Beginner's Python workshop: Instructions for Instructors

You should:

- Download the files for the workshop from: `https://github.com/rbricheno/pygamezero-workshop`
- Familiarise yourself with the workshop booklet and these instructions
- Set up and test the classroom computers (especially sound!)
- Ideally have some experience programming Python, to be able to answer more complicated questions.

The booklet for the workshop **pygame-zero-intro.pdf** should be self-explanatory. Participants should be able to start and continue by themselves as long as the workshop computers are set up correctly. Older children should be able to complete all of the workshop by themselves within the hour. Younger children probably won't finish the workshop without help.

Set up

The workshop booklet is written with a Raspberry Pi in mind, so check that the instructions in the booklets reflect what you will be doing before you give them out. If you need to change the booklet, you can edit **pygame-zero-intro.odt** using LibreOffice Writer and then export your customised booklet as a PDF.

mu and **Pygame Zero** should both already be installed on the Raspberry Pi, if you are using a different computer you will need to make sure they are installed.

If a directory called "`mu_code`" does not exist in the user's home directory, create it.

Copy the images, music, and sounds directories from the workshop download into the "`mu_code`" directory. They can be merged with the existing directories with the same names (if there are any). Workshop participants should work in the "`mu_code`" directory and save their Python file there. This way, Pygame Zero will have access to the images, sounds and music.

Additionally, make a copy of all the workshop files in a directory called "`pygamezero-workshop`" in the user's home directory (not strictly required, but recommended as it might be useful in case we need to copy and paste code, or in case a user deletes all the files in "`mu_code`" for example).

To test the workshop, open mu then copy the content of the file "`pygamezero-workshop/11_game_sound.py`" into the mu editor window. Play, and check:

- You can save the file as a new file into the mu_code directory
- The game runs
- Sound can be heard

Tips

Make sure sound is working before the workshop. We used CEEDs for this workshop and found that we needed to install a driver. We also found that the sound did not work properly if we put it on to full volume. When we set the sound to half volume and left it there, the sound worked fine.

Try and circulate the room regularly, making sure nobody is left behind. In particular, in step 3 there is a lot of typing required. You may want to copy and paste the code for the `check_keys` function (or other problem areas) from the sample code stored in the "`pygamezero-workshop`" directory for users who are having trouble.

Check that users are not typing in all of the listings for each step one after another. Remind them that they only need to add the changes to the code that they already have rather than adding in the entire page of code to the end of what they already have.

For participants who finish the workshop early, tell them the secret: that there is a whole other game that they can make now, by replacing the images and sounds. We have provided images called "shark" (player), "fish" (food), "puffer" (treat), and "sea" (field). We have also provided a "crunch" and a "munch" sound. They can now make an underwater game! Encourage them to mix and match the sounds and images, and add their own code to change the game's behaviour such as changing the score for each piece of food.

Have fun!