

Project Conway: Getting Started Guide

This is a document to support the usage of software required in the running of Project Conway, although other platforms will work with manual setup, all installation scripts and documentation were written for a Debian based system.

Project Conway should be considered two separate applications: *Web Application* and *Display Adapter*.

Web Application – Explains what Project Conway is, teaches the user about Conway's Game of Life and allows them to input their pattern and choose a time to simulate their pattern on the display.

Display Adapter – Receives the user's patterns from the web application and display's it on the display at the correct time, this is considered "run mode". Whilst there are no user's patterns to display, the software will go into "screensaver mode".

This guide assumes some knowledge on using a linux terminal, please see: <https://help.ubuntu.com/community/UsingTheTerminal> if you do not feel confident.

Table of Contents

Project Conway: Getting Started Guide.....	1
Web Application.....	2
Requirements.....	2
Getting the software.....	2
Installation.....	2
Setup the software.....	3
Running the software.....	4
Display Adapter.....	6
Requirements.....	6
Getting the software.....	6
Installation.....	6
Setup the software.....	7
Running the software.....	8

Web Application

Requirements

- Debian based platform
 - an account with sudo privileges is required for installation, if *virtualenv* is not already installed
 - it is possible to use other *nix based systems but it will require manual set-up
- Python 3
- GNU make
- GIT (source code management)

Getting the software

Guide

- Open a terminal
- Change directory to a location that you can safely run software from:

```
cd <filepath of a location you can run software from>
```

- Use git to download the software:

```
git clone https://github.com/CO600GOL/Game_of_life.git
```

Outcome

You should now have a folder named "Game_of_life" in your directory, this folder contains all the necessary code for Project Conway.

Installation

Guide

- Change directory to the ProjectConway code

```
cd <file path chosen previously>/Game_of_life/ProjectConway
```

- Use the command:

```
make setup
```

Outcome

If all goes well, this will create a virtual environment which will have the software required to run the web application. This folder will be:

<file path chosen previously>/Game_of_life/ProjectConway/env

Setup the software

Project Conway will likely require you to set up some variables to meet your needs. I will now talk you through them.

Config file:

<file path chosen previously>/Game_of_life/ProjectConway/projectconway/__init__.py

File type:

This is a python file, the configuration section is the python dictionary named "project_config"

Parameters:

Name	Type	Description
display_name	string	This is a string representing the location in which the display is currently situated. Displayed on scheduler page.
display_address	string	This is a string representing the address of the location the display is currently situated. Displayed on scheduler page.
display_link	string	This is a string containing a link to a website giving more information about the location or the display. Displayed on scheduler page.
display_picture	string	This is a string containing a filepath or link to a picture representing where the display is situated.
start_date	None or datetime.date	None: this means the web application will accept patterns from the current date of when the pattern is submitted. datetime.date: this means the web application will only accept patterns from a given date.
date_range	None or datetime.timedelta	None: this means there is no cap on the date that that the pattern will be scheduled datetime.timedelta: this means there will be a cap on when patterns are accepted for. This parameter takes into consideration the value that was given to "start_date"
starting_time	datetime.time	This object represents the time of day that patterns will start to be accepted from.
closing_time	datetime.time	This object represents the time of day that patterns will stop being accepted from.
scheduling_gap	datetime.timedelta	This object represents the time gap in which a pattern is allowed to run, i.e. the time slice that is dedicated to a customer's pattern.

Config file:

<file path chosen previously>/Game_of_life/GameEngine/game_of_life/__init__.py

File type:

This is a python file, the file contains variables used the web application.

Parameters:

Name	Type	Description
X_CELLS	integer	This is a number that represents the number of horizontal cells allowed in the grid.
Y_CELLS	integer	This is a number that represents the number of vertical cells allowed in the grid.

Config file:

<file path chosen previously>/Game_of_life/ProjectConway/projectconway/development.ini or production.ini

File type:

These are pyramid (the web framework used in Project Conway) configuration files. There are a lot of options in this file that would take a tremendous amount of effort to explain, so I will explain options that are important to Project Conway. Find further documentation here:

<http://docs.pylonsproject.org/projects/pyramid/en/latest/narr/project.html>

Parameters:

Name	Type	Description
sqlalchemy.url	string	This is string that represents a URL path to a database that the uses.
host	string	This in an IP address which the web application listens too. "0.0.0.0" means accept requests from any machine.
port	string	This is a number representing the TCP port in which connections are accepted on.

Running the software

All methods of running the software assume you have have a terminal open in the ProjectConway folder, as mentioned above. You must also have already completed the Installation stage.

Testing

```
make run_develop
```

This command will execute the web application given the variables in development.ini. To stop the site running in this mode press: "Ctrl + c"

Running

```
make run
```

This command will execute the web application given the variables in production.ini. To stop the site running in this mode press: "Ctrl + c"

Daemon mode

```
make daemon
```

This will install the web application as a Linux daemon. This means that it will begin to run in the background of the operating system, starting automatically when the computer is started. This mode uses the variables in production.ini.

To start the daemon use:

```
sudo service projectconway start
```

To stop the daemon use:

```
sudo service projectconway stop
```

In this mode, output is logged to: /var/log/projectconway.log

Display Adapter

Requirements

- Debian based platform
 - an account with sudo privileges is required for installation, if *virtualenv* is not already installed
 - an account with privileges to read and write to serial services. See: https://fedorahosted.org/fldigi/wiki/Documentation/HOWTO/Serial_Port_Setup
 - it is possible to use other *nix based systems but it will require manual set-up
- Python 3
- GNU make
- GIT (source code management)

Getting the software

Guide

- Open a terminal
- Change directory to a location that you can safely run software from:

```
cd <filepath of a location you can run software from>
```

- Use git to download the software:

```
git clone https://github.com/CO600GOL/Game_of_life.git
```

Outcome

You should now have a folder named “Game_of_life” in your directory, this folder contains all the necessary code for the Display Adapter.

Installation

Guide

- Change directory to the DisplayAdapter code

```
cd <file path chosen previously>/Game_of_life/DisplayAdapter
```

- Use the command:

```
make setup
```

Outcome

If all goes well, this will create a virtual environment which will have the software required to run the web application. This folder will be:

<file path chosen previously>/Game_of_life/DisplayAdapter/display_env

Setup the software

The DisplayAdapter will require some setup in order to work correctly with the web application, this will require documenting:

Config file:

<file path chosen previously>/Game_of_life/DisplayAdapter/display_adapter/__init__.py

File type:

This is a python file, the file contains variables used in the DisplayAdapter.

Parameters:

Name	Type	Description
db_name	string	This is a string representing the name of the sqlite3 database that will be created in <file path chosen previously>/Game_of_life/DisplatAdapter. This file will contain all the user's patterns that will be downloaded from the web application.
db_reciever_url	string	This is a string representing the URL path to receive the users patterns from the web application, this will need to be changed to work with the web application you previously set up.
db_reciever_polling_period	integer	This integer represents the polling period time for downloading user's patterns from the web application. The default value is every 1 minutes.
serial_name	string	This is a string representing the serial file path for the display. This is "/dev/ttyACM0" by default. Find more info here: http://www.cyberciti.biz/faq/find-out-linux-serial-ports-with-setserial/
serial_baudrate	integer	This is an integer which represents the baud rate in which to communicate with the pre-set rate of the display. This is 9600 by default.
sleep_time	float	This is the amount of time each turn is displayed on the display, in seconds. Changing this value would change the number of turns that can happen in a user's time slice, it would also require updating to the user feedback in the web application.
runmode_config dictionary		
iterations	integer	This is an integer that represents the number of times a customer's pattern is flashed on the screen before the evolution commences.

full_frames	integer	This is an integer that represents the number of frames a full screen will be displayed for per flash.
pattern_frames	integer	This is an integer that represents the number of frames the user's pattern is displayed per flash.
screensaver_config dictionary		
pause_frames	integer	This is an integer that represents the the number of frames the previous frame is paused for before the screen saver wipes away the display.

Running the software

This section assumes you have successfully installed the DisplayAdapter and have a terminal open in the location in which it is installed, please see above.

Running

```
make run
```

This will run the DisplayAdapter given the variables mentioned above. To leave this program, use: "Ctrl + c"