

Raspberry Pi

Computer Graphics

Astronaut Reaction Time Game

By Jocelyn Kahendaguza and Sakiko Ominato

**Abstract**

In this document you will install a game called Astronaut Reaction Time Game to Raspberry Pi. Just follow the instruction and you will have the game! We did this project in the Computer Graphics class. Enjoy exploring the technology.

 ****

**Content List**

1. Project description

What is Raspberry Pi ?

Which program do you install in this project?

What will you learn?

1. What you need

Install an Operating System

1. Start your Raspberry Pi

Install an Operating System

1. Install guide

Set the Stage with a Space Theme

Create Variables to Store Data

Begin the Reaction Time Script

1. Bibliography
2. **Project Description**

# What is Raspberry Pi?

A credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. It is a capable little device that enables people of all ages to explore computing, and to learn how to program in languages like Scratch and Python.

(Resource URL: <https://www.raspberrypi.org/help/what-is-a-raspberry-pi/>)

# Which program do you install in this project?

You install a game called Astronaut Reaction Time Game. You can find this project by accessing Raspberry Pi webpage-> Education on the upper bar-> resources-> Astronaut Reaction Time Game (URL: <https://www.raspberrypi.org/learning/astronaut-reaction-times/>). In this game you will create a game using Scratch.

# What will you learn?

By creating a Scratch reaction game you will learn:

* Why astronauts need to have super-sharp reactions in space, and the average distance travelled by the ISS.
* How to import images into Scratch and use them as backgrounds and sprites.
* How to store data like time and distance in variables.
* Use multiplying operators to calculate distance travelled by the ISS

(Resource URL: [https://www.raspberrypi.org/learning/astronaut-reaction-times/)](https://www.raspberrypi.org/learning/astronaut-reaction-times/%29)

1. **What You Need**
* Raspberry Pi
* SD card
* Keyboard and mouse
* power supply
* Display and connectivity cables (HDMI)
* SD card reader and writer

****

1. **Start Your Raspberry Pi**
2. Install an operating system
3. Plug your HDMI to your Raspberry Pi
4. First put in an SD card, plug in a power cord, keyboard and mouse
5. Press '+input' button on your computer
6. Follow the steps 'how to make a game'
7. Try it out
8. When fit to your expectations, save the game

# Install an Operating System Image

 NOOBS is an easy operating system installer which contains Raspbian. It also provides a selection of alternative operating systems which are then downloaded from the internet and installed.

* Using a computer with an SD card reader, visit <https://www.raspberrypi.org/downloads/noobs/> .
* Click on the Download ZIP button under ‘NOOBS (offline and network install)’, and select a folder to save it to.
* Extract the files from the zip.

Once you've downloaded the NOOBS zip file, you'll need to copy the contents to a formatted SD card on your computer.

To set up a blank SD card with NOOBS:

* Format an SD card that is 4GB or larger as FAT.
* Download and extract the files from the NOOBS zip file.
* Copy the extracted files onto the SD card that you just formatted so that this file is at the root directory of the SD card. Please note that in some cases it may extract the files into a folder; if this is the case then please copy across the files from inside the folder rather than the folder itself.
* On first boot the "RECOVERY" FAT partition will be automatically resized to a minimum, and a list of OSs that are available to install will be displayed.

In detail -> Format your SD card

It is best to format your SD card before copying the NOOBS files onto it. To do this:

Visit the [SD Association’s website](http://www.sdcard.org/) and download [SD Formatter 4.0](https://www.sdcard.org/downloads/formatter_4/) for either Windows or Mac.

* Follow the instructions to install the software.
* Insert your SD card into the computer or laptop’s SD card reader and make a note of the drive letter allocated to it, e.g. G:/
* In SD Formatter, select the drive letter for your SD card and format it.

Drag and drop NOOBS files

* Once your SD card has been formatted, drag all the files in the extracted NOOBS folder and drop them onto the SD card drive.
* The necessary files will then be transferred to your SD card.
* When this process has finished, safely remove the SD card and insert it into your Raspberry Pi.

First boot

* Plug in your keyboard, mouse and monitor cables.
* Now plug in the USB power cable to your Pi.
* Your Raspberry Pi will boot, and a window will appear with a list of different operating systems that you can install. We recommend that you use Raspbian – tick the box next to Raspbian and click on Install.
* Raspbian will then run through its installation process. Note this can take a while.
* When the install process has completed, the Raspberry Pi configuration menu (raspi-config) will load. Here you are able to set the time and date for your region and enable a Raspberry Pi camera board, or even create users. You can exit this menu by using Tab on your keyboard to move to Finish.

Logging in and accessing the graphical user interface

* The default login for Raspbian is username pi with the password raspberry. Note you will not see any writing appear when you type the password. This is a security feature in Linux.
* To load the graphical user interface type startx.
1. **Install Guide**

# Set The Stage With A Space Theme

1. If you are using a Raspberry Pi you can open Scratch by clicking on **Menu** and **Programming**, followed by **Scratch**.
2. Create a new file by selecting **File** and **New**.
3. Delete the Scratch Cat sprite by right-clicking on it and selecting **Delete** from the menu that is displayed.
4. For this project, you need a space-themed background and an Astronaut sprite. To add a background in Scratch, click on stage in the sprites palette and then click on Backgrounds next to the scripts tab.
5. Next, add a new sprite by clicking on the import a new sprite icon on the sprites palette (which looks like the image below), selecting Astronaut-Tim from the choices and clicking **OK**.



1. Save your Scratch project work by clicking on **File** and **Save As**. Name your program **Astronaut Reaction Game** and save it in your home directory or some place that you can find it later.

#  Create Variables To Store Data

1. To create a variable, click on Variables in the blocks palette and then click Make a Variable. The New Variable window opens and asks you to type a name for your variable.
2. Name the first variable time and ensure that for all sprites is checked before clicking **OK**. A variable holds a value that can be changed. The time variable you have created is an example of a value that can be changed and used inside different scripts. You will use it to store the reaction times of players.
3. You’ll see some orange blocks are added to your Variable area called time, and a small counter box will appear on the stage.

# Begin The Reaction Game Script

1. Click on your Sprite to select it in the sprites palette.
2. Select the When green flag is clicked control block from the blocks palette and place it onto the scripts area.
3. Then click on Looks and connect the say for 2 secs block to the first control block on the scripts area. Amend the text to say Hello! British ESA Astronaut Tim Peake here. Are you ready?.
4. Search US Keyboard on Google, go to images. Find where to press to insert ! mark. You need this step because you have a different type of key board.
5. Add a wait 1 secs block underneath.
6. Connect another say block and change the text to Hit Space!.
7. Click on Sensing and connect the reset timer block. This will set the timer to 0 so that you will get an accurate measurement of how long it takes for someone to hit the space bar.
8. Use the control block wait until and place a key space pressed? sensing block inside the white space of the wait until block. This will pause the program until the player presses the space bar.
9. Then connect another say block. Once the space bar has been pressed, you want to display the reaction to the player. To do this, you need to place an operators block called join hello world inside the white space in the say block. Replace the word world with the word seconds.
10. You will then need to replace the word Hello with another join hello world operators block, replacing the Hello text with Your reaction time was and the world text with the timer sensing block.
11. Finally, select the set time to block from the variables section and add it to your script. Place the timer sensing block inside where it reads 0.



1. Save your game and test it works by clicking on the green flag. When Tim says "Hit Space!", press the space bar. Your time should be displayed like this:



# Comparing Player’s Reaction Time To The ISS Orbit

If you are happy with your reaction game and have tested that it works, then you can move onto adding to the script to compare the player's reaction time to how fast the ISS is travelling, to calculate how far it would travel in that time.

1. First you will need to make a new variable called distance in the same way you did earlier.
2. Attach a set distance to variable block to your script. Place an operators multiply block 0\*0 inside where it reads 0. To calculate the distance travelled by the ISS you need to take the player's reaction time, which is stored in the time variable, and multiply it by 7. This is because on average the ISS travels 7 kilometres per second!
3. Add the time variable block into the right hand side of the multiplying operator and type 7 in the other side, so that the whole block reads set distance to time \* 7.
4. Next, add a wait 4 seconds control block.
5. Then add a say block. As in the previous step, place a join hello world block inside. Replace World with kilometres. Insert another join Hello World block to replace Hello. Replace the Hello text in this new Join block with the text In that time the ISS travels around. Then replace World with a round operator block and fill the white space with the distance variable block like this:



1. Save your game and test that it works by clicking on the green flag.



<https://www.raspberrypi.org/learning/astronaut-reaction-times/worksheet/>

1. **Conclusion**

Through this project, I got the idea of how the game is programmed. Before, I couldn’t guess how it works, but now I can at least imagine it. Exploring the technology is interesting because as we do, some things around us starts to make sense. And of course, you can learn how to use them.

1. **Bibliography**

# Websites

<https://www.raspberrypi.org/help/what-is-a-raspberry-pi/>

<https://www.raspberrypi.org/learning/astronaut-reaction-times/>

<https://www.raspberrypi.org/help/quick-start-guide/>

<https://www.raspberrypi.org/learning/astronaut-reaction-times/worksheet/>

<http://www2.le.ac.uk/departments/law/current/writing-guide-and-document-template>

 <http://blogs.cdl.ch/cg2015/>

 <http://www.raspberrypi.org/documentation/installation/installing-images/>

 <https://www.raspberrypi.org/downloads>

 <https://www.raspberrypi.org/documentation/installation/noobs.md>