroring each other. What we realized soon after is that we weren't clocking in zeros for pixels that were to be disabled, and

we had not positioned the rows to be enabled in the right place (after enabling the latch and the output enable set to high).

After figuring that out we were able to enable the LED matrix pixel by pixel, by continuously running the display in the main.

**Challenges encountered to configure the game layout**

One challenge we faced was a bug where the array drawing the cars was going out of bounds and having unseen behavior, so our collisions were happening in unknown areas. We then realized that the array was going out of bounds and just had to make sure that it stayed in bounds and that fixed the issue.

Another challenge we faced was that initially to draw the cars we created a car struct and had an array holding all the vehicles. We then realized that creating this structure that it would take up too much memory. So, we then had to have set patterns of when the cars are being drawn and draw them at those times.

**Challenges encountered to build the Printed Circuit Board (PCB)**

Working on the PCB was a challenge for one main reason - time constraints. We had a good idea of how to construct the schematic for the PCB, however, since this occurred in parallel with the running changes that were happening with the rest of our project it was difficult to keep up.

We were able to make a schematic, layout, and send it off to a vendor in the span of a week, but it unfortunately didn't come on time. In addition to that, there were changes with GPIO pins that we used since one of our peripherals had to be changed in the last week. After this change was made, the PCB was out of the picture.

**Challenges encountered to configure sound output using Digital-Analog Conversion**

There were also some challenges with using the Digital-Analog Conversion to play Tetris-theme song. After finding the frequencies for each note, and putting it through the wavetable, we had to put it through a timer to keep replaying the song. When we put it through a timer, it was initially playing each note in beeps instead of continuously -- we realized that we were reinitializing the wavetable every time a note was played, therefore it was not continuous. We resolved this by only initializing the wavetable once in the main.

In addition, on commencing the game flickering sounds could be heard which was an extremely unpleasant experience. To solve the issue, we learnt about setting the priority of timer that was being used by the DAC. Setting the priority resolved the issue as it introduced smoothness to the audio output. Moreover, we observed that it potentially reduced the flicker in the display too.

**Challenges encountered to setup the CFAL1602C (OLED) display**

The prime challenge encountered while working on OLED display was that it displayed random letters or sometimes did not display the entire string. We realized that it could be possible that the input provided might be exceeding the length of content it could display. Fixing that resolved the issue. The level on which the user was currently on and the final score were successfully displayed.

**Recommendation to anyone working on something similar / using the similar components**

The lack of documentation for RGB LED Matrix was one of the biggest challenges, and the lack of clear cut instructions on how to implement it. One of the biggest sources of help was only repositories using Arduino, STM 32, and also understanding the different ports using the one detailed article by sparkfun.

It is highly encouraged to start building on the project idea at the earliest possible because as the semester progresses, the time constraint can make things really stressful and led to bad decisions which in turn results in missing out on making the most out of the project. Getting a display set up regardless of whether it is an RGB LED matrix or an LCD display, there is a lot of time investment needed to initially get the screen display to work.

For teams working on a PCB, it is recommended to get a good grasp of the initial circuit layout and finalize the schematic as soon as possible. Moreover, making two revisions of the board is also recommended.