Deployment Guide

Contents

Overview

This guide walks through deploying FastReaction V2.1.5 on a fresh Ubuntu system.

Target: Ubuntu 20.04 LTS or later
Python: 3.8+
Display: 4K (3840x2160) or 1080p (1920x1080)

Hardware Requirements

Minimum Requirements

- CPU: Dual-core 2.0 GHz or better
- RAM: 4GB
- Storage: 10GB free space
- Display: 1920x1080 minimum
- Input: Keyboard (required), Serial port (optional)
- Network: Ethernet or WiFi for API connectivity

Recommended Setup

- CPU: Quad-core 2.5 GHz
- RAM: 8GB
- Display: 4K (3840x2160) for optimal experience
- Input: Keyboard + optional external trigger device
- Network: Gigabit Ethernet

Optional Hardware

- **Serial trigger device** - For external reaction triggers (not required)
- **MQTT-enabled devices** - For remote control

- **Audio speakers** - For sound effects

Note: FastReaction does NOT require special hardware. Keyboard input is sufficient.

Software Prerequisites

System Packages

```
# Update packages
sudo apt update && sudo apt upgrade -y

# Python 3 and pip
sudo apt install -y python3 python3-pip python3-venv

# Qt5 and QML
sudo apt install -y python3-pyqt5 python3-pyqt5.qtquick python3-
pyqt5.qtmultimedia
sudo apt install -y qt5-qmltooling-plugins qml-module-qtquick-controls2
sudo apt install -y qml-module-qtmultimedia qml-module-qtgraphicaleffects

# Audio
sudo apt install -y pulseaudio alsa-utils

# X11 (for GUI)
sudo apt install -y x11-xserver-utils xorg

# Serial port tools (optional - only if using external triggers)
sudo apt install -y python3-serial setserial

# MQTT broker (if local)
sudo apt install -y mosquitto mosquitto-clients

# Dev tools (optional)
sudo apt install -y git vim curl wget
```

Python Packages

```
# Install from requirements.txt
cd /path/to/FastReaction
pip3 install -r requirements.txt

# Or install manually
pip3 install PyQt5==5.15.9
pip3 install paho-mqtt==1.6.1
pip3 install requests==2.28.2
pip3 install pyserial==3.5
pip3 install pygame==2.1.3
pip3 install numpy==1.21.6
```

Initial System Setup

1. Create User Account

```
# Create application user
sudo adduser eaa25-game2
sudo usermod -aG sudo eaa25-game2
sudo usermod -aG dialout eaa25-game2   # For serial port (if used)

# Switch to application user
su - eaa25-game2
```

2. Configure Display

```
# Set display resolution
xrandr --output HDMI-1 --mode 3840x2160   # For 4K
# or
xrandr --output HDMI-1 --mode 1920x1080   # For 1080p

# Auto-start X server on boot
echo "startx" >> ~/.bashrc
```

3. Configure Audio (if needed)

```
# Test audio
speaker-test -t sine -f 1000 -l 1

# Set default audio device
pactl set-default-sink alsa_output.pci-0000_00_1f.3.analog-stereo
```

Application Installation

1. Copy Application Files

```
# Create application directory
sudo mkdir -p /home/eaa25-game2/FastReaction
sudo chown eaa25-game2:eaa25-game2 /home/eaa25-game2/FastReaction

# Copy application files
sudo cp -r /path/to/source/FastReaction/* /home/eaa25-game2/FastReaction/

# Set proper ownership
sudo chown -R eaa25-game2:eaa25-game2 /home/eaa25-game2/FastReaction
```

2. Verify File Structure

```
cd /home/eaa25-game2/FastReaction
ls -la

# Expected structure:
# Fast_Reaction_V2.1.5.py        - Main application
# config.py                       - Configuration
# api/                            - API module
# utils/                          - Utilities
# *.qml                           - QML UI files
# *_backend.py                    - QML backends
# Assets/                         - Fonts, images, audio
# teams/                          - JSON backups
# logs/                           - Log files
```

3. Create Required Directories

```
# Create logs directory
mkdir -p /home/eaa25-game2/FastReaction/logs

# Create teams directory
mkdir -p /home/eaa25-game2/FastReaction/teams

# Set permissions
chown -R eaa25-game2:eaa25-game2 /home/eaa25-
game2/FastReaction/{logs,teams}
```

Systemd Service Setup

1. Create Service File

Create `/etc/systemd/system/fastreaction.service`:

```
[Unit]
Description=FastReaction Game Service V2.1.5
After=network.target graphical-session.target
Wants=network.target

[Service]
Type=simple
User=eaa25-game2
WorkingDirectory=/home/eaa25-game2/FastReaction
ExecStart=/usr/bin/python3 /home/eaa25-
game2/FastReaction/Fast_Reaction_V2.1.5.py
```

```
Restart=always
RestartSec=5
StandardOutput=journal
StandardError=journal

# Environment variables for GUI applications
Environment=DISPLAY=:0
Environment=XDG_RUNTIME_DIR=/run/user/1000
Environment=PULSE_RUNTIME_PATH=/run/user/1000/pulse
Environment=QT_QPA_PLATFORM=xcb
Environment=QML_IMPORT_PATH=/usr/lib/x86_64-linux-gnu/qt5/qml

# Security settings
NoNewPrivileges=true
PrivateTmp=true

[Install]
WantedBy=multi-user.target
```

2. Install Service

```
# Copy service file
sudo cp services/fastreaction.service /etc/systemd/system/

# Set permissions
sudo chmod 644 /etc/systemd/system/fastreaction.service

# Reload systemd
sudo systemctl daemon-reload

# Enable service
sudo systemctl enable fastreaction.service
```

Configuration

1. Update API Settings

```
# Edit config.py
nano /home/eaa25-game2/FastReaction/config.py

# Verify:
# - API credentials (email, password)
# - API base URL (production/development)
# - Game ID (UUID)
# - Game name
```

2. Configure Game Timer

```
# In config.py, adjust timer value
timer_value = 60000  # 60 seconds in milliseconds
final_screen_timer = 5000  # 5 seconds
```

3. Serial Port Configuration (Optional)

```
# Only if using external trigger device
# Find serial port
ls -l /dev/ttyUSB*

# In config.py:
enabled = True  # Change to True if using serial
port = "/dev/ttyUSB0"  # Adjust port name
```

4. MQTT Configuration

```
# In config.py, verify MQTT settings
broker = "localhost"
port = 1883

# Test MQTT connection
mosquitto_pub -h localhost -t "FastReaction/game/test" -m "hello"
```

Testing

1. Test Application Manually

```
# Run application directly
cd /home/eaa25-game2/FastReaction
python3 Fast_Reaction_V2.1.5.py

# Check for errors in terminal
# Verify:
# - Window opens
# - QML screens load
# - Audio works (if configured)
# - Keyboard input works
```

2. Test API Connection

```
# Check API connectivity
python3 -c "
from api.game_api import GameAPI
api = GameAPI()
if api.authenticate():
    print('API connection OK')
else:
    print('API connection failed')
"
```

3. Test MQTT (if configured)

```
# Subscribe to test topic
mosquitto_sub -h localhost -t "FastReaction/game/#" &

# Publish test message
mosquitto_pub -h localhost -t "FastReaction/game/start" -m "test"
```

Production Deployment

1. Environment Variables (Optional)

```
# Create environment file
sudo nano /etc/environment

# Add:
CAGE_API_BASE_URL="https://prod-eaa25-api.azurewebsites.net/"
CAGE_TIMER_VALUE="60000"
```

2. Start Production Service

```
# Enable service for auto-start
sudo systemctl enable fastreaction.service

# Start service
sudo systemctl start fastreaction.service

# Verify running
sudo systemctl status fastreaction.service

# Check logs
sudo journalctl -u fastreaction.service -n 50
```

3. Monitor Service

```
# Real-time log monitoring
sudo journalctl -u fastreaction.service -f

# Check service uptime
sudo systemctl status fastreaction.service | grep Active

# View recent errors
sudo journalctl -u fastreaction.service -p err -n 20
```

Service Management Commands

Direct systemctl Commands

```
# Start
sudo systemctl start fastreaction.service

# Stop
sudo systemctl stop fastreaction.service

# Restart
sudo systemctl restart fastreaction.service

# Status
sudo systemctl status fastreaction.service

# Enable auto-start
sudo systemctl enable fastreaction.service

# Disable auto-start
sudo systemctl disable fastreaction.service

# View logs
sudo journalctl -u fastreaction.service -n 50

# Follow logs
sudo journalctl -u fastreaction.service -f
```

Troubleshooting

Common Issues

Application Won't Start

```
# Check Python dependencies
pip3 list | grep PyQt5
pip3 list | grep paho-mqtt

# Check file permissions
ls -la /home/eaa25-game2/FastReaction/Fast_Reaction_V2.1.5.py

# Check display
echo $DISPLAY
xrandr
```

API Connection Fails

```
# Test network connectivity
ping prod-eaa25-api.azurewebsites.net

# Check API credentials in config.py
nano /home/eaa25-game2/FastReaction/config.py

# Test API manually
python3 -c "from api.game_api import GameAPI; api = GameAPI();
print(api.authenticate())"
```

MQTT Issues

```
# Check MQTT broker status
sudo systemctl status mosquitto

# Test MQTT connection
mosquitto_pub -h localhost -t test -m "hello"
mosquitto_sub -h localhost -t test

# Check MQTT config in config.py
nano /home/eaa25-game2/FastReaction/config.py
```

Audio Not Working

```
# Check PulseAudio
pulseaudio --check
pulseaudio -D

# List audio devices
pactl list sinks
```

```
# Test audio
speaker-test -t sine -f 1000 -l 1
```

Serial Port Issues (if using external triggers)

```
# Check serial port exists
ls -l /dev/ttyUSB*

# Check permissions
sudo usermod -aG dialout eaa25-game2

# Test serial communication
sudo python3 -c "
import serial
ser = serial.Serial('/dev/ttyUSB0', 9600, timeout=1)
print('Serial OK')
ser.close()
"
```

FastReaction Deployment Notes

Hardware Requirements

- **No special hardware required** - Keyboard is sufficient
- **Optional serial port** - For external trigger devices only
- **Display**: 4K (3840x2160) recommended, 1080p minimum
- **Audio**: Optional but enhances gameplay experience

Service Configuration

- Service name: `fastreaction.service`
- Service user: `eaa25-game2`
- Main application: `Fast_Reaction_V2.1.5.py`
- Working directory: `/home/eaa25-game2/FastReaction`

Configuration Requirements

- **Timer settings** - Configure game duration
- **API credentials** - Required for cloud integration
- **Serial port** - Optional, disabled by default
- **MQTT broker** - Optional for remote control

Testing Focus

- **Keyboard input** - Verify spacebar/arrow keys work
- **API connectivity** - Test authentication and polling
- **Display output** - Check resolution and QML rendering

- **Audio playback** - Verify sound effects work
- **Timer accuracy** - Ensure countdown is precise

Security Checklist

- ☐ Firewall configured (allow API, MQTT if needed)
- ☐ SSH key-based authentication
- ☐ Application user has minimal permissions
- ☐ Service runs as non-root user
- ☐ Logs are rotated and archived
- ☐ API credentials in environment variables (not hardcoded)
- ☐ MQTT uses authentication (if accessible remotely)
- ☐ System updates applied regularly

Backup Strategy

What to Backup

```
# Application code
/home/eaa25-game2/FastReaction/*.py
/home/eaa25-game2/FastReaction/*.qml
/home/eaa25-game2/FastReaction/config.py

# Team data (JSON backups)
/home/eaa25-game2/FastReaction/teams/*.json

# Logs (if needed)
/home/eaa25-game2/FastReaction/logs/*.csv

# Service file
/etc/systemd/system/fastreaction.service
```

Backup Script Example

```bash
#!/bin/bash
# backup_fastreaction.sh

BACKUP_DIR="/backup/fastreaction_$(date +%Y%m%d)"
mkdir -p $BACKUP_DIR

# Backup application
cp -r /home/eaa25-game2/FastReaction $BACKUP_DIR/

# Compress
tar -czf $BACKUP_DIR.tar.gz $BACKUP_DIR
rm -rf $BACKUP_DIR
```

```
echo "Backup created: $BACKUP_DIR.tar.gz"
```

Performance Tuning

Display Optimization

```
# Disable compositor (reduce latency)
killall compton

# Set performance governor
sudo cpufreq-set -g performance
```

Audio Latency

```
# Reduce audio latency in /etc/pulse/daemon.conf
default-fragments = 2
default-fragment-size-msec = 5
```

Monitoring

Log Rotation

```
# Create /etc/logrotate.d/fastreaction
/home/eaa25-game2/FastReaction/logs/*.csv {
    weekly
    rotate 4
    compress
    missingok
    notifempty
}
```

Health Check Script

```
#!/bin/bash
# health_check.sh

# Check if service is running
if systemctl is-active --quiet fastreaction.service; then
    echo "FastReaction service: OK"
else
    echo "FastReaction service: FAILED"
```

```
        sudo systemctl restart fastreaction.service
fi

# Check API connectivity
python3 -c "from api.game_api import GameAPI; api = GameAPI(); exit(0 if
api.authenticate() else 1)"
if [ $? -eq 0 ]; then
    echo "API connection: OK"
else
    echo "API connection: FAILED"
fi
```

Production Checklist

- ☐ Application installed in `/home/eaa25-game2/FastReaction`
- ☐ Python dependencies installed
- ☐ Qt5/QML packages installed
- ☐ config.py configured with production API
- ☐ systemd service created and enabled
- ☐ Display resolution configured
- ☐ Audio tested (if used)
- ☐ API connection tested
- ☐ MQTT tested (if used)
- ☐ Serial port configured (if used)
- ☐ Logs directory created
- ☐ Teams directory created
- ☐ Service starts on boot
- ☐ Backup strategy implemented
- ☐ Monitoring configured