TEAM 3

FROGGER

GAME MANUAL

Table of content

1. **Introduction** page 3
2. **List & Description of components** page 4
3. **Hardware Requirements** page 5
    a. **Software Requirements**
4. **Setting Up the FPGA Board** page 6-7
5. **How to play** page 8
6. **The Game** page 9
7. **Troubleshooting** page 10-12
8. **Technical Support** page 13

Introduction

Welcome to Frogger!

In this game, you control a brave little frog trying to cross a busy road filled with speeding cars. Your goal is to guide the frog safely to the other side, using both the digital game on the screen and the physical GoBoard to move. You have three lives (hearts), so each move counts!

Safe Zones and Danger Ahead

The road is dangerous, but there are a few safe zones along the way. The first line is safe, so you won't get hit by cars right away. As you hop forward, you'll find another safe zone in the middle of the road to rest before making your final push. The last safe zone is near the end, right before reaching the safety of the resting point on the other side.

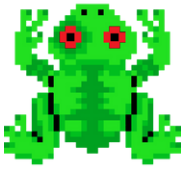
Challenge Increases with Every Level

As you cross successfully, the game gets tougher. The cars move faster with every level, and your frog will have to be quicker and smarter to avoid them. Your challenge is to survive until Level 15, where you'll win the game if you can keep your frog alive through the increasing speed and chaos!

Good Luck!

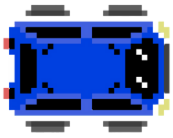
Hop carefully, avoid the cars, and use the safe zones wisely. Will you guide your frog to victory, or will the traffic prove too much to handle? Good luck!

List & Description of components



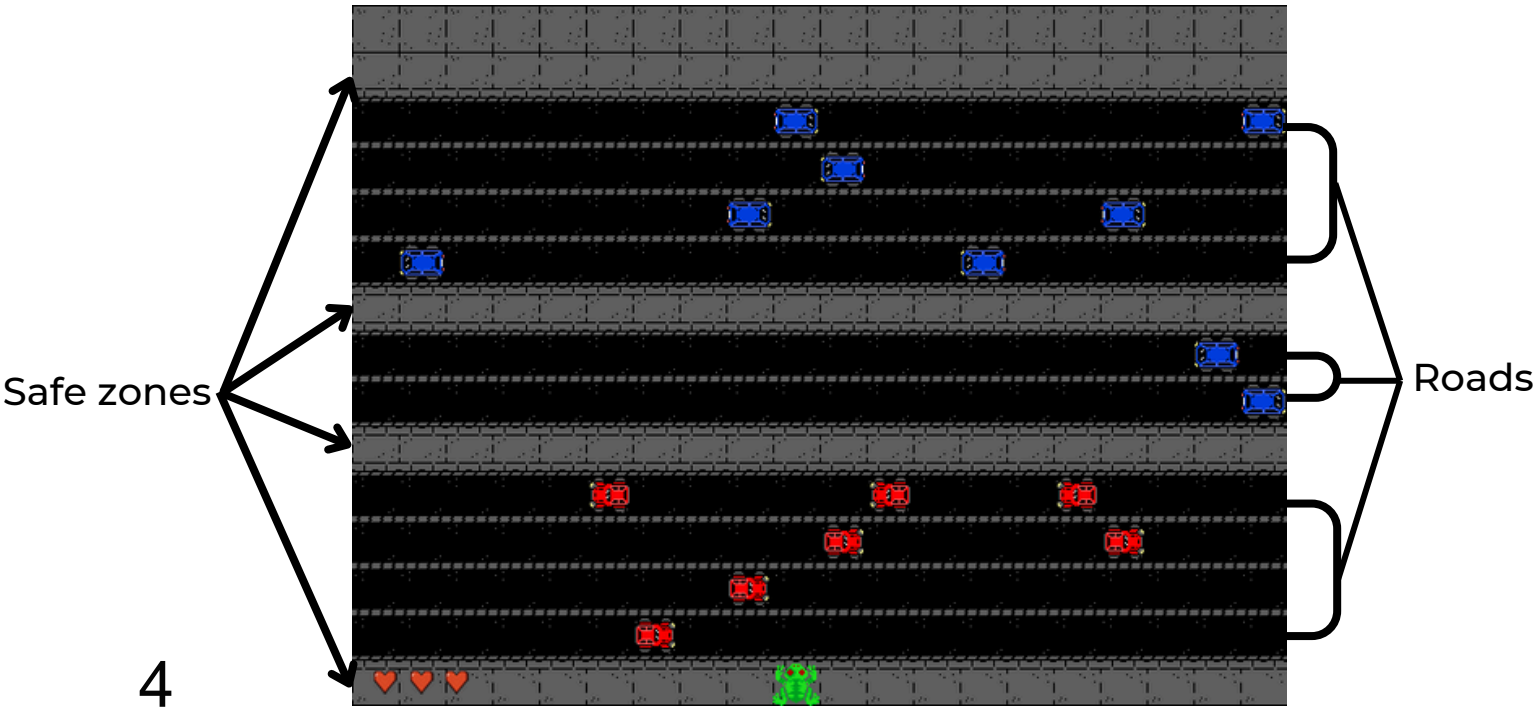
Description: The player's character. A frog that must hop across a busy road without getting hit by cars. Controlled by the player on the screen and the GoBoard. The goal is to cross the road safely while avoiding dangers.

Description: One of the obstacles. Red cars speed down the road and will cause the player to lose a life if the frog is hit. They appear in increasing numbers and speed as the levels progress.



Description: Another type of obstacle. These blue cars move at a different speed compared to red cars, adding variety and difficulty to the game. Just like the red car, hitting a blue car results in losing a life.

Description: Represents the frog's lives. The player starts with three lives (hearts), and each time the frog gets hit by a car, one heart is lost. When all hearts are lost, the game resets.



Hardware Requirements

NANDLAND Go Board FPGA

Lattice ICE40 HX1K FPGA chip
25 MHz on-board clock
USB connection for power and communication
VGA port for video output

VGA-Compatible Monitor

Minimum resolution: 640x480
Optionally, a VGA to HDMI converter for HDMI monitors

USB Cable

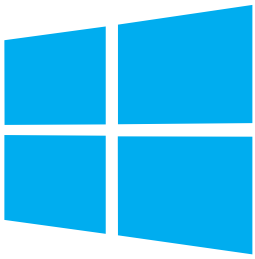
For connecting the Go Board to your computer and powering the FPGA

Power Supply

Go Board powered via USB or external power supply

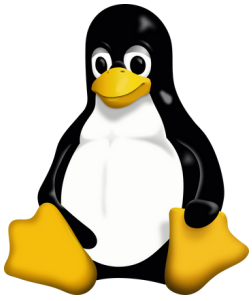
Software Requirements

Operating System



Windows

macOS



Linux

Setting Up the FPGA Board

Connect the FPGA

Use a USB cable to connect the Go Board to your computer.

Install apio

Follow the Apio setup guide on Nandland website to **install** Apio and its **required** dependencies on your system.

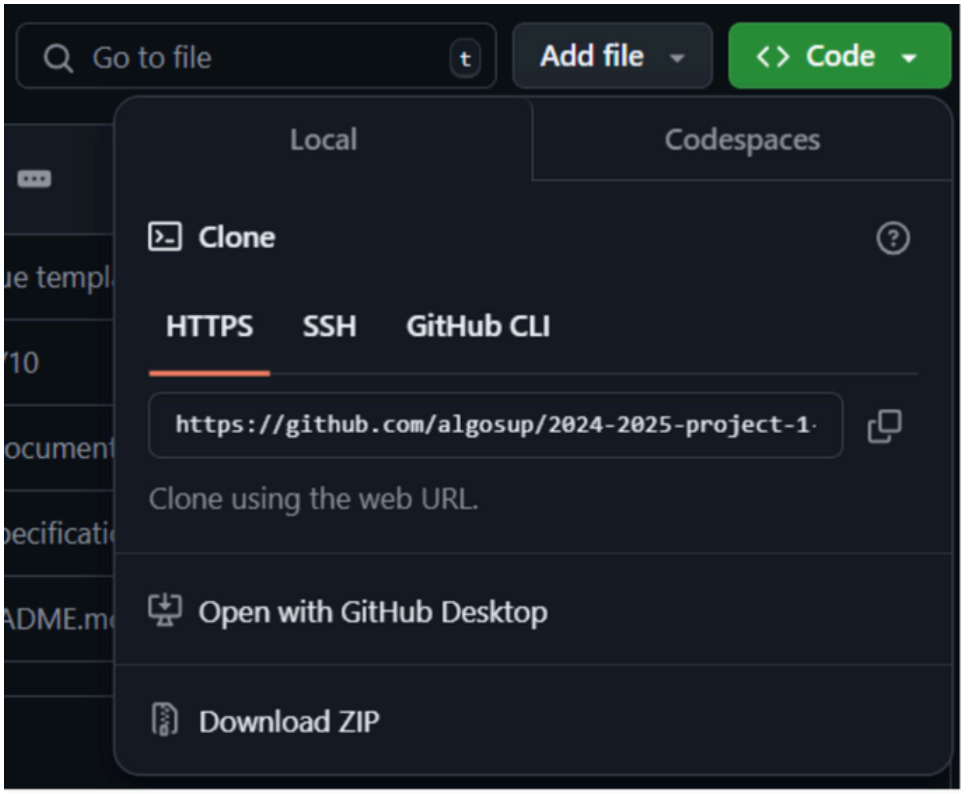
Connect VGA and Power

Connect the VGA cable to a monitor and make sure the board is powered either through USB or an external power source.

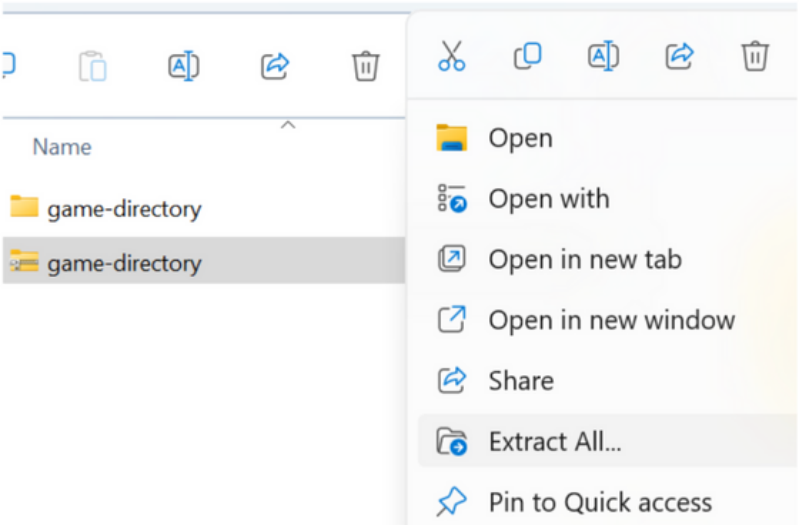
Download the game:

1. **Go** to our GitHub repository: https://github.com/algosup/2024-2025-project-1-fpga-team-3.

2. Click the **Code** button and select **Download ZIP**.



3. **Right-click** on the ZIP file and select "**Extract**" or "**Extract Here**" from the context menu



4. **Upload** the game to GoBoard:
    - **Open** your terminal.
    - Use the **cd** command to navigate to the directory where you extracted the game files.

      ```
      C:\Windows\System32>cd path/to/your/extracted/game-directoryc
      ```

    - **Run** the command Apio upload to upload the game to the GoBoard:

      ```
      C:\path\to\your\extracted\game-directory>apio upload
      ```

How to play

Get to know our GoBoard:

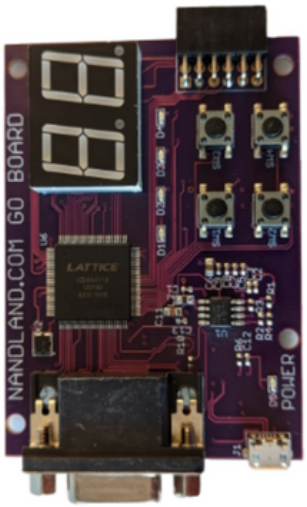


figure 1

Our GoBoard Has:

1. **Two 7-segment displays**.
2. **Four LEDs**.
3. **Two Ports**:
   - VGA.
   - Micro USB.
4. **Four switches**:
   - SW1
   - SW2
   - SW3
   - SW4

How to hold the GoBoard for Easier Gameplay:

To make it easier to play while holding the GoBoard, we recommend rotating the board so that SW3 (Up) is **positioned** at the **top**, or facing north. This orientation offers a more comfortable grip and better control. Refer to figure 1 for visual guidance.

How to use the GoBoard:

- **Use** the switches to control the frog's movements:
  - **SW3**: Move Up
  - **SW2**: Move Down
  - **SW1**: Move Left
  - **SW4**: Move Right
- Each time you **press** a switch, the frog **moves** one step in the chosen direction.
- **Reinitialize** the frog or **restart** the game after reaching level 15, by **pressing all** four switches at once.

The Game

Levels:
The game consists of 15 levels, each increasing in difficulty. At the start, it's easy for the frog to navigate through the cars, but as you progress, the car speeds increase, making it more challenging. The frog will need to move quickly and carefully to avoid getting hit.

Each level begins at the bottom center of the screen and ends at the top of the highest road. Once you successfully cross that final road, you advance to the next level. You can reach the top from any position.
The current level is displayed on the 7-segment display on your GoBoard.

Lives system:
Each time the frog gets hit by a car, it returns to the starting position at the center of the bottom of the screen. Since our frog is still learning to cross roads, we've given it three lives to help along the way.

The frog will only lose and return to Level 1 if it gets hit three times. However, if the frog successfully crosses all roads and completes all 15 levels, congratulations are in order! The frog has made it safely to the other side.

Troubleshooting

Troubleshooting for macOS:

If you encounter the message **zsh: command not found: apio**, follow these steps:

1. **Create** a virtual environment:
    - python3 -m venv path/to/venv
2. **Activate** the virtual environment:
    - source path/to/venv/bin/activate
3. **Install** the necessary dependencies:
    - python3 -m pip install xyz
4. **Install** apio:
    - pip3 install apio
5. **After** the installation, **upload** using:
    - apio upload

This should resolve the issue and allow you to use the apio command properly.

Troubleshooting

Troubleshooting for Windows:

If you encounter an error such as **'apio' is not recognized as an internal or external command**, operable program or batch file:

1. **Ensure** Python is **installed** and **added** to your PATH:
    - **Open** Command Prompt and type **python --version** to check if Python is installed.
    - If Python is not recognized, install it from the official Python website, and ensure you check the box "**Add Python to PATH**" during installation.
2. **Set up** a virtual environment:
    - python -m venv path\to\venv
3. **Activate** the virtual environment:
    - path\to\venv\Scripts\activate
4. **Install** required dependencies:
    - python -m pip install xyz
5. **Install** apio:
    - pip install apio
6. **Upload** using apio:
    - apio upload

This should resolve the issue and allow you to use the apio command properly.

Troubleshooting

Troubleshooting for Linux:

If you encounter an error such as bash: **apio: command not found:**

1. **Ensure** Python is installed:
    - **Open** a terminal and type **python3 --version** or **python --version** to check if Python is installed. If not, install it with: **sudo apt-get install python3**
2. **Set up** a virtual environment:
    - python3 -m venv path/to/venv
3. **Activate** the virtual environment:
    - source path/to/venv/bin/activate
4. **Install** required dependencies:
    - python3 -m pip install xyz
5. **Install** apio:
    - pip3 install apio
6. **Upload** using apio:
    - apio upload

This should resolve the issue and allow you to use the apio command properly.

Technical Support

For any hardware-related issues, please contact us at **team3@aftersale-algosup.com**.
Our team will assist you with troubleshooting and resolving your problem.
For any other queries or support requests, feel free to reach out via the same email, and we'll ensure your issue is addressed promptly.

Thank you for choosing our game!

Credits

Abderrazaq Makran – Technical Writer

Julian Reine – Program Manager

Lucas Aubard – Project Manager

Vivien Bistrel Tsangue – Technical Leader

Mariem Zaiane – Software Engineer

Manech Laguens – Software Engineer

Raphael Chiocchi – Quality Assurance