



PIPSTA007 – PIPSTA AND SCRATCH

Contents

Difficulty Level:.....	2
Time to Complete:	2
Who Should Read This Document	2
Introduction	3
Pre-Requisites	5
Getting Started.....	5
How it Works.....	6
Adjusting Functionality	7
Next Steps	7
Shutting Pipsta Down Safely	8

Revision History

Revision	Author	Date	Description
1.0	AH/GJ	27/11/14	First Release

Difficulty Level:



- This tutorial culminates in a fully-functional Scratch application and Python script for printing certificates.
- The detail of the Python Listener script is covered in a later tutorial.

Time to Complete:



- Thoroughly exploring the Scratch program could take 30-45 minutes

Who Should Read This Document

- ICT providers for schools,
- Those seeking solutions to meet the changes to the national curriculum as of September 2014,
- Those interested in providing printing support for *Scratch*,
- Those wanting to understand the functionality of the provided *Scratch Projects* and *Python Scripts*,
- Those wanting to expand on the functionality of the provided *Scratch Projects* and *Python Scripts*.



Introduction

Scratch is a graphical programming language, and is many children's first exposure to programming. Increasingly popular in schools and at home, this programming language works on a variety of platforms, including the Raspberry Pi.

See "*Pipsta and the National Curriculum*" to see how Pipsta and the Raspberry Pi can be used throughout the Key Stages.

Pipsta can be used in conjunction with Scratch to enhance pupils' and students' experience by providing them with something tangible as a memento from their lessons. Although the cost of such printed awards and rewards is low, it is the content of the print that confers great value: text, graphics and barcodes can be combined to provide, e.g.:

- Badges,
- Merits,
- QR-Codes (2D barcode) links,
- Quiz results sheets permitting review, comparison and corrections,
- Small certificates of achievement,
- Leader-boards, etc.



TIP:

Pipsta can be used with standard paper rolls or special linerless label rolls: an adhesive-backed continuous label medium which can be stuck and removed many times.

With linerless labels, temporary decorations and banners are easily produced; badges can be created that leave no residue on clothes and notes can be affixed to exercise books and other documents to be subsequently removed without damage. Merits can even be stuck to pieces of work and subsequently transferred to wall-charts.

At the time of writing, **Scratch 1.4** is provided as a part of the Raspberry Pi's Raspbian Wheezy operating system distribution and appears on the default desktop. Pipsta support is provided by means of '**Broadcasts**': messages sent from Scratch to be picked-up by a Python script '**Listener**' for processing. This means that Scratch blocks can be defined to instigate a variety of print activities.



TIP:

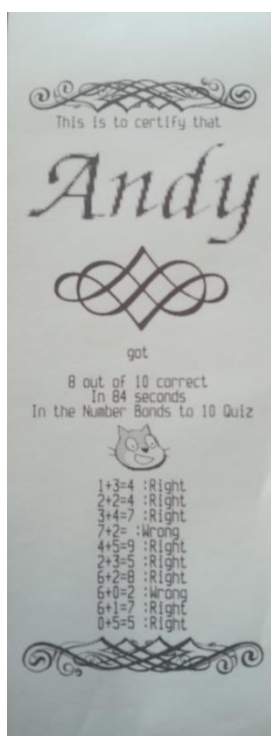
Whilst it is possible to write Scratch programs on other platforms (e.g. Windows PC) and transfer the program across to the Raspberry Pi thereafter, **you must ensure that you are using Scratch 1.4** or compatibility issues may arise. Whilst Scratch 2.0 is currently available, this is not available for the Raspberry Pi at this time.

In this simple example, we will demonstrate a simple 'Number Bonds' Scratch program which produces a short, personalised certificate. The key-features of the Scratch game are:



- 1) The game itself works 'out-of-the-box' at an Early Years Foundation Stage and/or Key Stage 1 level, so teachers and parents need have no programming experience,
- 2) The game attempts to engage children with the simple animation of a 'rocket' on a lunar surface, with the rocket moving further up the screen with each correct question,
- 3) The game can be easily configured by modification of a few key parameters in a block at the top of the game, including:
 - a. The Number Bonds maximum value (default is 10)
 - b. The minimum value of each term (specifically addressing whether zero is to be permitted or not)
 - c. The number of quiz questions per game (default is 10)
 - d. The award threshold at which pupils will be awarded 'play time' at the end. During 'play time' the rocket will start flashing multi-colours and will trail the mouse pointer position, effectively allowing the child to move and play with the rocket. By default, the threshold at which 'play time' is awarded is 8 questions.
 - e. The amount of 'play time' awarded (default of 30 seconds)
- 4) The Scratch code itself defines the elements that are printed in the certificate at the end of the game, once again meaning that the game can be tailored by teachers and parents with little or no programming experience.

By default, the certificate will produce an image similar to the following:



The virtues of this certificate are:

- Graphical interest: the print-out has a higher perceived value than a simple text print,
- Personalisation: the pupil's name is printed in a calligraphic font (scaled to the full width of the paper),
- Pupil's score is printed,
- The sums generated and answers provided by the child are listed, permitting review by pupil and teacher as per the National Curriculum.

There will be an ongoing programme of producing and maintaining a library of Scratch programs that cater to the various Key Stages of the National Curriculum; please check www.pipsta.co.uk regularly for updates.



Pre-Requisites

This guide is intended for users who have completed the mechanical build of their Pipsta and have successfully performed the first-time setup concluding in a simple print-out, as per:

- **PIPSTA002 - Pipsta B+ Assembly Instructions** or
- **PIPSTA003 - Pipsta A+ Assembly Instructions**
- and
- **PIPSTA004 – Pipsta First-Time Setup**

It is expected that you have the following items:

- Your assembled, working Pipsta
- Power supply for Raspberry Pi (5v, ideally no less than 2.0A rated)
- USB A to Micro B cable assembly (if not integrated into Raspberry Pi power supply)
- Micro SD Card configured as per the *Pipsta First-Time Setup tutorial*
- Consider an additional, reserve Micro SD Card!
- Access to two mains sockets for powering the Raspberry Pi and Pipsta printer
- USB Keyboard
- USB Mouse
- A Wi-Fi connection as per the *Pipsta First-Time Setup tutorial*
- Video/Monitor lead:
 - HDMI lead *or*
 - 3.5mm 4 pole jack plug to RCA composite cable *or*
 - HDMI to VGA adaptor and VGA cable
- Computer monitor or television (with HDMI, component video or VGA input as above)

Additional Configuration Step

- Open LXTerminal
- At the \$ prompt, enter:
sudo pip install scratchpy

to install the module that permits communications between Scratch and Python. This is a one-time operation, and will not need to be repeated once installed.

Getting Started

- 1) Power up your Raspberry Pi and printer
- 2) Wait for the Raspberry Pi to boot into the graphical desktop environment
- 3) Click on *File Manager* on the task-bar.
- 4) Navigate to **/home/pi/pipsta/Scratch/Game_with_Certificate**
- 5) Now right-click on **'number_bonds.sb'** and select 'Scratch' from the pop-up menu. This will launch the Scratch environment and load the Number Bonds game.
- 6) From File Manager, press **[F4]** to open LXTerminal and enter the following at the \$ prompt:
python certificate.py
This will start the Python Listener script ready for the end of the game when the certificate will be printed.



- 7) Back in Scratch, check that the Pipsta printer has a tick on it (indicating that the connection between Scratch and Python is live)
- 8) Now, in Scratch, press the green flag symbol:



to start the game.

- 9) Follow the instructions in the speech bubble from the 'rocket' to play the game
- 10) Once the game is completed, and whilst the certificate is being printed, you will be able to move the mouse around to have thirty seconds of free 'play time' with the now flashing rocket.

How it Works

The primitive game itself can be conceptually split into several sections:

- 1) Initialisation: all the variable are set to their default values
- 2) The player is invited to enter their name
- 3) Scratch broadcasts a 'generate name' command
- 4) Scratch broadcasts the player's name. The Python 'Listener' *certificate.py* begins processing the name data into a font image. If no font filename was provided, the default Raspbian font 'Chancery' will be used. By initiating this image generation *before* the game begins, this allows Python to get to work processing the image in the background while the game is being played, ensuring there are no additional delays at the end of the game.
- 5) Game:
 - a. The game loops a pre-defined number of times
 - b. random numbers (in the pre-defined range) are used to generate questions
 - c. the player is invited to input the answer to each question
 - d. the player is provided with feedback on each question
 - e. the 'rocket' moves upwards as a simple animation if the player's answer was correct
 - f. at the end of the game, the player is told how many questions they got correct
- 6) Certificate:
 - a. Scratch broadcasts a 'top flourish' command. The Python Listener uses the *imaging library* to load a calligraphic flourish, and uses *bitarray*, *PyUSB* and *LibUSB libraries* to manipulate and send the data to the Pipsta printer.
 - b. Scratch broadcasts a 'centre justify' command. The Listener prepares to centre justify subsequent text
 - c. Scratch broadcasts a 'newline' command to feed paper
 - d. Scratch broadcasts the raw text 'This is to certify that'
 - e. Scratch broadcasts two 'newline' commands
 - f. Scratch broadcasts a 'display name' command, drawing-upon the image generated during the game.
 - g. Scratch broadcasts a 'mid flourish' command
 - h. Scratch broadcasts two 'newline' commands
 - i. Scratch broadcasts the text 'got'
 - j. Scratch broadcasts two 'newline' commands



- k. Scratch joins a message together to detail the number of questions correct and the total number of questions, and broadcasts this to the Listener.
 - l. Scratch broadcasts a 'newline' command
 - m. Scratch joins a message together detailing the amount of time taken in seconds and broadcasts this
 - n. Scratch joins and broadcasts a message detailing the number bonds to <maximum>
 - o. If the number of questions successfully answered is greater-than or equal-to the award threshold set in initialisation, a command to print the Scratch Cat image is broadcast, otherwise, no such broadcast is made and no cat image will be produced.
 - p. Scratch broadcasts a 'newline' command
 - q. There now follows a block of code which retrieves and broadcasts each of the questions and answers provided, along with text declaring the answer to be right or wrong.
 - r. Scratch broadcasts a 'bottom flourish' command, prompting the Listener to load the flourish image and *rotate it* prior to printing it just as before.
 - s. Finally, Scratch will broadcast 5 newline commands to feed the certificate past the tear-bar.
- 7) Play time: If the player exceeded the Award Threshold, whilst the Python Listener is producing the certificate, the player can move the mouse pointer to get a now flashing 'rocket' to fly around the screen.
- 8) End of game: finally, the rocket will stop flashing and return to its starting position in readiness for the next game. The 'game over' text will be displayed on-screen.

Adjusting Functionality

- 1) Try an alternative font for generating the player's name by issuing a command in the following format:
python certificate.py "<your-font-path-and-name-here>"
- 2) Append a merit to the bottom of the certificate for players answering all questions correctly by issuing more broadcasts from Scratch.

Next Steps

If this is the end of your session, see the section entitled *Shutting Pipsta Down Safely*, or –to continue on to learn how the Listener Script works with the following tutorial:

➤ ***PIPSTA107 –Scratch Python Listener Code Tutorial***



Shutting Pipsta Down Safely

Whilst the printer is resilient when it comes to powering down, the Raspberry Pi must undergo a strict shutdown process to avoid corrupting the Micro SD card. The most straightforward method of doing this is to double-click the 'Shutdown' icon on the desktop.



TIP:

If you are already in LXTerminal, type **sudo shutdown -h now** to shut-down the Raspberry Pi immediately.



TIP:

Always make sure ALL activity on the Raspberry Pi's green LED (the LED on the right) has stopped before removing the power!

■End of Document■