**Flight Simulation Game - Operating Instructions**

# Overview

The purpose of this document is to provide a brief description of the Flight Simulation Game applications, as well as guidance in starting and operating each application.

For detail regarding host names, login details, IP addresses and the like, refer to section: **Appendix A - Hardware List**

# Initial Setup

All PCs and the Raspberry Pi connect to the office 3G router - or stand-alone router (without external internet access). Please note the sequence of app startup, as this is important.

**Note:** There are two setups detailed in Appendix A - Hardware List, this guide is written around the **condensed setup**, as shown in Table 8 - Condensed Setup.

Table - - Initial Setup Steps

|  |  |
| --- | --- |
|  | controller laptop  Using **piclublaptop03**:   1. Start the controller laptop 2. Verify the laptop is connected to the network 3. Verify the mysql-server service is running. If unsure, use the command below:   > sudo service mysql-server start |
|  | raspberry pi   1. Turn on the Raspberry Pi 2. Verify the Pi is to connected to the network |

# Pre-Launch Configuration Checks

In this section, you will verify the apps are configured correctly and ready to run. The steps below refer to host names, IP addresses and login credentials which are used in the apps’ config files. These details can be found in section Appendix A - Hardware List.

Table - - Configuration Checks

|  |  |
| --- | --- |
|  | the game - config checks  Using the **Raspberry Pi**:   1. Navigate to the game directory   > cd ~/flightsim/picode   1. Open the game’s config file and verify the following settings are correct for the current database setup   > nano digitalenginesimulator.cfg  [databaseconnection]  hostname=<mysql server ip address>  username=<db username>  password=<db password>  dbname=<database to be used>   1. Using the same config file opened above, verify the game is **not in test mode**, by setting the fakepeddling value to False.   [testing]  fakepeddling=False |
|  | registration app - config checks  Using **piclublaptop03**:   1. Navigate to the app directory   > cd ~/flightsim/registration   1. Open the game’s config file and verify the following settings are correct for the current app setup   > nano config.json   |  |  | | --- | --- | | site | http://<local ip address>:<port>[[1]](#footnote-1) | | host | 0.0.0.0[[2]](#footnote-2) | | port | 5000 |  1. Open the game’s database config file and verify the following settings are correct for the current database setup   > nano db\_config.json   |  |  | | --- | --- | | database | flightsim | | host | <ip address of the database server> | | user | <database user id> | | password | <database password> | |
|  | scoreboard app - config checks  Using **piclublaptop03**:   1. Navigate to the app directory   > cd ~/flightsim/scoreboard   1. Open the app’s config file and verify the following settings are correct for the current app setup   > nano config.json   |  |  | | --- | --- | | host | 0.0.0.0[[3]](#footnote-3) | | port | 5001 |  1. Open the app’s database config file and verify the following settings are correct for the current database setup   > nano db\_config.json   |  |  | | --- | --- | | database | flightsim | | host | <ip address of the database server> | | user | <database user id> | | password | <database password> | |
|  | profiler app - config checks  Using **piclublaptop03**:   1. Navigate to the app directory   > cd ~/flightsim/profiler   1. Open the app’s config file and verify the following settings are correct for the current app setup    1. The dir\_graph and dir\_graph\_player keys must contain the relative path to the scoreboard app’s images directory[[4]](#footnote-4)   > nano config.json   |  |  | | --- | --- | | dir\_graph | ../scoreboard/static/images | | dir\_graph\_player | ../scoreboard/static/images/player\_graphs |  1. Open the app’s database config file and verify the following settings are correct for the current database setup   > nano db\_config.json   |  |  | | --- | --- | | database | flightsim | | host | <ip address of the database server> | | user | <database user id> | | password | <database password> | |

# App Startup Instructions

The sections below contains guidance for starting the apps.

Tip: As all apps are run from piclublaptop03, it might be useful to use a **TMUX** session to launch and control the apps.[[5]](#footnote-5) Alternatively, you can open a new console session for each app.

Table - Startup Instructions

|  |  |
| --- | --- |
|  | database startup  If the MySQL service startup was preformed as part of step Table 2-1 - Initial Setup Steps, this step can be skipped.   1. Start the controller laptop 2. Verify the laptop is connected to the network 3. Verify the mysql-server service is running. If unsure, use the command below:   > sudo service mysql-server start |
|  | Registration app - startup  Using **piclublaptop03**:  > cd ~/flightsim/registration  > python app.py   1. Watch for the IP address and port to be output to the console - this will be the address used in Firefox to load the interface[[6]](#footnote-6)   Using **FireFox**:   1. Open a new tab and enter the host’s IP address and port (as observed when starting the application) into the address bar    1. For example: 192.168.123.63:5000 2. Press F11 to change the browser to full-screen mode 3. Zoom in/out so the page content fits on a single page, without scrolling |
|  | scoreboard app - startup  Using **piclublaptop03**:  > cd ~/flightsim/scoreboard  > python app.py   1. Watch for the IP address and port to be output to the console - this will be the address used in Firefox to load the interface[[7]](#footnote-7)   Using **FireFox**:   1. Open a new tab and enter the host’s IP address and port (as observed when starting the application) into the address bar    1. For example: 192.168.123.63:5001 2. Press F11 to change the browser to full-screen mode   Zoom in/out so the page content fits on a single page, without scrolling |
|  | profiler app - startup  Using **piclublaptop03**:  > cd ~/flightsim/profiler  > python profiler.py |
|  | the game - startup  Using the **Raspberry Pi**:  > cd ~/flightsim/picode  > sudo python digitalenginesimulator.py digitalenginesimulator.cfg  **Alternatively**, if the Pi is running in GUI mode, the game can be launched by:   1. Double**-**click the Launch Pi.sh icon on the desktop 2. Click “Execute” when prompted |

# Operating Instructions

The sections below contain guidance for operating the apps.

Table - Operating Instructions - Registration

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | registration app - use Application Overview The registration app is used to record each player’s alias, details and avatar image in the database. The scoreboard app then uses this data to manage the player queue and high scores. Application Operation A new player will do the following, using the registration app’s web UI:   1. Select their player alias (a combination of a colour, animal and a number) 2. Enter their guardian’s details 3. Select an avatar image 4. Click **Register!**   Clicking the **Register!** button will store the new player’s information to the database, along with a status of ‘QUEUED’ and a gamehighscore of 0. Field Requirement Details  |  |  | | --- | --- | | Colour | required | | Animal | required | | Number | required | | Guardian’s email address | optional | | Guardian’s phone number | optional | | Avatar | required | |

Table - Operating Instructions - Scoreboard

|  |  |
| --- | --- |
|  | scoreboard - use Application Overview This dashboard is set to refresh every (*2*) seconds[[8]](#footnote-8); which will automatically pick up new players added to the queue, new high scores and the profile of the player who has just completed the game. Queue Management: Options Three options have been built into queue for easy management.   1. **Set a player’s status** to **PLAYING** 2. **Move** a player to the back of the queue 3. **Remove** a player from the queue  Queue Management: Use The following items provide further detail on the queue management options listed above.   1. Set a player’s status to **PLAYING**    * **In THE QUEUE section, click the QUEUED label**      + This action will update the player’s status to **PLAYING**, make the player appear in the NOW PLAYING section, and send the player to the Pi to begin the game 2. To move a player to the back of the queue:    * **In THE QUEUE section, click the player name**      + This option will update the queueposition value in the database with the current epoch time; thus moving the player to the back of the queue 3. Remove a player from the queue    * **In THE QUEUE section, click the player’s avatar and select Yes to the prompt[[9]](#footnote-9)**      + This action will change a player’s status to DELETED; while **keeping** the player’s record in the database |

Table - Operating Instructions - The Game

|  |  |
| --- | --- |
|  | the game - use Application Overview The Flight Simulation Game application is the game itself. As the player cycles, this is the program which tracks and displays the player’s profile to the screen in front of the player. Player management is controlled by the Queue Manager / Scoreboard application. Application Operation Once invoked, the application will wait until a player is selected for play.  A player is selected for play using the Queue Manager / Scoreboard application. (For further instruction refer to section Table 4 - Operating Instructions - Scoreboard.)  To start the game for a selected player, the player must press the “START” button on the handlebars of the bike.  The game will run for 90 seconds. When complete, the player’s score is displayed, and the program resets and waits for the next player.  Function keys are used to control the behaviour of the application.   * ESC = Exit the application and close |

# Appendix A - Hardware List[[10]](#footnote-10)

Table - Full Setup

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Function | Machine Type | Hostname | IP Address | Username | Password |
| MySQL Database Server | Dell Latitude E5530 | n/a | 192.168.123.40 | root | Password!1 |
| MySQL Database Server |  |  |  | admin | Adm1n.01 |
| Flight Simulation Game | Raspberry Pi 3 | raspberrypi | 192.168.123.50 | piclub | Clubber5! |
| Queue Manager / Scoreboard | HP Probook 6560b core i5 | piclublaptop01 | 192.168.123.61 | piclub | Clubber5! |
| User Registration | HP Probook 6560b core i3 | piclublaptop02 | 192.168.123.62 | piclub | Clubber5! |
| Controller / Monitor | Dell Latitude E5530 | piclublaptop03 | 192.168.123.63 | piclub | Clubber5! |

Table - Condensed Setup

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Function | Machine Type | Hostname | IP Address | Username | Password |
| Flight Simulation Game | Raspberry Pi 3 | raspberrypi | 192.168.123.50 | piclub | Clubber5! |
| Controller | Dell Latitude E5530 | piclublaptop03 | 192.168.123.63 | piclub | Clubber5! |
| User Registration |  |  |  |  |  |
| Queue Manager / Scoreboard |  |  |  |  |  |
| MySQL Database Server |  |  |  | root | Password!1 |
| MySQL Database Server |  |  |  | admin | Adm1n.01 |

1. This value **must match** the port key. [↑](#footnote-ref-1)
2. Use 0.0.0.0 if running the registration web UI from a remote device. Use the the local IP address if running the web UI locally. [↑](#footnote-ref-2)
3. Use 0.0.0.0 if running the scoreboard web UI from a remote device. Use the the local IP address if running the web UI locally. [↑](#footnote-ref-3)
4. If you don’t know where the images directory lives in the scoreboard’s structure, open the scoreboard’s directory tree and do a recursive search for ‘images’. [↑](#footnote-ref-4)
5. TMUX *should* already be installed on all laptops. If you have questions on how / why to use TMUX, see authors. [↑](#footnote-ref-5)
6. If 0.0.0.0 is displayed as the IP, then use the PC’s IP address in the browser. If 127.0.0.1 is displayed, use 127.0.0.1 in the browser. As an aside, 0.0.0.0 indicates the app is configured for ‘public’ access, i.e.: can be accessed from other (e.g. portable) devices on the network. [↑](#footnote-ref-6)
7. If 0.0.0.0 is displayed as the IP, then use the PC’s IP address in the browser. If 127.0.0.1 is displayed, use 127.0.0.1 in the browser. As an aside, 0.0.0.0 indicates the app is configured for ‘public’ access, i.e.: can be accessed from other (e.g. portable) devices on the network. [↑](#footnote-ref-7)
8. The refresh rate on initial deployment is 2 seconds, and can be updated using the

   <meta http-equiv=’refresh’ content=’2’/> line in ./templates/scoreboard.html. [↑](#footnote-ref-8)
9. Note: The prompt will disappear quickly, due to the screen refresh. It might take a couple attempts to catch the timing right. [↑](#footnote-ref-9)
10. All blank fields are ‘same as above’. [↑](#footnote-ref-10)