User Manual – FROGGER

Project 1 – Team 4 ALGOSUP

1. Introduction

- **Game Overview:** Brief description of the game, its goal, and any unique features.
- **Objective:** Main objective
- **Target Audience**

2. System Requirements

- **Platform**
- **Minimum/Recommended Specifications**

3. Installation Instructions

- **How to Install:** Step-by-step guide on how to download, install, and launch the game. (Windows/MocOS devices)
- **Uninstall Instructions**

4. Game Controls

- **Button/ Switch Mappings**

5. Gameplay Instructions

- **Game Mechanics**
- **Levels**
- **Scoring System**
- **Lives and Game Over**

6. Tips and Strategies

- **Basic Tips**

7. Troubleshooting

Introduction

Game Overview

Frogger™ is a classic arcade game where players guide a frog from the bottom to the top of the screen, navigating a road filled with fast-moving vehicles. The frog must avoid cars. Players earn points for each successful crossing and must complete each level. The game becomes more challenging with faster obstacles. When the player get's hit by a car the game scores restets to 0. The game ends when the player is not able to cross the road.

Objective

The main objective in **Frogger™** is to guide the frog safely across the road and river, ultimately reaching one of five homes at the top of the screen. Players must avoid vehicles. Successfully crossing the road where difficulty increases.

Target Audience

Frogger™ appeals to players of all ages, offering simple yet challenging gameplay. Its intuitive mechanics attract casual players, while its increasing difficulty and high-score potential engage more dedicated gamers. Retro gaming enthusiasts and those with nostalgic ties to 1980s arcade games will also enjoy it. The game's family-friendly nature makes it suitable for younger audiences as well.

System Requirements

Platform

The game runs on the **Nandland Go Board**, a hardware development platform.
Learn more about the Nandland Go Board.

Nanadland Go Board website (**Source:** https://nandland.com/the-go-board/)

Minimum/Recommended Specifications

- **Hardware**: Nandland Go Board (pre-loaded with the game)
- **Power Source**: Micro USB power supply
- **External Screen**: VGA display connected via the Go Board's video output
- **Computer**: Required to upload the game if not pre-loaded; source code available on GitHub: GitHub Repository

For detailed instructions on uploading the game to the board, refer to page (...).

Software

No additional software or operating system is needed; the game runs directly on the Nandland Go Board.

Instalation instructions

How to install for a Windows device:

This section explains how to install all necessary components for **FROGGER™ Project** on a Windows device, including Python, APIO, and Git. It also details how to clone the project repository from GitHub and upload the code to your development board.

Pre-requirements

- The Nandland GoBoard
- A micro-USB cable to power the board and upload the source code
- A VGA cable to display the game on an external display
- An external display

Step 1: Install Python

1. **Download Python**:
   Go to https://www.python.org/downloads/ and download Python 3.x.x.
2. **Run the Installer**:
   During installation, check the box for **"Add Python to PATH"**, then click **Install Now**.
3. **Verify Installation**:

```
python --version
```

Step 2: Install PIP

1. **Verify PIP Installation**:
   Run the following command in the Command Prompt:

```
pip --version
```

If you see a version number, PIP is installed. Otherwise, you can follow the official PIP installation guide here.

Step 3: Install APIO

1. **Open Command Prompt**:
   Press Win + R, type cmd, and press Enter.
2. **Install APIO**:
   Run the following command:

```
pip install apio
```

3. **Verify APIO Installation**:
   After installation, check if APIO is installed by running:

```
apio --version
```

4. **Install Required APIO Packages**:
   Install the necessary packages by running:

```
apio install system scons
```

If you're working with FPGA boards, you can also install:

```
apio install icestorm yosys arachne-pnr
```

Step 4: Install Git

1. **Download Git**:
   Download Git for Windows from https://git-scm.com/download/win.
2. **Run the Installer**:
   Follow the installation steps and use the default settings unless you need specific configurations. Git Bash can be useful for Unix-style commands.
3. **Verify Installation**:
   Run the following command in the Command Prompt:

```
git --version
```

Step 5: Clone the Project Repository

1. **Open Command Prompt or Git Bash**:
   You can use either Command Prompt or Git Bash.
2. **Navigate to Your Desired Folder**:
   Use the cd command to navigate to the folder where you want to clone the repository:

```
cd C:\Users\YourUsername\Documents
```

3. **Clone the Repository**:
   Run the following command to clone the project repository:

```
git clone https://github.com/algosup/2024-2025-project-1-fpga-team-4
```

4. **Navigate to the Cloned Folder**:
   After cloning, navigate into the cloned repository:

```
cd 2024-2025-project-1-fpga-team-4
```

Step 6: Upload the Code to Your Development Board

1. **Connect Your Board**:
   Connect your development board (FPGA or microcontroller) to the computer via USB.
2. **Verify APIO Configuration**:
   Ensure the apio.ini file is present in the cloned repository (it should be if it's already on GitHub).
3. **Upload the Code**:
   To upload the code to your board, run the following command from within the project folder:

```
apio upload
```

APIO will compile the code and upload it to the connected board.

4. **Monitor the Upload Process**:
   You should see a progress log in the Command Prompt. Once the upload completes, the code should run on your board.

Troubleshooting

- **Python or PIP Not Recognized**:
  Ensure Python and PIP are installed and added to the PATH. Reinstall if necessary, ensuring the **"Add Python to PATH"** box is checked.
- **APIO Upload Errors**:
  - Check that the board is properly connected.
  - Ensure drivers are installed correctly (refer to the Zadig installation in previous steps if required).
  - Verify that the apio.ini file is configured for the correct board.

How to install for a MacOs device:

This section explains how to install all necessary components for FROGGER™ Project on a macOS device, and how to run the project using the provided bash script.

Step 1: Obtain the Project Source Code

There are two ways to obtain the project files: **Download as a ZIP** (easier) or **Clone with Git** (for advanced users).

Option 1: Download the Repository as a ZIP File

1. **Go to the GitHub Repository Page**:
   Visit the repository page in your browser:

   ```
   https://github.com/algosup/2024-2025-project-1-fpga-team-4
   ```

2. **Download as ZIP**:
   On the GitHub page, click the green **Code** button and select **Download ZIP**.
3. **Extract the ZIP File**:
   After downloading, extract the ZIP file by double-clicking it. This will create a folder containing all the project files, including the bash script.

Option 2: Clone the Repository Using Git (for Advanced Users)

1. **Check Git Installation**:
   Git is typically installed on macOS by default. To check, run the following command in Terminal:

```
git --version
```

If Git is not installed, you'll be prompted to install it. Follow the instructions on-screen.

2. **Clone the Repository**:
   In Terminal, navigate to the folder where you want to clone the repository, then run:

```
git clone https://github.com/algosup/2024-2025-project-1-fpga-team-4
```

3. **Navigate to the Cloned Folder**:
   After cloning, navigate to the project folder by running:

```
cd 2024-2025-project-1-fpga-team-4
```

Step 4: Run the Bash Script to Upload Code

Once you've obtained the project files, either by downloading the ZIP or cloning the repository, you can run the provided bash script to upload the code to your development board.

1. **Run the Script**:
   After making the script executable, simply run the following command to execute it:

```
./Frogger
```

The bash script will handle the rest, including uploading the code to your development board.

Troubleshooting

- **Python Not Recognized**:
  If python3 is not recognized, ensure Python 3 is installed and added to your PATH. You can use Homebrew to install Python easily.
- **APIO Upload Errors**:
  - Ensure your board is properly connected to the macOS device.
  - Double-check that all necessary drivers (e.g., for FPGA boards) are installed and configured.

How to Install for Apple Devices

Step 3: Obtain the Project Source Code

Option 1: Download the Repository as a ZIP File

1. Visit the GitHub page and download the ZIP.
2. Extract the ZIP file.

Option 2: Clone with Git (for Advanced Users)

1. Check Git installation.
2. Clone the repository:

```
git clone https://github.com/algosup/2024-2025-project-1-fpga-team-4
```

Step 4: Run the Bash Script

1. Make the script executable:

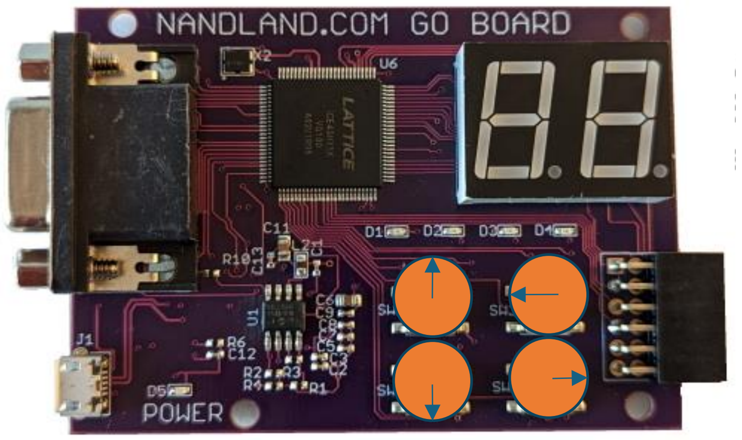
```
chmod +x Frogger
```

2. Run the script:

```
./Frogger
```

4. Game Controls

- **Button/ Switch Mappings:**



  - Top left button: actions the upwards movement of the frog
  - Bottom left button: actions the downard movement of the frog
  - Top right button: actions the left motion of the frog
  - Bottom right actions the right motion of the frog

5. Gameplay Instructions

Game Mechanics

The objective is to guide the frog to the top of the screen while avoiding obstacles, represented as red cubes (cars). As you reach the top, the difficulty increases across 8

levels. Upon completing all levels, a "GG" message will appear on the 7-segment display to congratulate you.

Levels

- **Level 1**: One moving car (red block) crosses the screen at a moderate speed.
- **Higher Levels**: The number of cars and their speed increase, making it progressively harder to reach the top.

Scoring System

- Each successful crossing adds **1 point** to the 7-segment display.
- If the frog gets hit, it respawns at the starting position without losing points.

Lives and Game Over

- **Unlimited Lives**: The player has unlimited attempts to reach the top. The challenge is to beat your previous score. The player score will be reset everytime the frog collides with a car meaning that the score will be set back to 0.

6. Tips and Strategies

Basic Tips

- **Positioning**: Stay near the center of the screen to anticipate incoming cars and react quickly.
- **Patience**: Wait for the right moment to move, especially as car speed increases in later levels.

Strategy for Success

The most effective strategy is to move cautiously and avoid rushing. Staying centered gives you the best view of obstacles and more time to react.

9. Troubleshooting

Issues

- **Game Crash (System Overflow)**:
  If the game crashes, unplug and reconnect the micro-USB cable to restart. If the issue persists, contact the development team.
- **Frog Movement Not Working**:
  If the frog isn't responding, it could be a button malfunction. Use a soft brush to clean dust or debris from the board. If the issue continues, contact the Nandland team for potential hardware issues.

Contact for Support

This game was developed by **Team 4** at **ALGOSUP International Software Development School** in Vierzon, France.

- **For game-related issues**: Submit an issue on the GitHub repository, or reach out via LinkedIn (profiles in the README.md).
- **For hardware issues**: Contact Nandland support at https://nandland.com/contact/.

8. Credits and Legal Information

- **Developer Team:**

The development of this FROGGER™ inspired game was driven by students from ALGOSUP SAS students from team 4, each playing a crucial role in bringing this project to life.

- **Project Manager:** Jason GROSSO
  Managed the day-to-day operations of the project, coordinating tasks, tracking progress, and ensuring timely delivery of milestones.
    - Linkedin: https://www.linkedin.com/in/jason-grosso-847b39251/?originalSubdomain=fr
    - GitHub: https://github.com/JasonGROSSO/JasonGROSSO
- **Program Manager:** Mathis PASCUCCI
  Oversaw the overall program, ensuring alignment with strategic goals and managing high-level coordination between stakeholders and the project team.
    - Linkedin: https://www.linkedin.com/in/mathis-pascucci-8b759732a/
    - GitHub: https://github.com/Mathis441
- **Tech Lead:** Clémentine CUREL
  Provided technical direction and oversaw the development team, ensuring that the architectural design and coding standards were followed.
    - Linkedin: https://www.linkedin.com/in/clementinecurel/
    - GitHub:https://github.com/Clementine951
- **Software Engineer:** Guillaume DERAMCHI, Victor LEROY
  Built and implemented the features and functionalities of the product, ensuring the code met both functional and technical requirements.
    - Guillaume DERAMCHI
        - Linkedin: https://www.linkedin.com/in/guillaume-deramchi/
        - GitHub: https://github.com/Guillaume18100
    - Victor LEROY:
        - Linkedin: https://www.linkedin.com/in/victor-leroy-64baa3229/
        - GitHub: https://github.com/Victor-Leroy
- **Quality Assurance (QA):** Emilien CHINSY
  Tested the product rigorously to identify and resolve bugs, ensuring the highest standards of quality were met before the product release.
    - Linkedin: https://www.linkedin.com/in/emilien-chinsy-5a794632b/
    - GitHub: https://github.com/EmilienChinsy
- **Technical Writer:** Ian LAURENT
  Created the user manual, to ensure users have clear and accurate information.
    - Linkedin: https://www.linkedin.com/in/ian-h-laurent/
    - GitHub: https://github.com/Ianlaur

We are also grateful for the contributions of external partners and consultants who helped make this product a success.

- **Licensing Information:**

2024-2025-project-1-fpga-team-4 is distributed under the MIT License, which allows you to freely use, modify, and distribute the software, including for commercial purposes. However, you must retain the original copyright notice and this permission notice in any copies or significant portions of the software.

The software is provided "as-is," with no warranties or guarantees. The developers are not responsible for any issues, damages, or liabilities that arise from using the software.

Full MIT License text: