

# Fausse Nostalgie v1.0

A retro-inspired terminal clock and calendar application that recreates the aesthetic of IBM terminals from the 1980s and 1990s.

---

## About This Project

**Fausse Nostalgie** (French for "False Nostalgia") is a terminal-based clock and calendar application designed to evoke the look and feel of vintage IBM office terminals. This was my first Python project, created as a self-directed learning exercise to explore terminal manipulation, ANSI escape sequences, and time handling in Python.

## Development Background

- **Built with:** Python 3.9+
- **Development environment:** CachyOS Linux
- **Code editor:** Kate
- **Tested on:** Konsole, Kitty, Alacritty
- **AI assistance:** Developed with support from ChatGPT and Claude for error checking and debugging
- **Status:** First release - whilst thoroughly tested on Linux, macOS and Windows compatibility is untested

**Important:** This application was built and tested exclusively on Linux. Whilst it should work on macOS, Windows support is currently unavailable due to platform-specific terminal handling requirements. If you encounter bugs on macOS, please report them as issues.

---

## Features

- **Large digital clock** with ASCII art display
  - **Interactive calendar** with month/year navigation
  - **Multi-timezone support** - track time across 9 global cities
  - **8 colour schemes** - red, amber, green, ice blue, dark blue, dark green, grey, white
  - **12/24-hour format toggle**
  - **Retro IBM terminal aesthetic** with authentic styling
  - **Terminal resize support** - adapts to window changes
  - **Flicker-free rendering** using double-buffering
- 

## Controls

### Global Controls

Key	Action
<div>Tab</div>	Switch between clock and calendar
<div>T</div>	Toggle 12-hour / 24-hour time format
<div>C</div>	Cycle through colour schemes
<div>1-9</div>	Toggle timezone display (press again to remove)
<div>0</div>	Reset to local time only
<div>Ctrl+C</div>	Exit application

### Calendar Mode

Key	Action
<div>H</div>	Previous month
<div>L</div>	Next month
<div>J</div>	Previous year
<div>K</div>	Next year

---

### Included Timezones

The application includes 9 timezones spanning the globe:

Key	City	GMT Offset
①	Los Angeles	GMT-8
②	New York	GMT-5
③	São Paulo	GMT-3
④	London	GMT+0
⑤	Paris	GMT+1
⑥	Cairo	GMT+2
⑦	Dubai	GMT+4
⑧	Tokyo	GMT+9
⑨	Sydney	GMT+11

## Installation

### Prerequisites

- **Python 3.9 or higher** (required for `zoneinfo` support)
- **Unix-like terminal** (Linux, macOS, BSD)

### Linux Installation

#### Arch Linux & Derivatives (Manjaro, EndeavourOS, CachyOS, etc.)

```
bash

# Install Python and pip
sudo pacman -S python python-pip

# Install pyfiglet
pip install pyfiglet

# Clone the repository
git clone https://github.com/Alex-Bowser-Dev/fausse-nostalgie.git
cd fausse-nostalgie

# Run the application
python fausse_nostalgie.py
```

## Debian & Derivatives (Ubuntu, Linux Mint, Pop!\_OS, etc.)

```
bash

# Install Python and pip
sudo apt update
sudo apt install python3 python3-pip

# Install pyfiglet
pip3 install pyfiglet

# Clone the repository
git clone https://github.com/Alex-Bowser-Dev/fausse-nostalgie.git
cd fausse-nostalgie

# Run the application
python3 fausse_nostalgie.py
```

## Red Hat & Derivatives (Fedora, CentOS, Rocky Linux, etc.)

```
bash

# Install Python and pip
sudo dnf install python3 python3-pip

# Install pyfiglet
pip3 install pyfiglet

# Clone the repository
git clone https://github.com/Alex-Bowser-Dev/fausse-nostalgie.git
cd fausse-nostalgie

# Run the application
python3 fausse_nostalgie.py
```

## Gentoo

```
bash
```

```
# Install Python (usually pre-installed)
```

```
emerge --ask dev-lang/python
```

```
# Install pip
```

```
emerge --ask dev-python/pip
```

```
# Install pyfiglet
```

```
pip install pyfiglet
```

```
# Clone the repository
```

```
git clone https://github.com/Alex-Bowser-Dev/fausse-nostalgie.git
```

```
cd fausse-nostalgie
```

```
# Run the application
```

```
python fausse_nostalgie.py
```

## FreeBSD Installation

```
bash
```

```
# Install Python and pip
```

```
pkg install python3 py39-pip
```

```
# Install pyfiglet
```

```
pip install pyfiglet
```

```
# Clone the repository
```

```
git clone https://github.com/Alex-Bowser-Dev/fausse-nostalgie.git
```

```
cd fausse-nostalgie
```

```
# Run the application
```

```
python3.9 fausse_nostalgie.py
```

## macOS Installation

```
bash
```

```
# Install Homebrew (if not already installed)
```

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/i
```

```
# Install Python
```

```
brew install python
```

```
# Install pyfiglet
```

```
pip3 install pyfiglet
```

```
# Clone the repository
```

```
git clone https://github.com/Alex-Bowser-Dev/fausse-nostalgie.git
```

```
cd fausse-nostalgie
```

```
# Run the application
```

```
python3 fausse_nostalgie.py
```

## Windows Installation

**Note:** This application currently requires Unix-specific terminal features (`termios`, `tty`, `signal.SIGWINCH`) and **will not run natively on Windows**.

For Windows users, consider these alternatives:

- **WSL2 (Windows Subsystem for Linux)** - Recommended

```
powershell
```

```
# Install WSL2 (PowerShell as Administrator)
```

```
wsl --install
```

```
# After restart, open WSL and follow Debian/Ubuntu instructions above
```

- **Git Bash / MinTTY** - May have limited functionality
- **Cygwin** - Provides Unix-like environment

---

## Customisation Guide

### Adding or Modifying Colour Schemes

Colours are defined using RGB values in the `PALETTES` dictionary (around line 99). To add a new colour:

1. Add your colour to the `PALETTES` dictionary:

python

```
PALETTES = {
    "red": {"FG": "\033[38;2;255;80;80m", "BG": "\033[48;2;0;0;0m"},
    # ... existing colours ...
    "purple": {"FG": "\033[38;2;200;100;255m", "BG": "\033[48;2;0;0;0m"}, # New!
}
```

2. Add the colour name to `PALETTE_ORDER` (around line 112):

python

```
PALETTE_ORDER = [
    "red",
    "amber",
    # ... existing colours ...
    "purple", # New!
]
```

**RGB Format:** `\033[38;2;R;G;Bm` where R, G, B are values from 0-255

- `FG` = Foreground (text colour)
- `BG` = Background colour (default is black: `0;0;0`)

## Changing Timezones

Timezones are defined in the `TIMEZONES` list (around line 39). To modify:

1. Find the IANA timezone name from the [tz database](#)
2. Modify the entry:

python

```
TIMEZONES = [
    {"name": "LA", "tz": ZoneInfo("America/Los_Angeles"), "gmt": "GMT-8"},
    {"name": "BERLIN", "tz": ZoneInfo("Europe/Berlin"), "gmt": "GMT+1"}, # Changed!
    # ... rest of timezones ...
]
```

### Key components:

- `name` - Display name (keep it short, max 10 characters)
- `tz` - IANA timezone identifier
- `gmt` - GMT offset for display purposes

## Changing Default Colour

Modify `palette_index` (around line 120):

```
python

palette_index = 2 # 0=red, 1=amber, 2=green, 3=ice_blue, etc.
```

## Changing Status Messages

Edit the `STATUS_MESSAGES` list (around line 65):

```
python

STATUS_MESSAGES = [
    "TIME SYNCED",
    "CALENDAR ONLINE",
    "YOUR CUSTOM MESSAGE", # Add your own!
]
```

Messages rotate every 15 seconds (configurable via `STATUS_INTERVAL`).

## Adjusting Frame Width

Change `BASE_WIDTH` (around line 18):

```
python

BASE_WIDTH = 76 # Default width - increase for wider terminals
```

**Note:** Wider frames require larger terminal windows.

---

## Known Issues & Limitations

1. **Windows compatibility** - Not currently supported due to Unix-specific terminal requirements
  2. **macOS testing** - Untested; please report any issues you encounter
  3. **Terminal size** - Minimum recommended: 80x24 characters
  4. **UTF-8 encoding** - Some terminals may display timezone names incorrectly
  5. **Python version** - Requires Python 3.9+ for `zoneinfo` support
- 

## Contributing

This is a learning project, but contributions are welcome! If you find bugs or have suggestions:

1. **Fork the repository**
2. **Create a feature branch** (`git checkout -b feature/amazing-feature`)
3. **Commit your changes** (`git commit -m 'Add some amazing feature'`)
4. **Push to the branch** (`git push origin feature/amazing-feature`)
5. **Open a Pull Request**

## Particularly Helpful Contributions

- Windows compatibility layer
  - macOS testing and bug reports
  - Additional colour schemes
  - Performance optimisations
  - Documentation improvements
- 

## Licence

This project is licenced under the **MIT Licence** - see the [LICENSE](#) file for details.

**TL;DR:** You're free to use, modify, and distribute this software, even for commercial purposes, as long as you include the original copyright notice.

---

## Acknowledgements

- **IBM** - For the iconic terminal aesthetics that inspired this project
  - **pyfiglet** - For the ASCII art text generation
  - **ChatGPT & Claude (Anthropic)** - For AI assistance in debugging and error correction
  - **The Python Community** - For excellent documentation and learning resources
- 

## Contact

**Author:** Alex Bowser

**Project Link:** <https://github.com/Alex-Bowser-Dev/fausse-nostalgie>

---

Enjoy your retro terminal experience!

