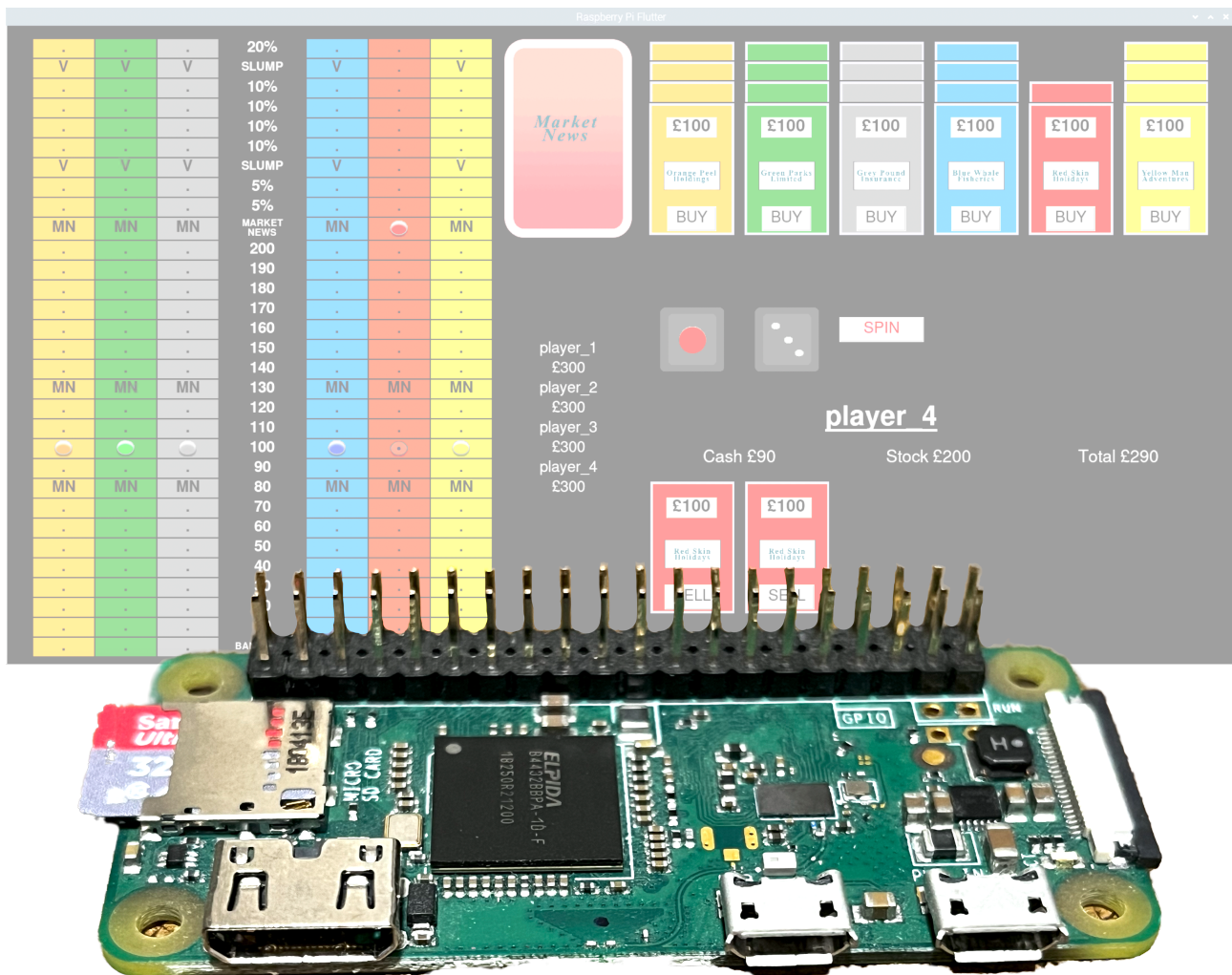


RPi zero-w Flutter game (pifl)



Introduction

This is a game originally produced by Spears which was taken over by Mattel in 1994, but this stock market trading game appears to stop being available sometime in the 1970s. Consequently the only way to purchase it in its original board format is on the second hand market.

Having enjoyed playing Flutter with friends, and finding it difficult to purchase an original, I decided to code a TV screen version for the Raspberry Pi.

The Game

Basically this is a game where you gamble on stocks to make a profit. You start off with £300 and through buying and selling one or more of the six available stocks you aim to increase your overall worth. The first player to achieve an overall value of £600 or more wins the game. There are plenty of gotchas provided by the Market News cards and your stocks will decrease in value if they haven't reached the profit section (above Market News) before the market closes. This happens when any stock reaches the top row.

Implementation

Moving a board game to a TV screen has its challenges. Apart from the issue of displaying the state of the game's board and that of the individual players, there is also the problem of managing the interactions of those taking part.

When designing the TV screen layout its limitations on space meant reducing the number of players to 6 (from 8) became necessary.

It had to be possible to control all aspects of the game using a web browser so each player could interact using their individual device (iPad, smart phone or computer). Also the game had to be self starting and subsequently controlled without direct access to the RPi's operating/window system. With this in mind the Rpi was configured to provide a WiFi access point (AP) on power up.

The programme itself is written in Python using Tkinter for the graphics display and interactions.

Installation

The software design assumes its installation on a dedicated SD card as the game is automatically loaded on boot.

Before the games installation script can be run the SD card must be loaded with Raspberry Pi OS and desktop and this must be configured with an internet connection.

The game is distributed in a folder called 'FlutterBuild'. This folder contains all the elements which must be installed on the Rpi to make the game function.

The main executable is contained in a bundled package containing all the Python libraries and support files. This makes it independent of the Rpi's installed state.

The FlutterBuild directory is distributed as a Zip file which must be unzipped to a location of your choosing (I use ~/Desktop). Within the 'FlutterBuild' directory is a shell script called 'FlutterBuild.sh'. This script installs the game and its environment. Before the shell script can be run it must be made executable. To do this open a terminal session, change directory to the FlutterBuild directory and type:-

```
sudo chmod 755 FlutterBuild.sh
```

Then to run the script type:-

```
./FlutterBuild.sh
```

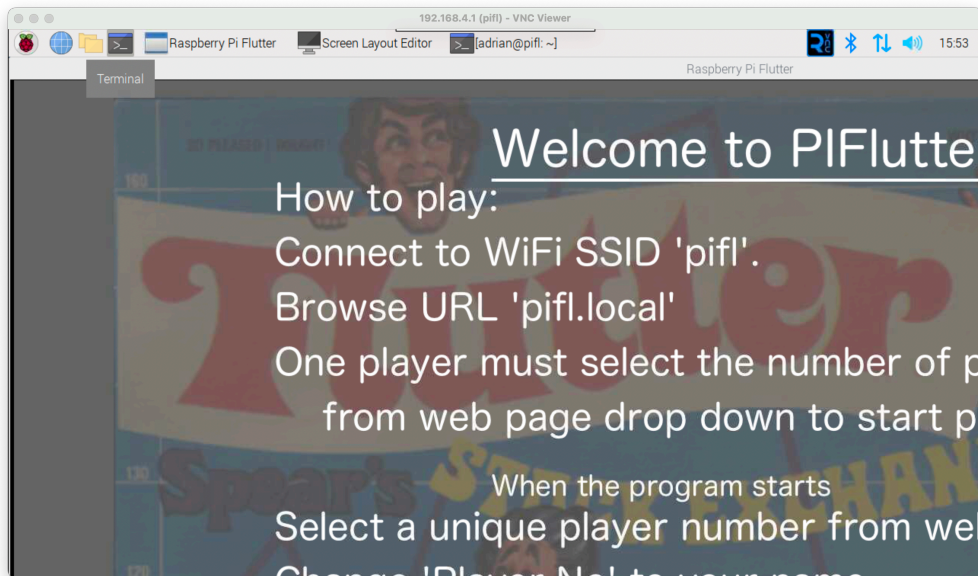
As the script runs it does the following:

1. Creates an autostart file to start FlutterV241 on boot.
2. Installs Apache and PHP for web browser support.
3. Copies Flutter's web browser support files to /var/www/html directory.
4. Copies the FlutterV241 package to the Desktop.
5. Creates dnsmasq.conf file to provide a range of dhcp addresses.
6. Creates hostapd.conf to define Access Point.
7. Changes hostname to 'pifl'.
8. Enables new dnsmasq and hostapd.

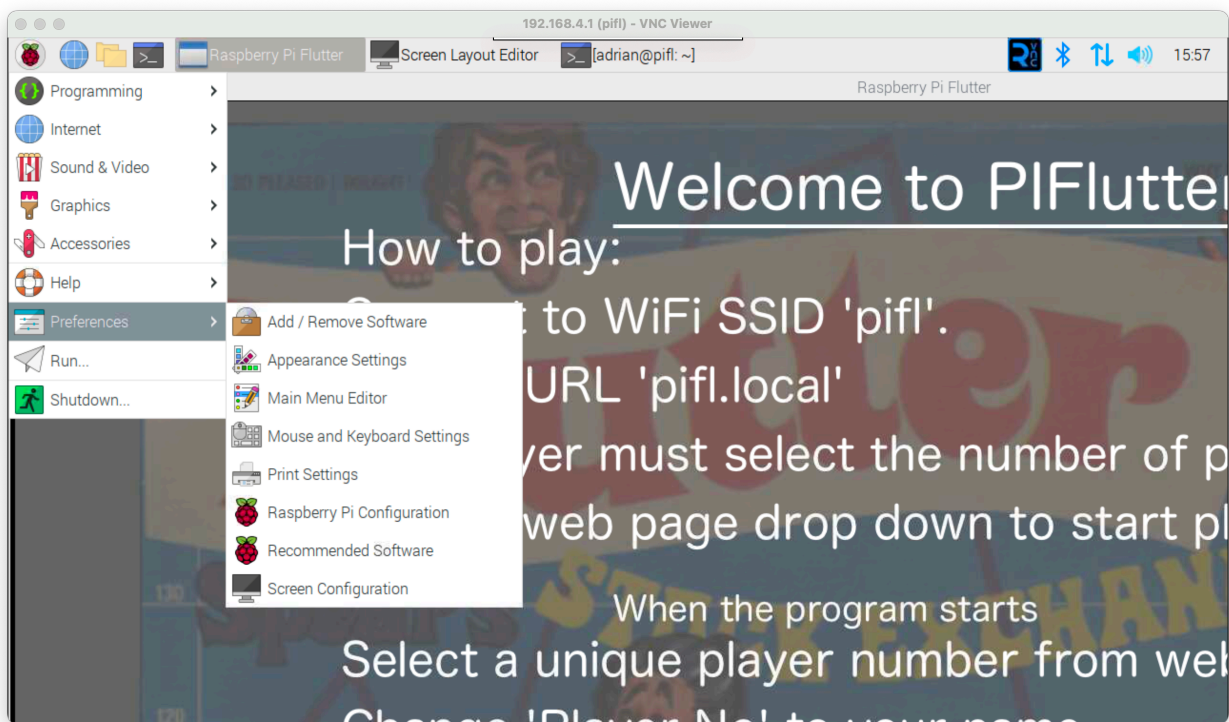
Where the script changes configuration files it keeps copies with the word 'saved' added to the name. It can be re-run without overwriting these saved configuration files.

On completion the script says 'Now Reboot'

When the RPi restarts and enters the desktop environment, the FlutterV241 programme is unpacked and loads. This takes about 2.5 minutes from starting a reboot. If PiFlutter doesn't fit on the screen (as shown below) then the resolution needs setting.

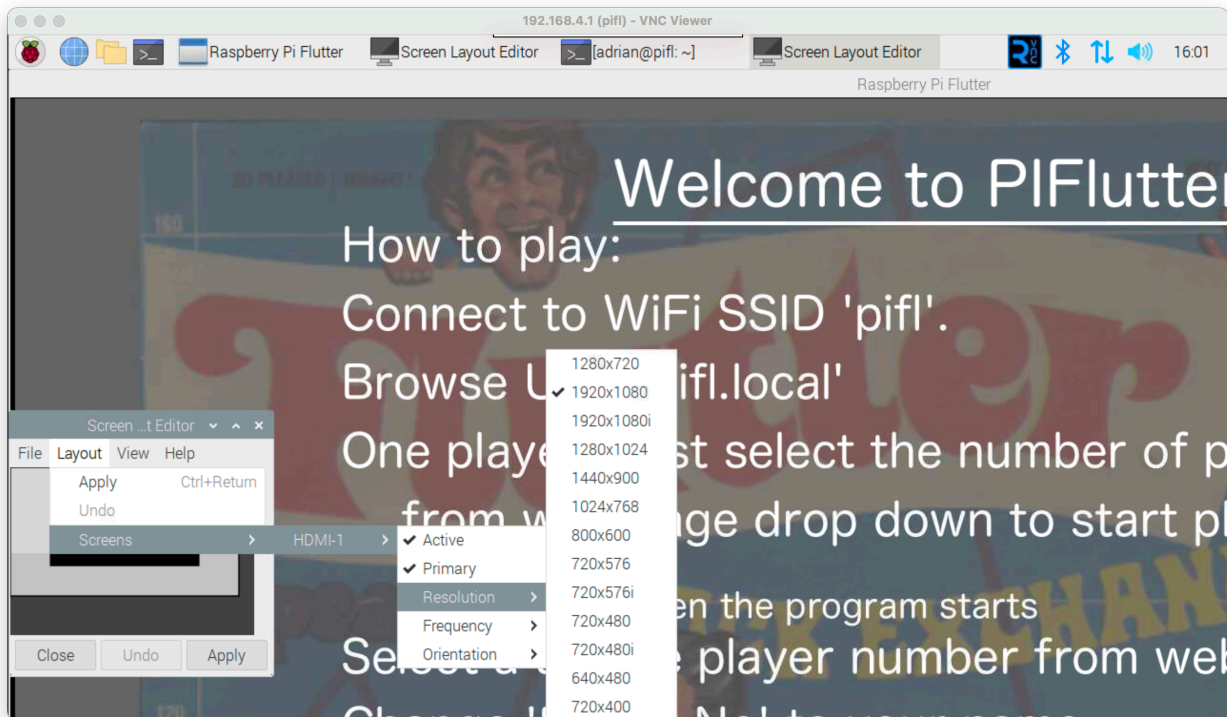


This can be done using menu>Preferences>Screen Configuration

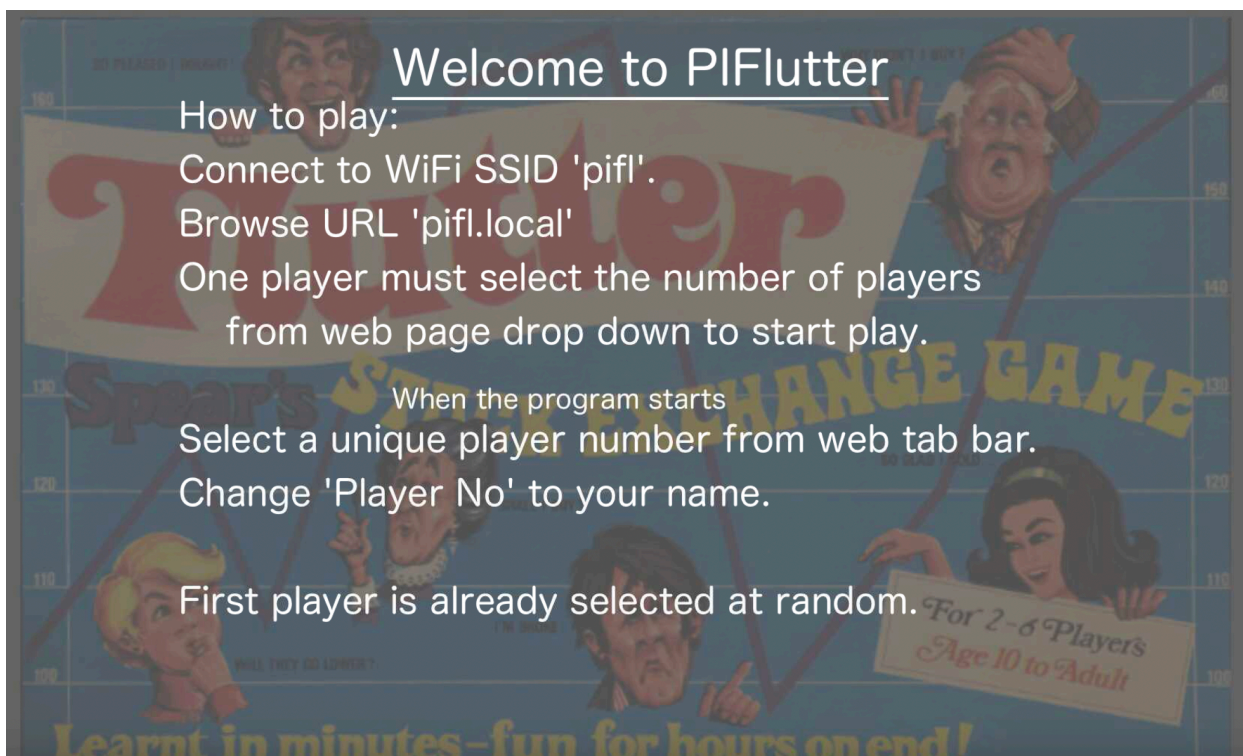


Follow the menus as shown below and select 1920x1080. Click Apply and accept the change when prompted.

Using this method will ensure the resolution change is kept for subsequent reboots.

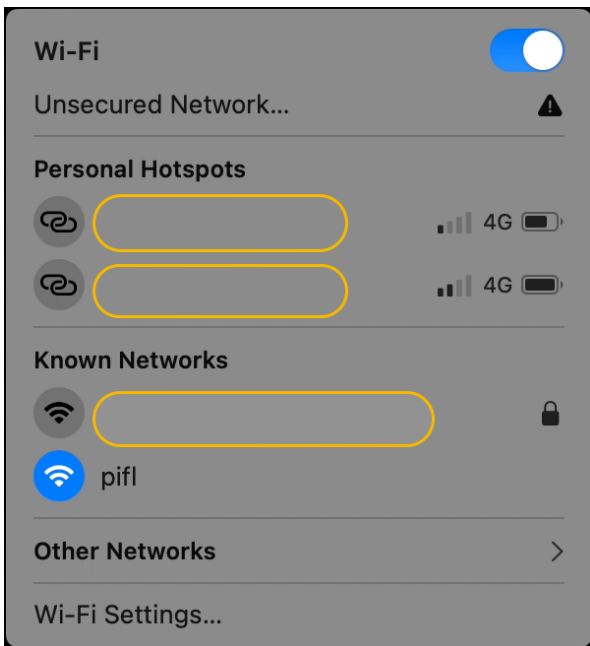


The welcome screen should now display correctly as shown below.



Initial Web Page

To see this page you must connect your remote device to WiFi SSID 'pifl'. This is shown here on the WiFi Known Networks list.



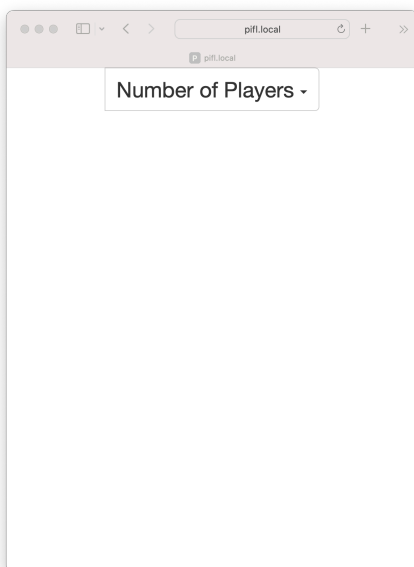
Notice that 'pifl' is an Unsecured Network.

This isn't an issue because, as configured, there is no bridge to the internet.

This also means that no password is needed to connect.

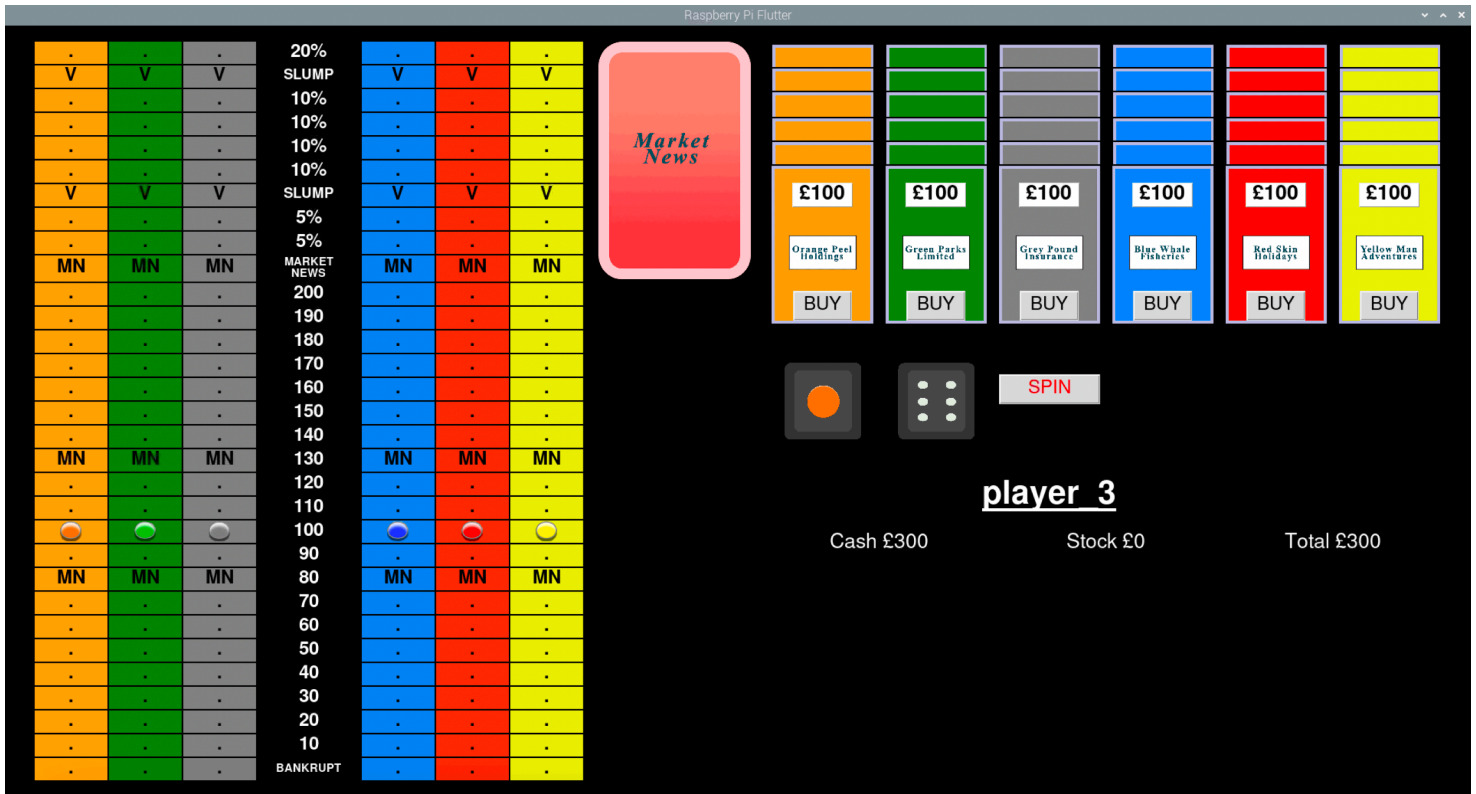
The areas marked in yellow are masking out personal data.

Once connected to the 'pifl' Access Point go to URL 'pifl.local' or '192.168.4.1' on web browser and select Number of Players from the Dropdown list to start the game.

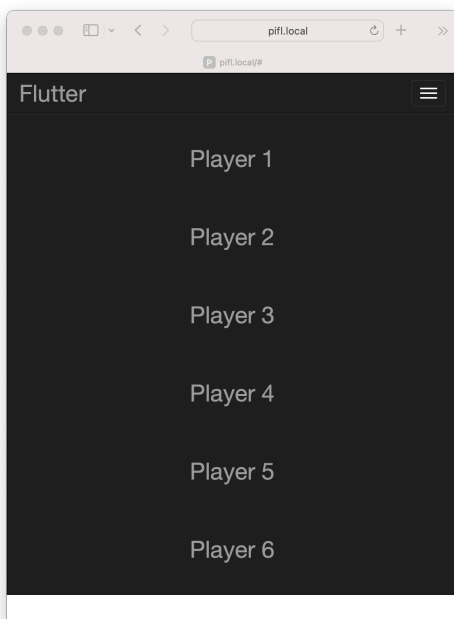


Six Players Selected

This shows the main board on the left, Market News in the centre and stock cards available to the right. Below the stock card display is a colour dice, a number dice and a spin button. Below this is the state of the current player, which in this case is player 3 chosen at random on start-up.



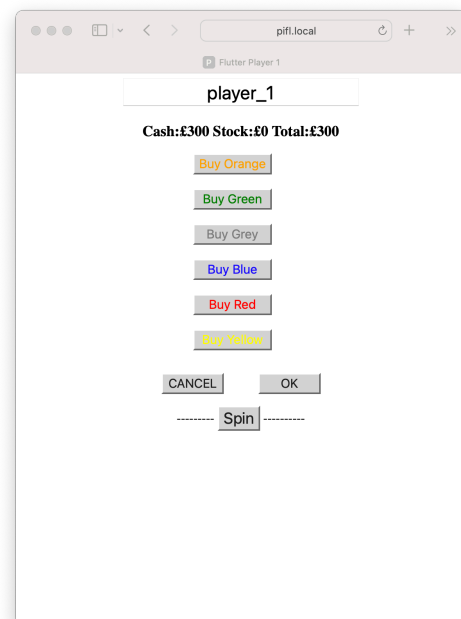
Once the game starts up and the above display appears the web page changes as shown below.



After selecting player 1, the page changes as shown on the right.

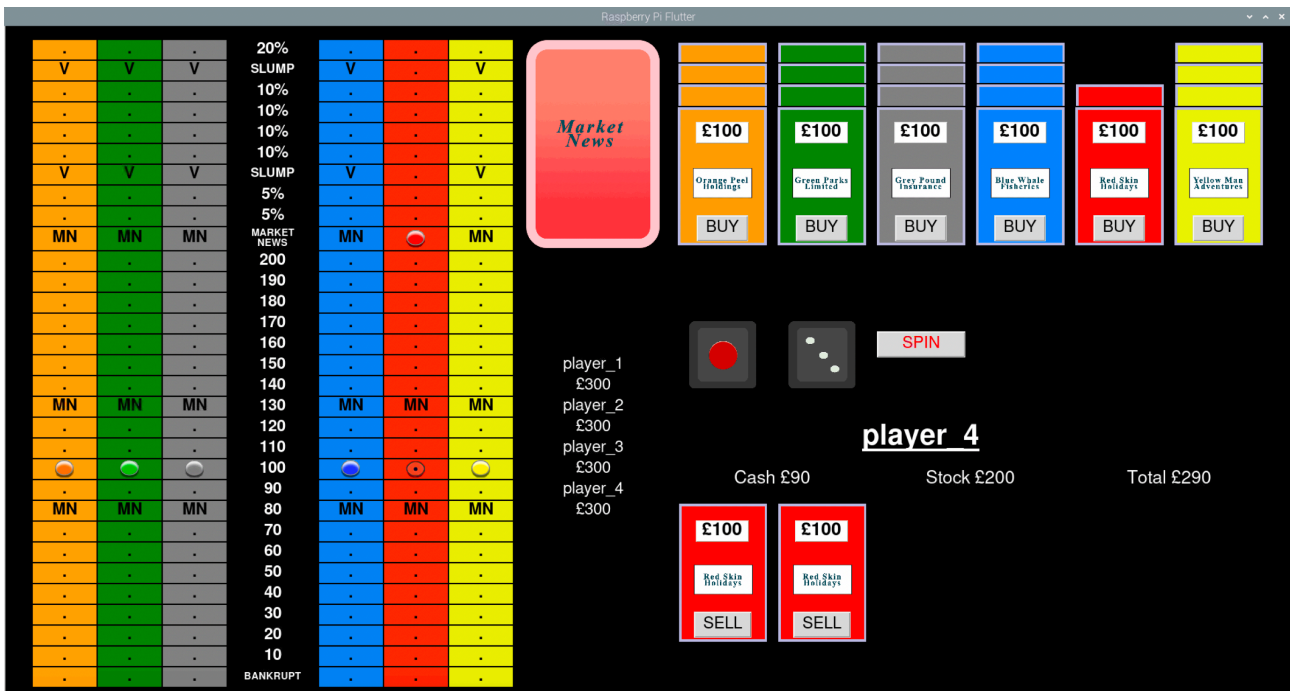
Each player selected will be provided with their own web page.

The name 'Player_1' can be edited to the players own name

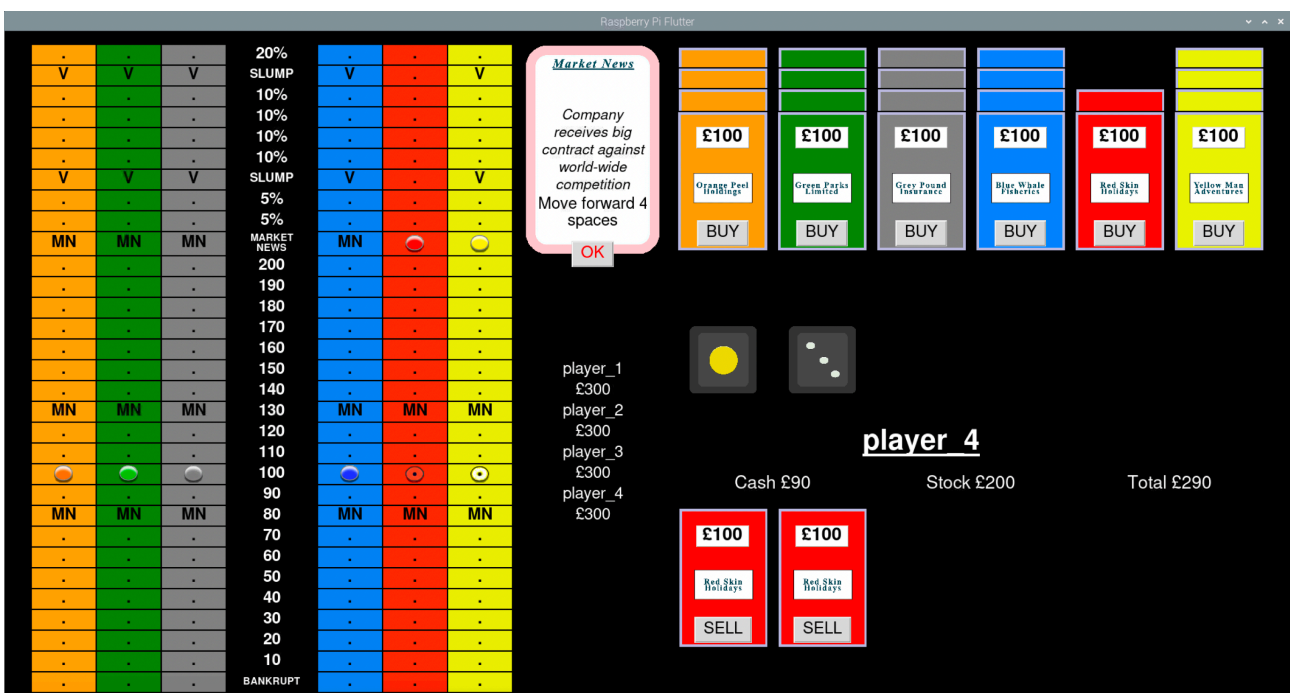


Play Example

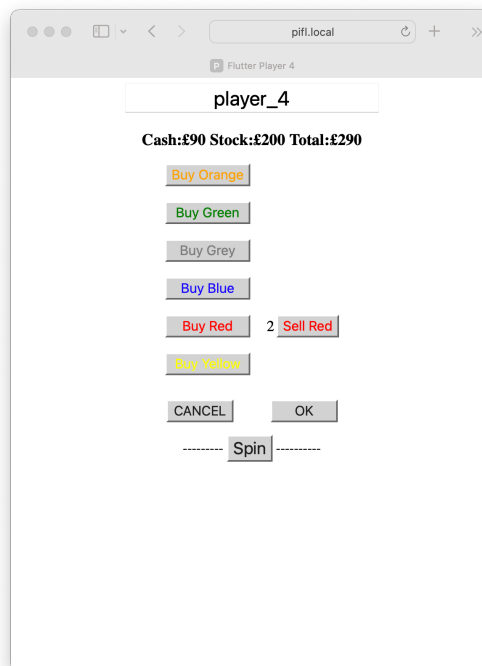
This image shows a 4 player game where the Red stock has landed on MN and automatically advanced to Market News and turned up an 'anti-slump' card (if a peg lands on a 'V' it returns to its parent peg location). The anti-slump card removes the slump markers. Player_4 has taken advantage of the situation and bought two red shares for £100 each plus a £5 buyers fee, and now has £90 cash left from the initial £300.



After Player_4 spins the dice the yellow stock lands on Market News and will move forward 4 spaces when Player_4 clicks 'OK'.

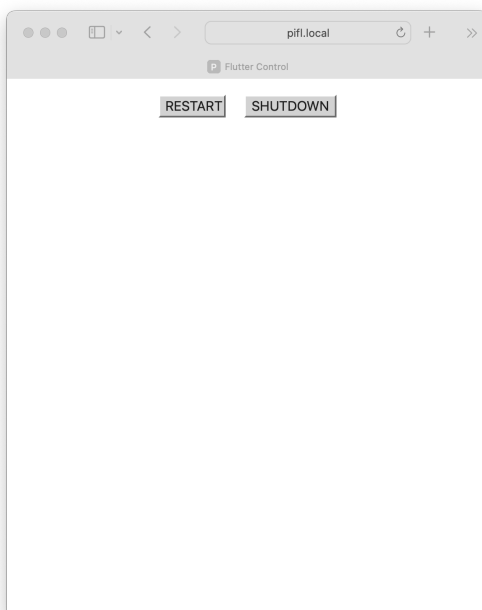


This shows Player_4's Web Page after buying 2 Red shares.



When one of the players reaches £600 in their total value (cash+stock) the game ends with a message showing who has won.

When any player presses OK the game will unload and the Raspberry Pi will enter a shutdown sequence with a delay of 1 minute. It is important to allow the system to shutdown before cycling the power to restart it, as powering off too early can leave the SD card in an unknown state.



With the above issue in mind the software provides a reboot/shutdown web page which becomes available once the game has cycled through all the players. This is so that a clean shutdown can be assured if the game is cancelled, or in the case of requiring a restart.

By entering URL 'pifl.local' in a browser during play this web page will appear.

Pressing RESTART or SHUTDOWN will do as it says. RESTART invokes a reboot of the system so takes a 3 minutes or so.