


Chapter 18

Using Motion Infographics to Teach Computer Programming Concepts

Apostolos Syropoulos

 <https://orcid.org/0000-0002-9625-1482>

Greek Molecular Computing Group, Greece

ABSTRACT

Although programming languages are expressive tools, their expressive power is quite limited. A direct consequence of this is that novice users have difficulty grasping the basic programming constructs mainly because there is a discrepancy between natural languages and programming languages (e.g., How do we express common human tasks in Python?). A relatively easy way to tackle this discrepancy is to use motion infographics. However, when pupils and students are familiar with abstraction, a basic idea of computational thinking, then one can use motion infographics that show familiar objects and/or ideas that should be used to teach the corresponding programming constructs. The author has used this approach to create motion infographics that explain conditional and repetitive constructs. However, this does not really work with recursion as one has to resort to the use of trees to explain this programming concept. All the motion infographics presented here have been produced with XeLaTeX, an open source tool that can be used to produce excellent printed/electronic documents.

INTRODUCTION

At the time this chapter was written, I was working on a book whose goal was to teach programming to everyone with practically no previous knowledge. I have to say that writing a textbook about programming is by no means an easy task, although many think quite the opposite. Experience shows that when one introduces new concepts and ideas, drawings that visualize them are extremely helpful. This happens mainly because humans comprehend images, pictures, and drawings easier than oral or written descriptions. Thus, when one speaks about variables, she should use visual aids to make the idea of a variable more accessible. For example, the idea that a variable is a container that holds numbers and/

DOI: 10.4018/978-1-6684-5934-8.ch018

or character strings is quite instructive. However, single pictures are not sufficient for other commands, because most of them involve some change to the state of a computer and/or the values of some variables. In this case, it is far more instructive to use a series of images and/or drawings or, even better, some sort of short animations that will visualize all the action that takes place when a specific command is executed. Naturally, animations are useful in all kinds of presentations and eBooks. Unfortunately, authors of traditional books on programming have no choice but to use series of images and/or drawings.

Once one has decided what each animation should include, it is necessary to also decide how to realize the idea. Since I am a proponent of Open-Source software, I have opted to realize my creations as Animated PNG (APNG) files. The frames of my APNG motion infographic should be simple Portable Network Graphic (PNG) files that I had to create somehow. Since I am an *amateur* programmer, I prefer to write code that creates my frames. For this reason, I have opted to use tools from the company of TeX and its friends.

Computational thinking was introduced by Jeannette Marie Wing (Wing, 2008) [see also (Pollak & Ebner, 2019) for a more recent overview of computational thinking; and also (Neumann et al., 2021) for a guide on how to teach computational thinking]. One could define computational thinking as a problem-solving methodology that makes use of the fundamental notions of informatics. Computer programming and computational thinking are not the same thing but knowing how to program helps people to apply computational thinking in many and different kinds of problems. This means that if one provides learners with motion infographics that easily allow them to understand and, consequently, generalize the ideas presented by them, then learning would be much easier. I call the process of generating such motion infographics *inverse computational thinking*. Thus, my aim is to show how to generate motion infographics with open-source tools and to explain how to apply inverse computational thinking to create them.

Python¹ is a widely used programming language that was designed by Guido van Rossum, a Dutch programmer. The language is quite popular² and, based on my own experience as a computer science teacher, it has an *increasing-returns* learning curve, which means that learners progress slowly at the beginning but their skills rise over time until full proficiency is obtained. In addition, Python is widely available, and one can use it on smartphones, tablets, desktop computers, etc. In addition, one can run Python programs using a web browser since there are some web pages that provide this ability.

Plan of the Chapter

First of all, I will describe how to generate drawings and animations using Open Source tools. In particular, I will explain what steps are needed in order to create an APNG file. But I will not give a recipe describing how to create any drawing. Instead, I will just describe the tools that I have used to create my own drawings. Then, I will discuss what kind of drawings and animations can be used to teach basic computational constructs—conditional and repetition commands. Next, I will discuss how one can use infographics to allow learners to get an understanding of recursion, which is a powerful yet not so obvious programming technique. The chapter will conclude with the customary conclusions section.

Creating Infographics Using Open-Source Tools

I am sure that there are proprietary tools that can be used to construct infographics, however, I am in favor of open-source tools for two reasons. First of all, such tools can be improved by contacting the author and asking him/her to include a new feature that will increase their functionality. In most cases, authors

13 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/chapter/using-motion-infographics-to-teach-computer-programming-concepts/313905

Related Content

Contemporary Issues on Agriculture and Health: Financial Framework, Reforms, and Competition Policy

Elena di Paolo, Giustino Lo Conte, Federica Correnteand Gianclaudio Festa (2022). *Handbook of Research on Healthcare Standards, Policies, and Reform* (pp. 195-216).

www.irma-international.org/chapter/contemporary-issues-on-agriculture-and-health/302492

Technology in Education, Win or Fail?: Are Teacher Candidates Being Prepared?

Elizabeth Gound (2022). *Handbook of Research on Global Education and the Impact of Institutional Policies on Educational Technologies* (pp. 243-266).

www.irma-international.org/chapter/technology-in-education-win-or-fail/291759

The Influence of Culture on Innovation and CSR Practices

Carla Marisa Magalhães (2024). *Research Anthology on Business Law, Policy, and Social Responsibility* (pp. 1678-1701).

www.irma-international.org/chapter/the-influence-of-culture-on-innovation-and-csr-practices/335790

International Healthcare Experiences: Caring While Learning and Learning While Caring

Jon P. Wietholter, Renier Coetzee, Beth Nardella, Scott E. Kincaidand Douglas Slain (2019). *Healthcare Policy and Reform: Concepts, Methodologies, Tools, and Applications* (pp. 1262-1288).

www.irma-international.org/chapter/international-healthcare-experiences/209180

The Challenges Behind the Development of an E-Mentoring Model for Online Distance Education: A Pilot Experience

Lina Morgado, Ana Paula Afonso, Nathalie Ferretand Marta Gomes (2022). *Handbook of Research on Global Education and the Impact of Institutional Policies on Educational Technologies* (pp. 363-376).

www.irma-international.org/chapter/the-challenges-behind-the-development-of-an-e-mentoring-model-for-online-distance-education/291765