Experimental Procedure

Introduction

Do you play video games? Have you ever wished you could change a feature in a game you have played, or even make your own game? That's exactly what you will get to do in this project! This project will show you how to get started with a simple arcade-style game, but you can add on to the game as much as you want. If you have not already, you will need to set up your Raspberry Pi before you begin. Here is a video of the basic game and some possible additions:

https://www.youtube.com/watch?v=tOAWSHGRocw

Starting Your Program

If you have never written a program in Scratch before, watch this video before you continue:

https://www.youtube.com/watch?v=X3XcWlw-lg

In this project you will be controlling Scratch characters on screen. The individual characters are called sprites, and the area where they move around is called the stage. Every new program starts out with a blank white stage and a single cat sprite, like this:

![Figure 2](image)

In this project you will write small sections of code and test them one at a time. This will give you a better understanding of how your video game works, so you can make your own changes later.

The first thing you will do is write a simple program that lets you move the cat sprite around the screen. Create the following program (see the tips below Figure 3 if you need help):
To create a variable, click **Data** then **Make a variable**.

The **move ___ steps** and **turn ___ degrees** blocks can be found under **Motion**.

Now, run your program by clicking the green flag. The cat should continuously move forward, while you use the arrow keys to steer it left and right. Try stopping your program, entering different values for the **cat_speed** and **cat_turning** variables, then running the program again. Experiment with a few different values until you feel like you have good "control" of the cat. For example, if the cat moves too fast to control easily, then decrease the speed. If it takes too long to turn, increase the turning angle. There are no right or wrong numbers; it is a matter of personal preference.

Finally, the rectangular blue button in the upper-left corner of the stage can be used to make the stage full-screen. You can use this button to make the game full-screen when you play, then click it again when you need to edit your program.

---

**Adding a Second Sprite**

Remember from the introduction video, above, that there are two sprites in the game, the cat (which you control), and a dog, which automatically chases the cat. Next, you will need to create the dog sprite and give it its own program (each individual sprite's program is called a **script**). Click the **Choose sprite from library** button next to "New sprite".
Search for the Dog1 sprite under Animals then click OK (of course, if you don't want to do a classic cat-and-dog chase, you can pick something else!). You will now see both the dog and the cat in the Sprites area of the screen. When you select the Dog1 sprite, you will see that the script area of the screen is now blank. The script you wrote earlier only applies to the Cat1 sprite.
Write a new script for the dog sprite (make sure you have the dog sprite selected, not the cat):

Now, run your program again. You should still be able to control the cat, and the dog will automatically follow the cat around! Adjusting the `dog_speed` variable will change the game's difficulty. The faster the dog moves, the harder it will be to avoid.

Finally, do not forget to save your program!

**Making it a Game**

So, now you can control an on-screen character, but there is no way to win or lose your game. The following improvements will help make this feel more like a real video game:

- Make the game end when the dog catches the cat.
- Add a way to keep score.
- Make the dog and cat go back to their original starting locations when you run the program.
- Hide some of the variables that clutter up the stage area.

To make these improvements, modify your cat's script to look like this:
Next, modify the dog's script to look like this (remember that the dog sprite has its own "Scripts" tab, so you have to click on the dog to edit its program).

Now, when you run the program, the cat should automatically go to the middle of the screen, and the dog should go to the left side of the screen. When the dog catches the cat, the cat will stop and display your final score. You can restart the game by clicking the green flag.

Do not forget to save your work!

**Building Your Own Controller**

To build your circuit, you will need to use a breadboard. If you have not used a breadboard before, watch this video before you continue:

https://www.youtube.com/watch?v=6WReFkUIk

If you have ever played a video game console like Xbox®, PlayStation® or Nintendó®, then you have used a video game controller. Have you ever wondered how the inside of a controller works? Now you will get a chance to build your own simple controller on a breadboard using two buttons. This video will give you a quick introduction to the buttons that come with your kit:

https://www.youtube.com/watch?v=0hZ4pQ6KCPA

To build the circuit, you will just need jumper wires and the two pushbuttons.
Follow the directions below to assemble the circuit. Note that the buttons each have four pins.
<table>
<thead>
<tr>
<th>Part</th>
<th>Picture Reference</th>
<th>First Hole</th>
<th>Second Hole</th>
</tr>
</thead>
<tbody>
<tr>
<td>Black M-M jumper wire</td>
<td><img src="image1" alt="Image" /></td>
<td>J17 (Pi Wedge GND)</td>
<td>Ground bus (right side)</td>
</tr>
<tr>
<td>Blue M-M jumper wire</td>
<td><img src="image2" alt="Image" /></td>
<td>J22</td>
<td>Ground bus (right side)</td>
</tr>
<tr>
<td>Blue M-M jumper wire</td>
<td><img src="image3" alt="Image" /></td>
<td>J27</td>
<td>Ground bus (right side)</td>
</tr>
<tr>
<td>Red M-M jumper wire</td>
<td><img src="image4" alt="Image" /></td>
<td>J20</td>
<td>A7 (Pi Wedge G4)</td>
</tr>
<tr>
<td>Green M-M jumper wire</td>
<td><img src="image5" alt="Image" /></td>
<td>J25</td>
<td>A8 (Pi Wedge GE1)</td>
</tr>
<tr>
<td>Pushbutton</td>
<td><img src="image6" alt="Image" /></td>
<td>E20</td>
<td>F20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>E22</td>
<td>F22</td>
</tr>
<tr>
<td>Pushbutton</td>
<td><img src="image7" alt="Image" /></td>
<td>E25</td>
<td>F25</td>
</tr>
<tr>
<td></td>
<td></td>
<td>E27</td>
<td>F27</td>
</tr>
</tbody>
</table>
Your controller should now look like this:
Modifying Your Program

Right now, your program is set to work with the arrow keys. You need to modify it to work with the buttons on your controller. Change the script for your cat sprite to look like this, by adding two if statements to monitor the input pins, and deleting the **when key pressed** sections. (right-click them and select "delete").

![Diagram of code changes](image)

**Figure 14**

Now, run your program again. Instead of using the arrow keys, you should be able to control the cat by pushing the two buttons on your breadboard.

?- Troubleshooting: My controller does not work!  [Show](#)

Going Further

Once you have your basic game working, there are plenty of changes you can implement to make the game even more exciting. Here are just a few ideas:

- Use the **if on edge, bounce** block under **Motion** to make sprites "bounce" off the edges of the stage, instead of running partially off the screen and getting stuck.
- Add "power-ups" to the game. For example, add more sprites that give the cat a speed boost (by increasing the "cat_speed" variable) when it touches them. You can do this using the **touching ___?** block under **Sensing**.
- For a really advanced challenge, make the power-up sprites disappear when the cat touches them, and reappear in random locations. You can do this with the **show and hide** blocks under **Looks** and the **pick random ___ to ___** block under **Operators**.
- Give the game a selectable difficulty level, for example, easy, medium, and hard, where the dog moves faster with each increase in difficulty. You can make "buttons" for the player to click on to select the difficulty using the **when this sprite clicked** block under **Events**.
- Rather than letting the game run forever until the dog catches the cat, make it so the player wins the game when reaching a certain score.
- Create a feature to keep track of the high score.
- Make the game even harder by adding a second sprite that also chases the cat.

Frequently Asked Questions (FAQ)

**Kit General Questions**

- Who is the kit appropriate for? ([@gen-question2](#))
- Are the kit parts reusable? ([@gen-question3](#))
- Aren't there other Raspberry Pi kits on the market? How is yours different? ([@gen-question5](#))
- I already have a Raspberry Pi. Can I just buy the circuit parts separately? ([@gen-question-1have](#))
- What programming language do the projects use? ([@gen-question10](#))

**Setting Up and Using Your Raspberry Pi FAQ**
These answers apply to the Raspberry Pi Model 3B+ which comes with the Raspberry Pi Projects Kit (http://www.sciencesbuddies.org/store-send?utm%3A%2F%2Fwww.homesciencesstore.com%3Fproduct%3Draspberry-pi-projects-kit%3Fm%3A3B%3B1). If you are using a different Raspberry Pi model you will need to do some research on your own—we suggest starting with the Raspberry Pi Foundation’s Setting up your Raspberry Pi page (https://projects.raspberrypi.org/en/projects/raspberry-pi-setting-up/).

- How do I connect my Raspberry Pi to my TV or computer monitor? (Question1)
- Can I use a laptop as a display and/or keyboard? (Question1)
- How do I connect my Raspberry Pi to the internet? (Question2)
- How do I shut down or reboot my Raspberry Pi? There’s no power button! (Question3)
- How can I adjust the Raspberry Pi’s display resolution? (Question4)
- I have everything connected properly, why can’t I hear any sound? (Question5)
- Why won’t my Raspberry Pi turn on? (Question6)
- My Raspberry Pi starts to boot up, but then it freezes or the screen goes blank. What is wrong? (Question7)
- My Raspberry Pi froze and is not responding to mouse or keyboard input. What should I do? (Question8)
- My Raspberry Pi is acting strangely (it suddenly will not boot up properly, certain programs do not work, etcetera). What is wrong? (Question9)
- I think I corrupted my Raspberry Pi’s SD card. What should I do? (Question10)
- I need help with a question, related to my Science Buddies Raspberry Pi Projects Kit or Raspberry Pi Circuits Parts Only Kit, not listed here. Who can I ask? (Question11)

Kit General Answers

Q: Who is the kit appropriate for?
A: The kit is meant for anyone (ages 8 and up) who wants to learn some basic programming and electronics skills while having fun. Students up to age 10, or older if their reading skills are behind grade level, may need adult assistance in reading and following the on-screen instructions. The projects included in the kit were beta tested and approved by students ages 8 to 16.

Q: Are the kit parts reusable?
A: Yes, all the electronics components in the kit can be re-used to do new projects or to repeat the projects.

Q: Aren’t there other Raspberry Pi kits on the market? How is yours different?
A: Yes, there are other Raspberry Pi kits, and some of them are quite good! The Raspberry Pi Projects Kit (http://www.sciencesbuddies.org/store-send?url=https%3A%2F%2Fwww.homesciencesstore.com%2Fproduct%2Fraspberry-pi-projects-kits%3Fm%3A3B%3B1) and Raspberry Pi Circuit Parts Only Kit (http://www.sciencesbuddies.org/store-send?url=https%3A%2F%2Fwww.homesciencesstore.com%2Fproduct%2Fraspberry-pi-circuits-parts-only%3Fm%3A3B%3B1) have been designed to contain the specific materials needed to do the accompanying Science Buddies Raspberry Pi projects. Our kit and associated projects are specifically meant for people who have no prior experience programming or connecting circuits. The projects are 100% beginner friendly with clear on-screen instructions, pictures, and videos. We think the kit, with its associated projects, is one of the most fun kits out there! But, if you are already an ace programmer or electronics guru, you may not find this the best fit for your own personal use. Even so, it may be a fun way for you to introduce others to programming and electronics.

Q: I already have a Raspberry Pi. Can I just buy the circuit parts separately?

Q: What programming language do the projects use?
A: The projects use Scratch 2. Scratch is a “graphical” programming language developed by the MIT Media Lab. It allows you to write code by clicking, dragging, and snapping together color-coded blocks. This allows beginners to write working code without worrying about formatting or typos. On the Raspberry Pi, Scratch allows you to control the general purpose input and output (GPIO) pins so your program can interact with a circuit in the physical world.

Note: three different versions of Scratch (1, 2, and 3) are available for the Raspberry Pi. The instructions for the Science Buddies projects (including example code) are specifically written for Scratch 2, which runs well on the Raspberry Pi model 3B+. If you have an older model Raspberry Pi, Scratch 2 may run more slowly, or may not run at all. If you want to use a different version of Scratch, you will need to consult the official documentation for Scratch 1.4 (https://www.raspberrypi.org/documentation/usage/gpio/scratch-1/README.md) or Scratch 3 (https://www.raspberry-pi.org/blog/scratch-3-desktop-hon-raspbian-on-raspberry-pi/) and modify the programming steps accordingly.

Setting Up and Using Your Raspberry Pi FAQ Answers

Q: How do I connect my Raspberry Pi to my TV or computer monitor?
A: The easiest way to set up your Raspberry Pi is to use an HDMI cable (included in the Science Buddies Raspberry Pi Projects Kit (http://www.sciencesbuddies.org/store-send?url=https%3A%2F%2Fwww.homesciencesstore.com%2Fproduct%2Fraspberry-pi-projects-kits%3Fm%3A3B%3B1)) to connect to a TV or computer monitor that has built-in speakers. If you are using a computer monitor with an HDMI port but no built-in speakers, you will also need separate speakers or headphones with a 3.5 mm audio plug (a regular “headphone jack”).

If your TV or monitor does not have an HDMI port, you will need an HDMI to DVI or HDMI to VGA adapter (see pictures in table below). DVI and VGA do not transmit sound, so you will need separate headphones or speakers if you are using one of those options.
Q: Can I use a laptop as a display and/or keyboard?
A: The short answer is "not easily." Many newer laptops have HDMI ports, but they only function as HDMI out, to send a video signal from the laptop to a television or projector. They do not work as HDMI in to display an external signal on the laptop's screen. The laptop's keyboard is only designed to work with the laptop itself, not as a standalone keyboard for an external device like the Raspberry Pi.

The longer answer is that, similar to the Remote Desktop feature on Windows and Mac computers, you can use special software to remotely operate a Raspberry Pi that is connected to the internet. This would allow you to control a Raspberry Pi using your laptop's screen and keyboard. This option is only recommended for advanced users, and you can find instructions here (https://www.raspberrypi.org/documentation/remote-access/vnc/).

Q: How do I connect my Raspberry Pi to the internet?
A: Unlike earlier models, the Raspberry Pi 3B+ contains built-in Wi-Fi functionality. It does not require an external USB Wi-Fi adapter. You can connect your Raspberry Pi to the internet by clicking the internet icon in the taskbar and searching for available Wi-Fi networks, just like you would on a Windows or Mac computer. Your Raspberry Pi also has an ethernet port, which you can use to plug directly into a router for a hardwired connection.

Q: How do I shut down or reboot my Raspberry Pi? There's no power button!
A: Unlike most computers, the Raspberry Pi does not have a power button. You can shut down or reboot by clicking the raspberry icon in the upper left corner of your desktop, then select Shutdown. After the Raspberry Pi has shut down, it is safe to unplug the micro-USB power cable. Plug the cable back in to reboot. Important: never unplug the power cable while the Raspberry Pi is still running. This can corrupt the SD card.

Q: How can I adjust the Raspberry Pi's display resolution?
A: Click the Raspberry Pi logo in the top-left corner of your desktop. Select Preferences, then Raspberry Pi Configuration, then click the Set Resolution... button on the System tab.

Q: I have everything connected properly. Why can't I hear any sound?
A: Right-click the speaker icon on the desktop taskbar. This allows you to manually select HDMI or analog (the headphone jack) for sound output. Make sure you have the proper output selected. Also, make sure your Scratch program is set to play a sound. You can write a simple program to test if your sound is working using the "when space key pressed" and "play sound meow" blocks.

Q: Why won't my Raspberry Pi turn on?
A: If your Raspberry Pi will not turn on (the screen remains blank after everything is plugged in), go through this checklist to make sure everything is set up properly.

1. Make sure your SD card is pushed in all the way (see Figure 1).
2. Make sure the red power LED on your Raspberry Pi (labeled "PWR," near the micro-USB port, see Figure 2) is on. This means the Raspberry Pi is receiving power from the micro-USB port. If the LED is not on, make sure you pushed the micro-USB connector into the micro-USB port all the way.
3. When you first plug the micro-USB cable in, the green LED (labeled "ACT," next to the PWR LED, see Figure 2) should flash several times. This LED flashes when the Raspberry Pi reads data from the SD card. After the Raspberry Pi is done booting up, it should turn off. If it does not flash at all, your SD card might not be inserted properly. Go back to step 1.
4. Make sure your display (television or monitor) is turned on. If your display is turned off, you will not see anything on the screen, even if the Raspberry Pi is on.
5. Make sure your display is set to the correct input. Many modern TVs have more than one HDMI input, and some computer monitors have DVI or VGA inputs in addition to HDMI.
Q: My Raspberry Pi starts to boot up, but then it freezes or the screen goes blank. What is wrong?
A: There may be a problem with your Raspberry Pi or SD card. If you are using the Raspberry Pi or SD card that came with the Raspberry Pi Projects Kit or Raspberry Pi Circuit Parts Only Kit purchased from our partner Home Science Tools, please contact them directly at service@homesciencetools.com for assistance.

Q: My Raspberry Pi froze and is not responding to mouse or keyboard input. What should I do?
A: First, be patient and give the Raspberry Pi a few minutes to try and process whatever it was doing. If you click on a whole bunch of things in rapid succession, or run a really complicated Scratch program, the Raspberry Pi might slow down or freeze temporarily.

Next, if you are using a wireless keyboard and mouse, make sure they have fresh batteries.

Finally, as a last resort, if your Raspberry Pi is not responding, unplug the micro-USB cable and plug it back in. In general, you want to avoid doing this, because suddenly cutting power to the Raspberry Pi without properly shutting it down first can corrupt the SD card, and prevent the Raspberry Pi from working properly.

Q: My Raspberry Pi is acting strangely (e.g. it suddenly will not boot up properly, certain programs do not work, etc.). What is wrong?
A: If your Raspberry Pi is not "dead," but seems to be behaving strangely, there is a chance that your SD card has become corrupted. This can happen if you unplug the
Raspberry Pi's power cord without properly shutting it down first. See the next question.

Q: I think I corrupted my Raspberry Pi's SD card. What should I do?
A: If the SD card came with the Raspberry Pi Projects Kit or Raspberry Pi Circuit Parts Only Kit you purchased from our partner Home Science Tools, please contact them directly at service@home-science-tools.com for assistance. Make sure to include a detailed description of the problem you are having. They will work with you to resolve the issue.

Q: I need help with a question, related to my Science Buddies Raspberry Pi Projects Kit or Raspberry Pi Circuits Parts Only Kit, not listed here. Who can I ask?
A: Science Buddies has a special area of our Ask an Expert forums dedicated to the Raspberry Pi Projects Kit. Please note that the forums are staffed by volunteers, and it may take a few days to get a response. If you are doing a science project, please do not wait until the day before the project is due to ask an urgent question.