Development of a Scratch Extension to Use Python in the Scratch Editor

Tomoya Tokairin * [a] and Mugen Nakano [b]

[a] Department of Production Systems Engineering, National Institute of Technology, Hakodate College, Hakodate, Japan
[b] Production System Engineering Course, National Institute of Technology, Hakodate College, Hakodate, Japan

*tokai@hakodate-ct.ac.jp

We have developed a Scratch extension that enables teachers and students to execute Python code directly within the Scratch editor. This extension supports teachers who use Scratch in STEAM education by facilitating the preparation of teaching materials. It also serves as an effective educational tool for students to learn Python. In this paper, we introduce the extension and present several examples to demonstrate its application.

Keywords: *educational tool, STEAM education, visual programming, development environment*

Introduction

Scratch, developed by the Scratch Foundation (2013), is a visual programming language that allows users to create programs by combining graphical blocks. It is designed to promote computational thinking and problem-solving skills of students. Scratch is not only a powerful tool for teaching programming, but also an effective platform for promoting STEAM (Science, Technology, Engineering, Art, and Mathematics) education. In fact, many teachers around the world incorporate Scratch into their classroom activities.

On the other hand, Scratch has certain limitations in handling the complex calculations that are often necessary in STEAM education. And students are generally unfamiliar with programming. As a result, teachers often need to prepare large and complex Scratch scripts in advance to support classroom activities.

Instead of creating large and complex Scratch scripts, teachers can also utilize Scratch extensions. For example, Scratch has some built-in extensions such as "Music" (which plays instruments) and "Translate" (which translates text into other languages). However, developing original extensions requires advanced programming skills.

To address this issue, Tokairin, T. & Nakano, M. (2025) have developed a Scratch extension that enables teachers and students to execute Python code directly within the Scratch editor. The extension supports teachers by facilitating the preparation of teaching materials because they can utilize many powerful Python modules to perform complex calculations within the Scratch editor.

Usage of the extension

First, open any web browser and navigate to

https://tmytokai.github.io/scratch-python/

Next, in the Scratch editor, click the button at the bottom-left corner to open the "Extensions Library". Then click the icon shown in Figure 1. After that, a set of blocks will appear in the “Block Palette” as illustrated in Figure 2.

The top block shown in Figure 2 loads Python code from a specified URL. The remaining blocks execute functions defined within the Python code.

Once Python code is loaded, no further network access is required. Therefore, Python code is executed very quickly. The appendix illustrates the technical architecture of the extension.

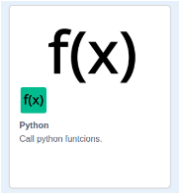

Figure 1: Icon of the extension.

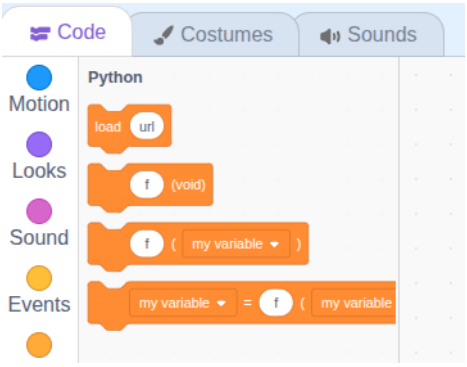

Figure 2: Blocks of the extension.

Examples

Example 1: Sorting

Here is an example of a sorting algorithm. Although sorting is one of the most fundamental algorithms, Scratch does not provide built-in blocks for sorting. As a result, users must create large and complex scripts on their own. Therefore, this example is useful in many applications.

First, upload the Python code shown in Figure 3 to any web server. This code includes a sorting function called "sort(x)". Next, execute the Scratch script shown in Figure 4. This script initializes a list x and sorts it by calling "sort(x)". The result is shown in Figure 5.

```
def sort(x):
    return sorted(x)

__export__ = ['sort']
```

Figure 3: Python code of example 1.

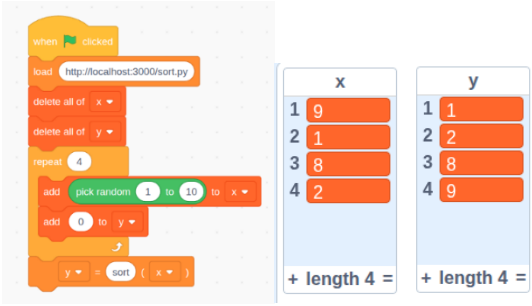

Figure 4: Scratch script of ex.1. *Figure 5: Result of ex.1.*

Example 2: Power spectrum

Complex calculations frequently arise in STEAM education. As an example, the power spectrum of list x can be easily obtained by Python code shown in Figure 6 and Scratch script shown in Figure 7. The result is shown in Figure 8.

Conclusions

We introduced a Scratch extension that executes Python code within the Scratch editor. This extension can be applied to a wide range of cases beyond the examples presented in the previous section. Therefore, it significantly reduces the workload for teachers.

Moreover, it serves as an precursor educational tool for students to learn Python in a familiar programming environment, i.e. Scratch.

Appendix

This extension is built on top of PyScript. PyScript, developed by the PyScript Development Team(2022), is a web-based platform that executes Python code directly in a web browser. For example, the HTML code shown

```
import numpy as np

def powerspectrum(x):
    return (np.abs(np.fft.fft(x))**2).tolist()
.
__export__ = ['powerspectrum']
```

Figure 6: Python code of example 2.

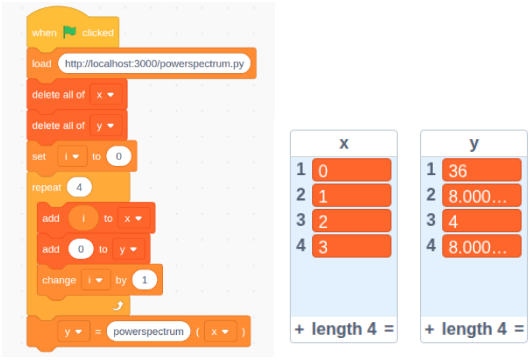

Figure 7: Scratch script of ex.2. *Figure 8: Result of ex.2.*

in Figure 9 displays random numbers in the browser's debug console.

```
<!doctype html>
<html>
<head>
<link rel="stylesheet"
href="https://pyscript.net/releases/2025.3.1/core.css">
<script type="module"
src="https://pyscript.net/releases/2025.3.1/core.js"></script>
</head>
<body>
<py-script>
    import random
    for i in range(5):
        print(random.randint(1, 100))
</py-script>
</body>
</html>
```

Figure 9: Example of PyScript.

References

PyScript Development Team. (2022). *What is PyScript?*. Retrieved from https://docs.pyscript.net/2025.3.1/user-guide/what/

Scratch Foundation. (2013). *About Scratch*. Retrieved from https://scratch.mit.edu/about/

Tokairin, T. & Nakano, M. (2025). Development of Scratch3 extension to collaborate with Python. *Proceedings of 30th KOSEN symposium 2025 in Okayama*, 241201.