

## Visual Programming for Visual Communication Design Students

**Yusup Sigit Martyastiadi**

Interaction Design, Visual Communication Design,  
Universitas Multimedia Nusantara  
E-mail: yusup.martyastiadi@umn.ac.id

### **Abstract**

Programming for Visual Communication Design students is a challenge in learning it. Their competences are developing visual communication materials for interactive media designs. Programming competence in learning of interaction design in the Visual Programming course focuses on programming logic. This study observes the use of the method of visual programming using the Bolt add-on (visual scripting). This data collection was carried out for Visual Communication Design students, Universitas Multimedia Nusantara when they took Visual Programming courses. The indicators of this research are studying the understanding of programming logic from several materials and the learning outcomes of each observed material. The researcher found positive results using visual scripting in the process of understanding programming logic for Visual Communication Design students on some of the material observed. In general, it can be concluded that the Bolt add-on in the UNITY game engine software can help students in working on interactive media design projects.

**Keywords:** *visual programming, interaction design, programming logic*

### ***Pemrograman Visual untuk Mahasiswa Desain Komunikasi Visual***

#### ***Abstrak***

*Pemrograman bagi mahasiswa Desain Komunikasi Visual menjadi tantangan tersendiri dalam mempelajarinya. Kompetensi mereka adalah mengembangkan materi komunikasi visual untuk desain media interaktif. Kompetensi pemrograman dalam pembelajaran desain interaksi pada mata kuliah Pemrograman Visual berfokus pada logika pemrograman. Penelitian ini mengamati penggunaan metode pemrograman visual dengan menggunakan aplikasi tambahan Bolt (visual scripting). Pengumpulan data ini dilakukan untuk mahasiswa Desain Komunikasi Visual Universitas Multimedia Nusantara saat mereka mengambil mata kuliah Pemrograman Visual. Indikator penelitian ini adalah mempelajari pemahaman logika pemrograman dari beberapa materi dan hasil belajar dari setiap materi yang diamati. Peneliti menemukan hasil positif menggunakan pemrograman visual dalam proses*

*pemahaman logika pemrograman bagi mahasiswa Desain Komunikasi Visual pada beberapa materi yang diamati. Secara umum dapat disimpulkan bahwa aplikasi tambahan Bolt pada piranti lunak UNITY (game engine) dapat membantu mahasiswa dalam mengerjakan proyek perancangan media interaktif.*

***Kata kunci:*** pemrograman visual, desain interaksi, logika pemrograman

## **INTRODUCTION**

Programming is an activity in the field of technology that is not easy to learn and use (Chao, 2016). Programming requires special skills to master it holistically (Myers, 1990). Basically, it takes stages in learning, by learning the programming language first and after that learning its application and use. Prospective programmers need to be focused and patient in learning these skills and consistently practice achieving their capacity as reliable programmers. The process of learning programming can be self-taught (non-academic) or take the academic path in college.

Visual Communication Design (VCD) is the study of the application of art and design in communicating messages visually. Through this study program, the students will understand the application of art and design in creating solutions to problems that exist in society. This is so that they can design visual communication in various media to create change for the surrounding environment. The VCD study program is a place for students to develop their creativity and innovation as well as their skills in the field of art - design to suit global competencies and future industrial needs. The UMN's VCD study program emphasizes the adaptation and modification of archipelago concepts in design works, both for print and digital needs (*Desain Komunikasi Visual*, 2022).

In the VCD study program, there is a Visual Programming course with code course ID 490. The description of this course is "The Visual Programming course is a basic introduction course for programming for interactive media. In this course, students will learn about the introduction and basic principles of programming logic for producing interactions in application projects and games. So that students are expected to be able to produce a digital interactive work that is using simple programming" (Martyastiadi, 2022).



Figure 1. Bolt in Unity Asset Store (Web Version)

Source: <https://assetstore.unity.com/packages/tools/visual-scripting/bolt-163802>

Prior to 2021, this Visual Programming course uses the C# programming language in the learning process for developing interactive media and games. The C# programming language is used in the UNITY game engine (Okita, 2020). As time goes by, starting in the Odd Semester (1<sup>st</sup> study period) of 2021/2022 (August 2021), the Visual Programming course introduces Bolt as visual scripting in the lecture process. The Bolt add-on can be downloaded freely from Unity Asset Store (as shown in Figure 1). However, researcher still include material using the C# programming language partially as part of the observation and comparison of understanding programming logic.

There are several visual programming applications developed with the aim for learning programming, including non-IT students (Broll et al., 2017; Chao, 2016; Ray, 2017). Those application are called visual programming languages (VPLs) (McGuffin & Fuhrman, 2020). On the other hand, these applications are called as Visual Scripting Languages (VSL) (Renger, 2022). Recently, visual programming languages (VPLs) is integrated in some game engine software (Chu & Zaman, 2021), such as Bolt in Unity game engine and so on. The Bolt as a visual scripting add-on provides easy options for users who are not familiar with complex programming languages (Knutsen, 2021; Unity Technologies, 2020). The options (as seen in Figure 2) are naming commands using the language we usually use (Human Naming) and naming instructions for programmers (Programmer Naming). The Human Naming (HN) scheme makes it easy for users to find and understand program instructions to be used. In addition, the commands are easy to

identify because they use colloquial terms. This is different from the Programmer Naming (PN) scheme where it uses terms that not everyone knows. For example, "int" in the program command in PN, while the program command in HN directly says "integer".

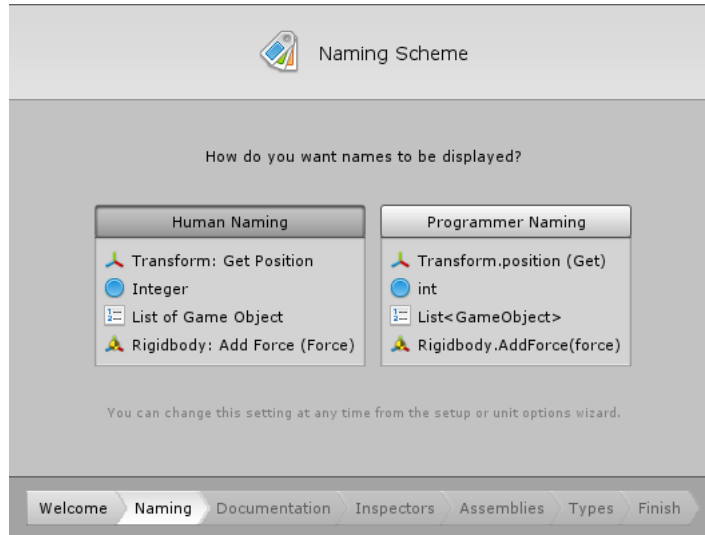


Figure 2. Naming Scheme in Bolt

Source: <https://docs.unity3d.com/2019.3/Documentation/uploads/Main/bolt-setupwizard2.png>

Figure 3 is the example of code using Bolt as visual scripting. McGuffin & Fuhrman (2020) point out that the Bolt uses “curves or line segments to connect blocks, and these connections may be equivalent to using temporary variables in a text-based language to store input and output values”.

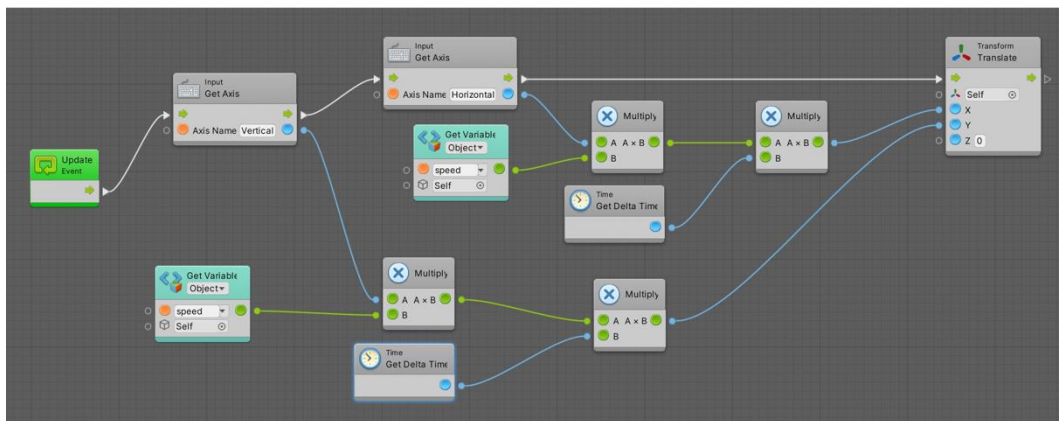


Figure 3. Example of Visual Scripting Using Bolt

Source: Screen captured from researcher's computer

## METHOD

This study observed 92 students who took Visual Programming courses. The respondents are a combination of students from 3 semesters from August 2021 to the present (1<sup>st</sup>, 2<sup>nd</sup> study period of 2021/2022 and 1<sup>st</sup> study period of 2022/2023). The data collection technique used observation during programming practice and limited interviews. The researcher did not observe all the material every week. The observation process is carried out on only a few materials based on three levels of material difficulty, namely easy, medium, and complex.

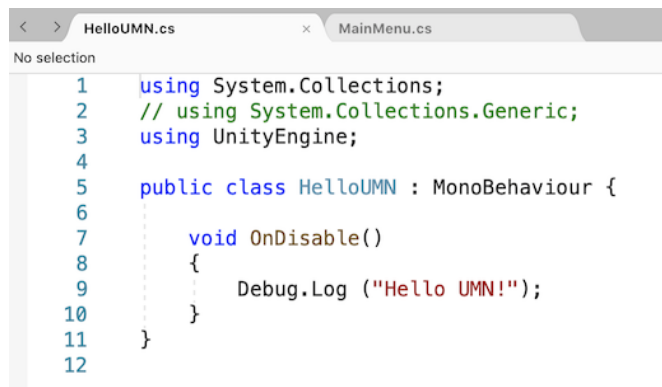
Basically, the observation was conducted by comparing the process - results of doing practical assignments using Bolt and the C# programming language. The observation activities were carried out as follows:

1. Observing students when compiling each C# coding and visual scripting command/ instruction on an assignment.
2. Observing students in understanding a programming logic on an interaction concept.
3. Observing students when developing an interactivity system

## OBSERVATION & DISCUSSION

### 1. Compiling C# Codes and Bolt Visual Scripting

This section describes the observation to students when they were compiling the C# codes and visual scripting. Here is example of some observation process, the researcher will not explain all of observation activities. First observation, the researcher gave task to the students for doing coding using C# programming language. The simplest instruction was print out “Hello UMN” to the UI of UNITY (as shown in Figure 4) and then the researcher did sample from some of the students to explain each line/row of the codes.

A screenshot of a code editor window with two tabs: 'HelloUMN.cs' and 'MainMenu.cs'. The 'HelloUMN.cs' tab is active and shows the following C# code:

```
1 using System.Collections;
2 // using System.Collections.Generic;
3 using UnityEngine;
4
5 public class HelloUMN : MonoBehaviour {
6
7     void OnDisable()
8     {
9         Debug.Log ("Hello UMN!");
10    }
11 }
12
```

Figure 4. Code of Print Out “Hello UMN!”  
Source: Screen captured from researcher’s computer

Most of them understood the code, although the instruction of the code did not mention about print out. The students understood that line 7 to 10 are command of print out the message of “Hello UMN!”. They agreed that there is easy to guess by using clue of “Hello UMN!”. The researcher points out that they were not really understand the codes, however they know the logic about print out the text.

The second experiment, the students have written codes of two instructions which are play game by loading the game scene and quitting the game (as we seen the codes in Figure 5). Most of the students faced difficulties to write the codes. They are not familiar with codes, even they don't know how to call the command code that they mean and the function of each code.

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.SceneManagement;
5
6 public class MainMenu : MonoBehaviour
7 {
8     public void PlayGame ()
9     {
10         SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 1);
11     }
12
13     public void QuitGame()
14     {
15         Debug.Log("Quit!");
16         Application.Quit();
17     }
18 }
19
```

Figure 5. Code of Play and Quit Game  
Source: Screen captured from researcher’s computer

Next step, I as researcher gave instruction using Bolt as visual scripting (as we see at Figure 6). The basic command is same with the C# programming above. There are loading the game scene and quitting the game.

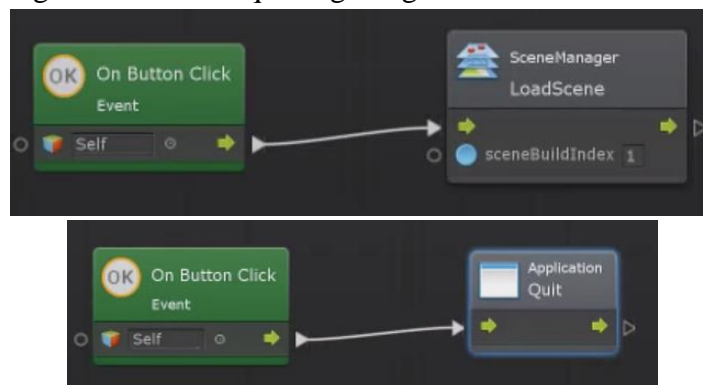


Figure 6. Bolt Visual Scripting of Play and Quit Game  
Source: Screen captured from researcher’s computer

Generally, after the students tried Bolt, most of them felt easier to understand the logic of programming. They have aided by the visual and human naming programming of Bolt visual scripting. Ray (2017) and Chao (2016) state that VPLs helps to visualize the programming logic. Coding by using visual scripting is more comprehensible rather than writing set of lines of C# codes. It also simplifies them to using scripting command by calling visualized “English-like” code naming.

## 2. Understanding Programming Logic on Interaction Concept and Developing Interactivity System

This observation section explains general information about the use Bolt for implement the students’ idea by developing their interactive media. As we see in Table 1, generally, the students face difficulty on using C# programming language. Scripting a set of C# codes is challenging for them who did not have IT background knowledge, especially while writing textual instructions.

**Table 1. Summary of Programming Understanding**

<b>Activities</b>	<b>Understand</b>	<b>Neutral</b>	<b>Don't understand</b>
C# Code	<b>10</b>	<b>15</b>	<b>67</b>
Using C#	<b>12</b>	<b>13</b>	<b>67</b>
Programming Logic of C#	<b>12</b>	<b>13</b>	<b>67</b>
C# Implementation	<b>5</b>	<b>33</b>	<b>54</b>
Bolt VisScript	<b>78</b>	<b>7</b>	<b>7</b>
Using Bolt	<b>82</b>	<b>3</b>	<b>7</b>
Programming Logic of Bolt	<b>85</b>	<b>0</b>	<b>7</b>
Bolt Implementation	<b>85</b>	<b>0</b>	<b>7</b>

However, there are still have some students (about 10-12 persons) who understand dan can use C# programming language. In the interview, they stated that they have previous programming experience. They learned programming at senior high school or programming training center. Broll et al. (2017) also state that this circumstance was happen in their study of visual programming environment for learning distributed programming. Consequently, the well-known programming students were much faster to understand the logic of programming and wrote the C# codes.

The use of Bolt improves understanding of programming logic. According to Table 1, the number of students (more than 75 persons) who understand programming logic is increasing sharply comparing to students who familiar with C# programming language. They felt easier using Bolt rather than wrote C# codes. Using Bolt feels easy to use because it looks like the hierarchy and order of data communication, namely variables and values given. Moreover, Bolt uses

instruction blocks and lines that are interconnected between instructions. It is also stated by Renger (2022) that “A visual scripting language is superior to text when it comes to expressing sequences of things, because you can lay things out spatially in a way that you can't in text-driven programming languages. The spatial dimension you have is one dimension vertically and if you have multiple flows somehow then you will interpret these lines with each other”. In their study, Kuhail et al. (2021) and Chao (2016) mention that VPLs was helping students learn how to program.

Basically, understanding programming logic is the goal of the Visual Programming course. As a researcher, I focus on this goal. Researchers do not expect students to be proficient in programming, but they understand programming logic. The point is that they understand the programming process so that it is easier to work with programmers one day. In this observation process, students' understanding is also seen when implementing visual scripting in the development of interactive media that they are doing. They don't spend much time programming, but they can focus more on creating 2D and 3D assets.

## CONCLUSION

This paper presents the observation of the use of Bolt visual scripting in Visual Programming course at Visual Communication Design Study Program, Universitas Multimedia Nusantara. In summary, the use of Bolt has positive impact to the students. The benefit of Bolt is easy to learn and help students to understanding programming logic. In addition, the Bolt helps students learn how to program. Furthermore, learning methods are needed that can be developed to increase the effectiveness of understanding programming logic in the future.

## REFERENCES

- Broll, B., Lédeczi, Á., Völgyesi, P., Sallai, J., Maróti, M., & Carrillo, A. (2017). A visual programming environment for learning distributed programming. *Proceedings of the Conference on Integrating Technology into Computer Science Education, ITiCSE*, 81–86. <https://doi.org/10.1145/3017680.3017741>
- Chao, P. Y. (2016). Exploring students' computational practice, design and performance of problem-solving through a visual programming environment. *Computers and Education*, 95, 202–215. <https://doi.org/10.1016/j.compedu.2016.01.010>
- Chu, E., & Zaman, L. (2021). Exploring alternatives with Unreal Engine's Blueprints Visual Scripting System. *Entertainment Computing*, 36(February 2020), 100388. <https://doi.org/10.1016/j.entcom.2020.100388>
- Desain Komunikasi Visual*. (2022). Universitas Multimedia Nusantara.



- <https://www.umn.ac.id/desain-komunikasi-visual/>
- Knutsen, K. Í. (2021). *Visual Scripting in Game Development* (Issue May). Metropolia University of Applied Sciences.
- Kuhail, M. A., Farooq, S., Hammad, R., & Bahja, M. (2021). Characterizing Visual Programming Approaches for End-User Developers: A Systematic Review. *IEEE Access*, 9, 14181–14202.  
<https://doi.org/10.1109/ACCESS.2021.3051043>
- Martyastiadi, Y. S. (2022). *Rencana Program Dan Kegiatan Pembelajaran Semester (RPKPS)*.  
[https://elearning.umn.ac.id/pluginfile.php/24606/mod\\_resource/content/0/RPKPS ID 490 Visual Programming.pdf](https://elearning.umn.ac.id/pluginfile.php/24606/mod_resource/content/0/RPKPS_ID_490_Visual_Programming.pdf)
- McGuffin, M. J., & Fuhrman, C. P. (2020). Categories and Completeness of Visual Programming and Direct Manipulation. *ACM International Conference Proceeding Series*. <https://doi.org/10.1145/3399715.3399821>
- Myers, B. A. (1990). Taxonomies of visual programming and program visualization. *Journal of Visual Languages and Computing*, 1(1), 97–123.  
[https://doi.org/10.1016/S1045-926X\(05\)80036-9](https://doi.org/10.1016/S1045-926X(05)80036-9)
- Okita, A. (2020). *Learning C# programming with Unity 3D*. CRC press.
- Ray, P. P. (2017). A Survey on Visual Programming Languages in Internet of Things. *Scientific Programming*, 2017. <https://doi.org/10.1155/2017/1231430>
- Renger, S. (2022). *Investigation into the criteria of embeddability of visual scripting languages within the domain of game development*. (Issue June) [Breda University of Applied Sciences].  
<https://doi.org/10.13140/RG.2.2.11976.39686>
- Unity Technologies. (2020). *Visual Scripting with Bolt*. Unity Technologies.  
<https://docs.unity3d.com/bolt/1.4/manual/index.html>