This book teaches you how to hook up your Raspberry Pi computer, install different games from a variety of genres, and set up emulators so you can play hundreds of classic arcade and console games. Moreover, you will also learn how to design, create, and play video games that you create from scratch.

After learning how to set up a Raspberry Pi, you will begin by creating your own version of Flappy Bird and a clone of the classic game Pong in the Scratch programming language. You will also be guided through the installation process for a wide range of gaming operating systems, such as PiPlay, RetroPie, and ChameleonPi for the Raspberry Pi. Furthermore, you will discover in-depth details about emulators that recreate classic 80s arcade games to consoles that many grew up with.

You will also learn more about installing games through Linux repositories, setting up controllers, programming pieces of your Minecraft world, and troubleshooting various issues that can crop up with your Raspberry Pi.

Who this book is written for
If you are someone who loves to play games and are interested in learning more about the capabilities of your Raspberry Pi, this book is for you. Basic knowledge of Raspberry Pi programming is expected.

What you will learn from this book
- Program games utilizing the Scratch language
- Install multiple operating systems
- Set up your Raspberry Pi computer
- Install and configure game system emulators
- Control your Minecraft world with the Python programming language
- Explore different kinds of joysticks, controllers, game pads, and other input devices
- Install applications in Linux


Raspberry Pi Gaming
Second Edition

Design, create, and play all kinds of video games on your Raspberry Pi computer

Shea Silverman

In this package, you will find:

- The author biography
- A preview chapter from the book, Chapter 1 'Getting Started with the Raspberry Pi'
- A synopsis of the book’s content
- More information on Raspberry Pi Gaming Second Edition

About the Author

Shea Silverman has been using computers since he was two years old. He has always been drawn to technology, video games, education, and the public sector. He is an employee at the Center for Distributed Learning at UCF, where he spends his time researching and developing new ways to enhance online learning. He is a member of the Orlando makerspace FamiLAB and an alumni of the University of Central Florida. His article entitled Hacking, Learning, and the Raspberry Pi was published in 2600: The Hacker Quarterly, he was a technical reviewer for Raspberry Pi Networking Cookbook, Packt Publishing, and is the author of Raspberry Pi Gaming, Packt Publishing.

You can find more information about him at http://www.sheasilverman.com

I would like to thank my wonderful wife, Kristene, who provides unending encouragement and support to my projects. I would like to thank my friends and family for their ongoing support, especially my grandma for always believing in me. Finally, I would like to thank Liz, Eben, and the Raspberry Pi Foundation for the creation of the Raspberry Pi, as well as the wonderful community that has flourished since its release.
Hi! Welcome to the wonderful world of the Raspberry Pi. In a few short years, the Raspberry Pi has amassed a rich diversity of software, cultivated by its incredible community.

In this book, we are going to explore the entertainment capabilities of the Raspberry Pi. From programming your own video games, to reliving classic moments with your favorite game systems, I'm positive Raspberry Pi gaming will help you unlock the capabilities of your device.

What This Book Covers

*Chapter 1, Getting Started with the Raspberry Pi*, will explain the various differences between the Raspberry Pi models, show you how to set up an SD card for use in your device, and finally how to hook up your Raspberry Pi.

*Chapter 2, Scratch*, will introduce the programming language and programming concepts required to build a game. By the end of this chapter, you will have made two games reminiscent of Flappy Bird and Ping Pong.

*Chapter 3, Raspberry Pi Gaming Operating Systems*, guides you through the different operating systems that are dedicated to video gaming. These distributions have been specially set up to offer a fun out-of-the-box experience.

*Chapter 4, Emulators*, explains the various gaming consoles that are available to be emulated on the Raspberry Pi. You will also learn how to access the Raspberry Pi App Store and use the built-in software repository.

*Chapter 5, Ported Games*, shows you how to install and use games that were originally for other systems but have now been reprogrammed to run on the Raspberry Pi.

*Chapter 6, Linux Games*, explores those games that are native to the Linux operating system.

*Chapter 7, Controllers*, will introduce and explain the world of controllers, gamepads, and arcade sticks, and how they can interface with the Raspberry Pi. You will also learn how to hook up your favorite console game pads to your Pi.

*Chapter 8, Troubleshooting*, will guide you through the common issues that crop up when utilizing a Raspberry Pi.

*Appendix, Games List*, provides you with a list of the native Linux games that are available in the Raspbian repositories.
The Raspberry Pi is an inexpensive, feature-rich modern computer created by the Raspberry Pi Foundation. Since the release of the Model B in 2012, the community surrounding the computer has grown, allowing for an incredible amount of projects and software to be created for the device. These range from programming languages, educational applications, hardware prototypes, and of course, video games.

In this chapter, you will learn the following topics:

• The different flavors of a Raspberry Pi
• Setting up an SD card
• Hooking up your Raspberry Pi

The different flavors of a Raspberry Pi
The Raspberry Pi Foundation has released four major models of the Raspberry Pi computer. They are Model A, Model B, Model B+, and Model A+. The Raspberry Pi’s CPU is the Broadcom BCM2835 chip. It contains an ARM processor running at 700 MHz and a powerful graphics chip. The board features HDMI and Composite (RCA) video outputs, USB ports, two expansion slots, a Micro USB port for power, and an array of GPIO (General-purpose input/output) pins to interact with the outside world.
Since all models share the same basic hardware platform, all the examples in this book are applicable to all the versions.

<table>
<thead>
<tr>
<th></th>
<th>Model A</th>
<th>Model A+</th>
<th>Model B</th>
<th>Model B+</th>
</tr>
</thead>
<tbody>
<tr>
<td>USB</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>Ethernet</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Video outputs</td>
<td>HDMI/ Composite</td>
<td>HDMI/ Composite via 3.5 mm jack</td>
<td>HDMI/ Composite</td>
<td>HDMI/ Composite via 3.5 mm jack</td>
</tr>
<tr>
<td>Memory</td>
<td>256 MB</td>
<td>256 MB</td>
<td>512 MB</td>
<td>512 MB</td>
</tr>
<tr>
<td>Storage</td>
<td>SD card</td>
<td>MicroSD card</td>
<td>SD card</td>
<td>MicroSD card</td>
</tr>
<tr>
<td>Power usage</td>
<td>300 mA</td>
<td>300 mA</td>
<td>700 mA</td>
<td>600 mA</td>
</tr>
<tr>
<td>Price</td>
<td>$25</td>
<td>$20</td>
<td>$35</td>
<td>$35</td>
</tr>
<tr>
<td>Differences</td>
<td>Low cost solution. Does not have built in Ethernet and it has only one 1 USB port.</td>
<td>Newest board. Low power, low cost, and much smaller form factor.</td>
<td>Original board. Balances features and price.</td>
<td>New revision to the B board. Has a new layout, 4 USB ports, and more GPIO pins.</td>
</tr>
</tbody>
</table>

## Setting up an SD card

The Raspberry Pi uses SD cards to contain its operating system and main storage space. A Raspberry Pi SD card contains two partitions, which are explained as follows:

- The first one, is the boot partition. This space contains the Linux kernel, required boot up files, and most importantly, the `config.txt` file. This file allows you to change the boot time parameters and customize some of the functions of the Raspberry Pi. These options include over-clocking the device, changing monitor settings, and the memory split between CPU and GPU, among numerous other options.

- The second partition contains a Linux partition, which holds all of your applications, configurations, and operating system files.

Preloaded SD card images are available, which make it quick and easy to get your Raspberry Pi up and running.
Choosing an SD card is an important step. There are many different combinations of card sizes and card speeds. 4 GB is the minimum size required for many of the operating systems (OS). I recommend that you start out with an 8 GB card. You will also see cards marked with Class 4, Class 6, and Class 10. This is the speed at which the card can be read and written to. I have found that the best bang for the buck is a Class 6 card, but don't worry too much about which one you choose.

Before we begin, you will need to download a suitable Raspberry Pi OS. We will be using the official operating system called Raspbian. You can download it from http://www.raspberrypi.org/downloads.

Warning! dd and Win32DiskImager can be used to overwrite your computer's own hard drive or other drives connected to your computer. Double and triple check that the drive you select is your SD card.

Creating the SD card in Windows
To create the SD card in Windows, you will need to download the program called Win32DiskImager by visiting http://sourceforge.net/projects/win32diskimager/.

Once you are done with the downloading, perform the following steps:

1. Unzip the Raspbian image by double-clicking on the Raspbian.zip file.
2. Select a place on your hard drive to save the extracted file.
3. Click on Extract files....
4. Insert the SD card into your computer's SD card reader.
5. Run Win32 Disk Imager.
Getting Started with the Raspberry Pi

6. Select the Raspberry Pi image on your hard drive.
7. Select the drive letter under the device that corresponds to the SD card.
8. Click on **Write**.

**Creating the SD card in Macintosh OS X**

OS X includes everything you need to create the SD card out of the box. We will use a utility called dd:

1. Double-click the Raspbian image ZIP file. It will automatically extract into the same place as the ZIP file.
2. Insert the SD card into your computer's SD card reader.
3. Open the terminal application (located in the **Applications** | **Utilities** folder).
4. Find the name of your SD card by typing `diskutil list`.

<table>
<thead>
<tr>
<th>#</th>
<th>TYPE NAME</th>
<th>SIZE</th>
<th>IDENTIFIER</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>GUID_partition_scheme</td>
<td>121.3 GB</td>
<td>disk0</td>
</tr>
<tr>
<td>1</td>
<td>EET</td>
<td>708.7 MB</td>
<td>disk0s1</td>
</tr>
<tr>
<td>2</td>
<td>Apple_HFS Macintosh HD</td>
<td>120.5 GB</td>
<td>disk0s2</td>
</tr>
<tr>
<td>3</td>
<td>Apple_Boot Recovery HD</td>
<td>658.0 MB</td>
<td>disk0s3</td>
</tr>
</tbody>
</table>

5. Unmount your SD card by typing `diskutil umountdisk <disk>`, that is, `/dev/disk2`.

   ```bash
   C02FY3EJDF91:~ sheas$ diskutil umountdisk /dev/disk2
   Unmount of all volumes on disk2 was successful
   C02FY3EJDF91:~ sheas$
   ```

6. Copy the OS image from your hard drive to the SD card by typing `dd if=/path/to/os/image.img of=<disk>` where `<disk>` is `/dev/disk2`, and path/to/os/image.img is the place to which you saved the image.
7. It can take anywhere from 15 minutes to over an hour for the image to be written to the SD card. It will look like nothing is happening until it finishes copying. When it is done, you will see a message showing how long it took to transfer in seconds.

```
C02FV3EJDF91:~ shea$ dd if=/Users/shea/raspbian.img of=/dev/disk2
3788800+0 records in
3788800+0 records out
193985500 bytes transferred in 1837.087116 secs (1055496 bytes/sec)
C02FV3EJDF91:~ shea$
```

You can press Ctrl + T at any time to see the current status.

Creating the SD card in Linux

Like OS X, Linux includes everything you need out of the box:

1. Insert the SD card into your computer's SD card reader.
2. Using the terminal of your system, find the name of your SD card by typing `sudo fdisk -l`.
3. If required, you can unmount your SD card by typing `umount <disk>` (which will be listed from the earlier command, i.e. `/dev/disk2`).
4. Copy the OS image on your hard drive to the SD card by typing `dd if=/path/to/os/image.img of=<disk>`.

Now that you have created your Raspberry Pi SD card, it's time to set up our device!

Using NOOBS

The Raspberry Pi Foundation has created a piece of software called New Out Of the Box Software, often abbreviated as NOOBS. This is a small operating system, which is used to install other OSs onto your Raspberry Pi.

Installing NOOBS is much easier than other installs:

2. Insert the SD card into your computer's SD card reader.
3. Drag and drop the NOOBS files onto the SD card and you are done!
At the first boot, you will be presented with a list of OSs available to be installed. Choose Raspbian and press Enter.

**Hooking up your Raspberry Pi**

It’s very easy to properly set up a Raspberry Pi. If you can build a Lego set, you can build your Raspberry Pi computer. The following steps will help you in this:

1. Place your SD card into the SD card slot on the underside of the Raspberry Pi.
2. Connect the HDMI or RCA cable to the respective connector on the Raspberry Pi, and plug the other end into your monitor.
3. Plug the Ethernet cable into the Ethernet jack (not applicable to Model A and A+) on the Raspberry Pi and the other end into your router or switch.
4. Connect the USB mouse and keyboard to the two USB ports available on the Raspberry Pi. If you are using Model A, connect a USB hub to your Raspberry Pi and the mouse and keyboard to the hub.
5. Plug the power supply’s Micro USB connector into the Micro USB port on the Raspberry Pi to turn it on.
6. A red LED, by the USB ports, will light up to indicate that the power is turned on.
7. On your screen, a square rainbow image will appear for a brief moment, followed by some quick moving text or a graphic loading screen.

Congratulations! You have successfully booted up your Raspberry Pi!

**Connecting to a Wi-Fi access point**

The Raspberry Pi and Raspbian can easily connect to the Internet via Ethernet, but when using a USB wireless device, you can also connect to a Wi-Fi network. Raspbian includes a graphical utility to make connecting to a Wi-Fi access point easy. Go through the following steps:

1. Boot up your Raspberry Pi.
2. At the login screen, enter your username and password (default is *pi* and *raspberry*, respectively).
3. In the command prompt, type `startx` to launch the desktop environment.
4. Using your mouse, double-click on the WiFi Config icon that is on the desktop.
5. The Wi-Fi configuration tool will appear shortly.

6. Click on **Scan** to scan for available wireless networks around you.

7. Double-click on the one you wish to connect to.

8. If it is a secured network, you will be asked for a password.
9. Enter the password and click on **Add**.

10. The **Status: Disconnected** message should now change to **Status: Completed**.
11. You are now connected to your wireless network.

**wpa_gui** saves your connection information. If you logout or reboot your Raspberry Pi, it will automatically try to connect to a previously added access point.

If you do not see **wlan0** appear in the **wpa_gui**, or if your Raspberry Pi is unstable, you might need to use a powered USB hub. You should connect your device to a powered hub, and then connect the hub to the Raspberry Pi. This will ensure that the correct amount of power is received.

You can also check whether your Wi-Fi adapter is supported by visiting [http://elinux.org/RPi_USB_Wi-Fi_Adapters](http://elinux.org/RPi_USB_Wi-Fi_Adapters).

**Summary**

In this chapter, you learned how to create your Raspberry Pi’s SD card, hook it up to your TV and other accessories, and connect it to the Internet. Now, it’s time to have fun with the software on your Raspberry Pi.

In the next chapter, you will learn how to create your own video game using the Scratch programming language.
Where to buy this book