# VISUAL PROGRAMMING – THE BENEFITS OF TECHNOLOGY IN THE VIDEO GAME CREATION FOR NON-PROGRAMMERS, ITS CHALLENGES, MYTHS, AND WHY IT SHOULD BE STANDARDIZED

**David CODJEBAȘ**

*Department of Industrial and Product Design, DJ-221 group, Faculty of Design,*
*Technical University of Moldova, Chisinau, Republic of Moldova*

Corresponding author: David Codjebaș, david.codjebas@dip.utm.md

Tutor/coordinator: **Liliana PROZOR BARBALAT**, University Assistant

***Abstract.*** *This paper explores the realm of visual programming in the creation of video games, shedding light on its benefits for non-programmers and designers, addressing challenges, dispelling myths, and advocating for its standardization. The article delves into the significance of Visual Programming Languages (VPLs) and their impact on the digital product market, particularly within the context of video game development. After providing a brief overview of what Visual Programming Language entails, its historical roots, the evolution of programming as a whole, it's rising popularity among game and level designers, and its applications withing the development of both big and indie game projects, it will further explore the role of VPLs in education and the development of logical thinking and nurturing interest in programming among children. Additionally, the article will analyze the success of Unreal Engine's Blueprint system, and compare its performance against traditional code, highlighting performance concerns and projecting a future of technology adoption for the game development industry.*

***Keywords:*** *blueprint system, designer tools, digital product development, education, unreal engine, unity engine, visual scripting.*

## Introduction

The paper will analyze what a Visual Programming Language is, its emergence and history, where it is used, and its benefits for individuals taking their first steps in the realm of creating digital products.

Programming Language (VPL) or Block-based Coding is a programming language that allows the user to create programs by visually manipulating program elements, instead of manually writing traditional code [1]. The graphical aspect of the language allows symbols and spatial arrangement of text to be used as elements of syntax or secondary notation. For example, most VPLs use and are based on the "boxes and arrows" concept (Fig. 1), where the boxes are treated as code blocks, and the arrows represent the logic connecting these blocks.

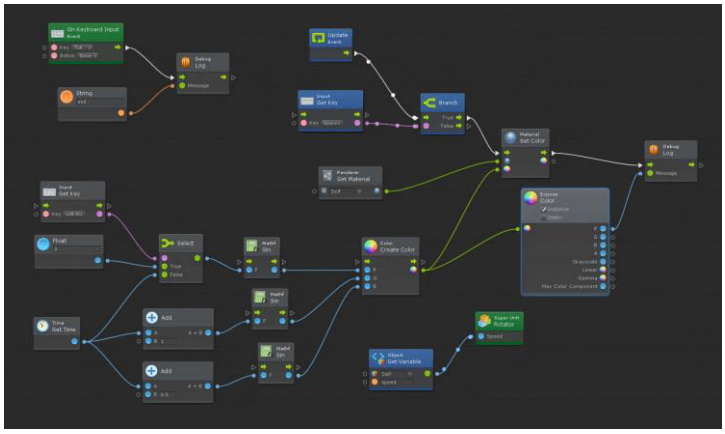

**Figure 1. Example of Visual Programming in Unity**

This programming method is extremely beneficial for individuals inclined towards graphic and spatial thinking. It enables beginners to quickly grasp the essence of programming and enhances the efficiency of prototyping and creating digital products, such as video games, movies, and the other interactive digital media [2]. Standardizing this technology will expand the space for enthusiasts interested in developing these products.

**History and development of programming**

Programming is the composition of a sequence of instructions that enable a computer to solve a given problem, starting from initial data [3].

The history of programming takes us back to the 17th century when J. Neper created Napier's bone logarithm, which was used for logarithmic calculations and significantly shortened the time required for computer calculations [4]. The following years people saw active development in the field of computer programming, although none of them could truly be called automatic ones.

In the 1840s, Ada Lovelace created the first computer program based on Charles Babbage's idea of punched cards. These cards essentially represented surfaces with holes placed in a specific order, allowing the computer to decipher the execution algorithm [5]. The continuous development of digital systems eventually led to the creation of electronic tabulation by H. Hollerith, which allowed the machines read the inputted data [6]. Following the success of this technology, he continued to expand the digital space and established the Tabulating Company Machine, which eventually became IBM (International Business Machines) [7]. IBM is directly responsible for the advanced computer technologies we are using today. For example, Microsoft gained widespread popularity because IBM wanted one of the first versions of Windows (Windows 85) to be installed on IBM-produced computers [8].

**Programming in the current day**

In today's world, programming is carried out through various Integrated Development Environments (IDEs) and various programming languages [9].

Microsoft is considered the leader in the market of digital creation products, as it owns the most popular IDEs called Visual Studio Code and Visual Studio. These applications are used by more than 14,000,000 people daily [10] Fig 2. Additionally, the company has created 39 programming languages, with the most popular ones being C#, TypeScript, and PowerShell [11].

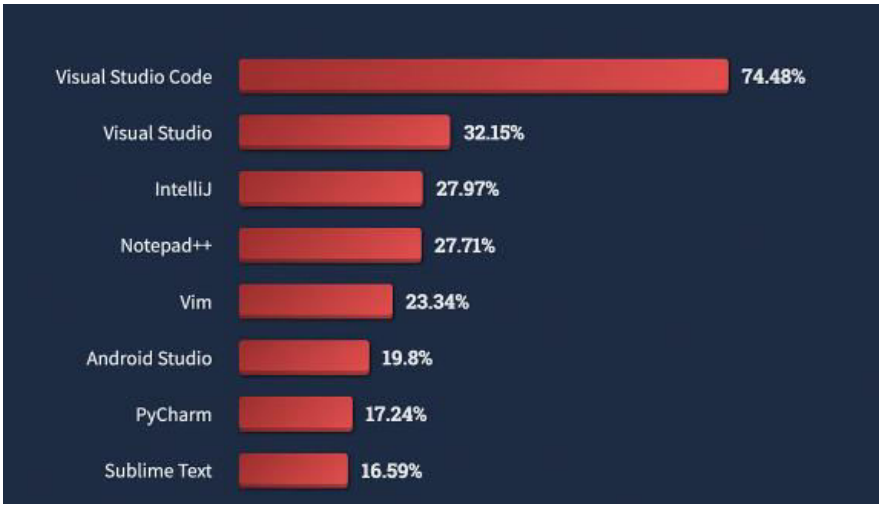


**Figure 2. The most popular Integrated Development Environments in 2021-2022**

**The importance of the C programming language family**

The C# language is a derivative of C++ [12], the latter being considered one of the most challenging programming languages but also the most flexible in terms of creative possibilities. C++ is based on the Object-Oriented Programming (OOP) paradigm, the idea of which is the

manipulation of the "object" concept through code and data [13]. All video games are created using an OOP language. Currently, C++ represents 55%, and C# represents 22% in the statistics of the most sought-after languages for companies involved in video game development Fig 3.

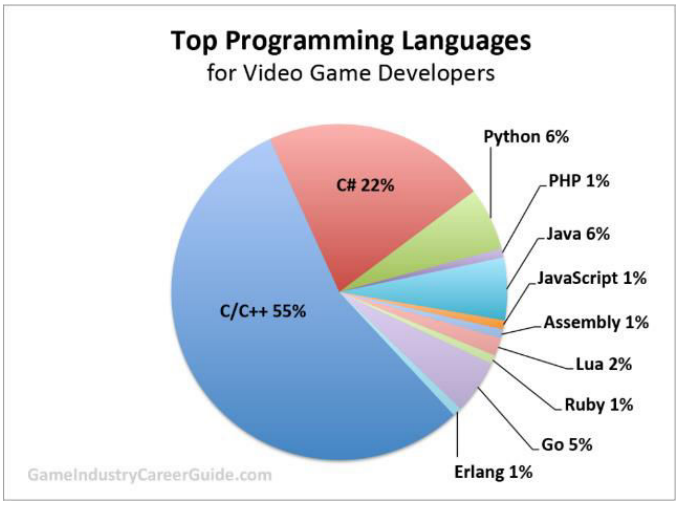


**Figure 3. The most popular programming languages used by game developers**

**The origin of the Visual Programing Languages**

Visual Programming (VPL - Visual Programming Language) is a method of computerized logic representation through graphical means rather than traditional written code [14]. This method has existed for several decades and is essential, much like today's languages, as an abstraction of computer language adapted for human perception [15].

Attempts to create a visual representation of code interconnections can be traced back to the 1960s-70s. Programmers tried to generate these diagrams through a system called Pygmalion and GRaIL (Graphical Input Language), which was very limited in its capabilities. It was only in the 1990s, when the graphical capabilities of computers were sufficiently advanced, that it became possible to create visualization systems that truly enhanced the application development process. This was facilitated by the Hypercard program developed by Apple in 1997 [16]. However, what truly revolutionized the field of visual programming was the first version of the Integrated Development Environment (IDE) called Visual Basic [17]. In this environment, the language named Visual Basic was combined with other relevant languages of the 90s, and, in combination with various graphic tools and code, enabled coding using blocks and arrows.

**Visual Programing as a tool for education**

In the year 2002, the Massachusetts Institute of Technology (MIT), specifically the MIT Media Laboratory, created a Visual Programming Language (VPL) called Scratch. It was designed as an educational tool with the goal of developing logical thinking and sparking interest in programming among children [18]. Scratch is a very good example of a Visual Language because it employs syntax in the form of blocks, each containing predefined logic, and these blocks can be directly connected to other similar blocks, creating a logical chain. Moldova introduced Scratch into the school education system in the academic year 2014-2015 as part of computer science lessons.

Due to the childish appearance of the application, visual programming was initially regarded by the public as something for beginners and was not taken seriously by companies. The reputation of Visual Programming Languages (VPLs) was further degraded by the fact that advanced logic created a chaos that was more challenging to comprehend than traditionally written code, as shown in Fig 4.

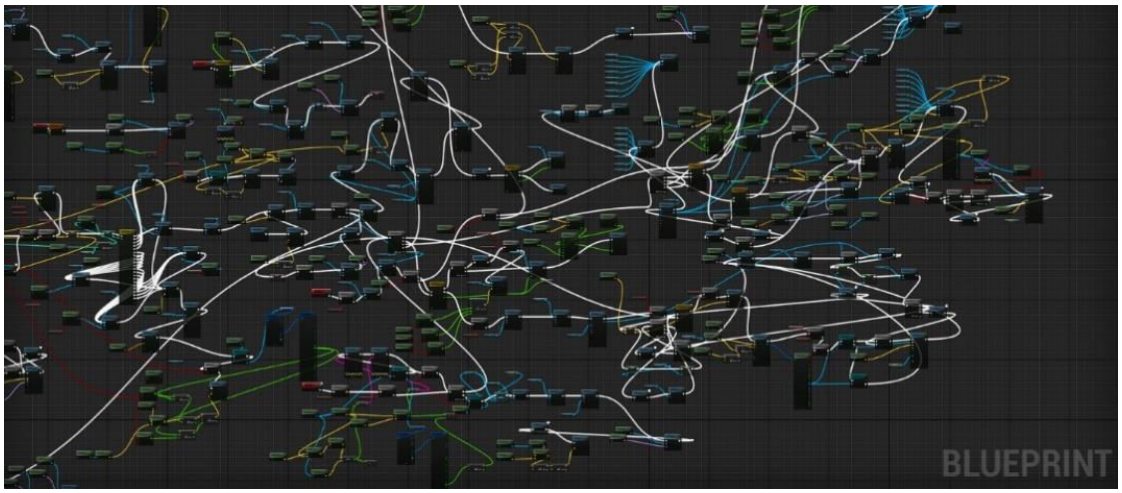


**Figure 4. Example of a disorganized visual scripting code**

**Visual Programing in the industry of Video Games**

Video games are created through a "Game Engine." During the 1990s-2000s, these engines were developed by companies for their own projects, a process that took on average from one to four years. Nowadays, this practice is less common, and companies like Epic Games, Unity Technologies, CryTech, and others offer their engines, which are mostly free and used for both amateur and AAA-class projects.

These engines are very similar to IDEs in that they provide a creative space and often include built-in physics systems, optimized for different platforms, lighting calculation systems, and more. Additionally, each engine uses a different programming language.

The most popular example of an engine that offers visual programming is the Unreal Engine, created by Epic Games [19]. The foundation of the engine is the C++ language, and it supports traditional code written in this language. However, the engineers at Epic Games have created an optimized and advanced environment for comfortable visual programming, naming this system "Blueprint."

The Blueprint system uses the standard introduced with Visual Basic and allows you to connect and create "Events," "Functions," "Values," and more using blocks called "nodes" and arrows [20]. In comparison to the VPL models mentioned above, the Blueprint system allows for the creation of extremely complex and advanced systems. Its effectiveness has already been demonstrated in the product market through the game Fortnite, which brought the company $20 billion from 2018 to 2023 [21].

**Visual Programing does not compare to traditional code in terms of performance**

The argument put forth by individuals who prefer traditional programming is that, during the execution phase, visual programming is not fast enough. In this context, taking Unreal Engine as an example, the Blueprint system uses a Virtual Machine that simulates the given code and transcribes it into C++, requiring significant computational power. However, in everyday terms, computer components are advanced enough to perform this conversion so rapidly that it is imperceptible to the human eye. Moreover, using this system makes it easier for you to visualize, edit, and analyze your code, even while the application you are creating is running. This capability is not afforded by traditional code, which may even disrupt the progress made.

**Examples of transitioning to Visual Programing**

Another giant in the video game creation industry is Unity Technologies, which owns a Game Engine similarly named Unity. Approximately 62% of all games available on the digital market have been created with this engine (Fig 5). The programming language used in this engine

is C#, and until the year 2021, when the company introduced the Bolt system, enabling visual programming, which, at times, is even more optimized than traditional programming [22].

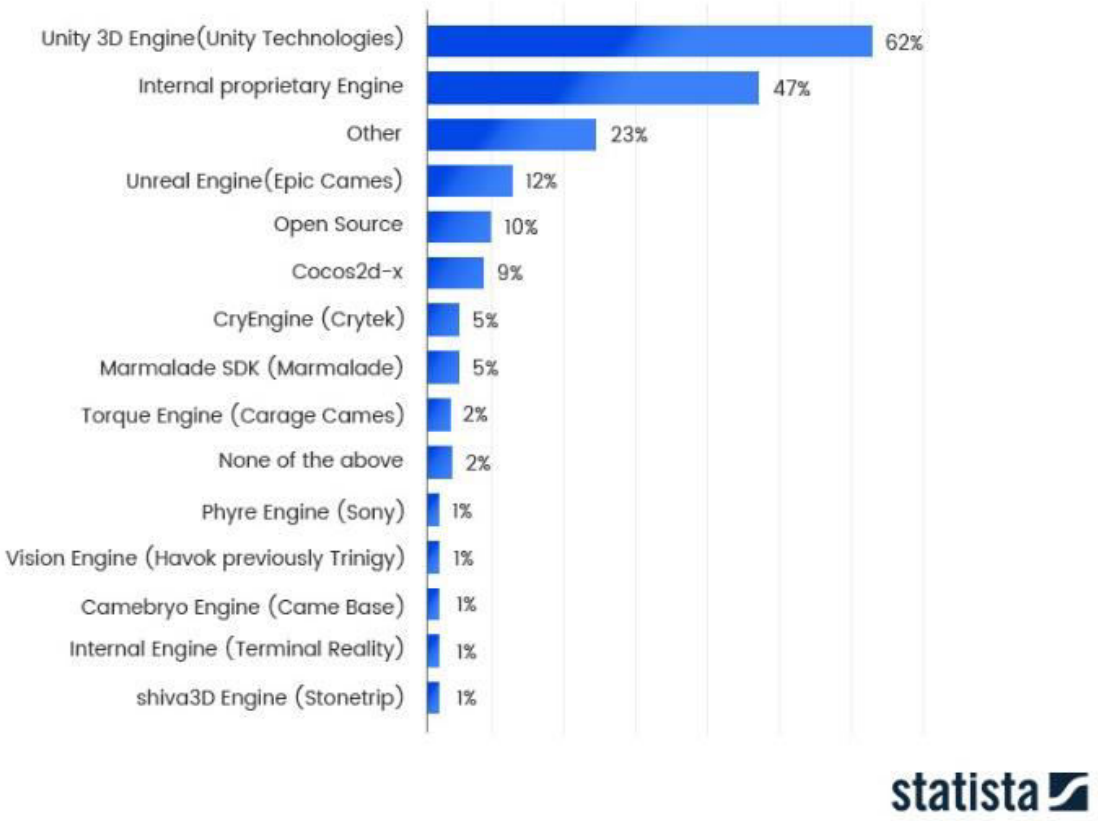


**Figure 5. Most popular game engines in 2023**

## Conclusions

In conclusion, we can deduce the following benefits of visual programming. It is very fast for prototyping; creating visual code allows you to quickly establish the logic for your concepts compared to traditional code writing. Rapid iteration allows you to swiftly edit the code, even while the application is running. Clear visualization: analyzing visual code directly shows you which portions are currently active and what data they are handling, which greatly aids in troubleshooting. Easily grasped by artists and designers: game development teams often include members without advanced programming knowledge. Therefore, through this visual method, we enable each team member to contribute to the project. Easy for children to understand: as demonstrated by Scratch, visual programming is engaging for children and can increasingly stimulate their participation in this digital world of creation.

## References

[1] *Ce este un limbaj de programare vizuală (vpl)? - definiție din techopedia - Dezvoltare 2023.* (2023). Icy Science. [online] [accesat 06.02.2024] Disponibil: https://ro.theastrologypage.com/visual-programming-language

[2] Technologies, U. (n.d.). *Unity visual scripting*. Unity. [online] [accesat 06.05.2024] Disponibil: https://unity.com/features/unity-visual-scripting#:~:text=Visual%20scripting%20in%20Unity%20helps,for%20faster%20prototyping%20and%20iteration.

[3] *dexonline*.(n.d.). [online] [accesat06.05.2024] Disponibil: https://dexonline.ro/definitie/programare

[4] BUZATU, A. (2011, February 21). *John Napier şi inventarea logaritmilor.* [online] [accesat 06.05.2024] Disponibil: https://www.scientia.ro/stiinta-la-minut/48-scurta-istorie-descoperiri-stiintifice/1361-john-napier-si-inventarea-logaritmilor.html

[5] POPA, C. (2022, October 10). *Cine a fost Ada Lovelace, femeia care a inventat programarea?* Radio Impuls. [online] [accesat 06.02.2024] Disponibil: https://www.radioimpuls.ro/cine-a-fost-ada-lovelace-femeia-care-a-inventat-programarea-20258412

[6] *Cine este Herman Hollerith?* - definiție din techopedia - Hardware 2023. (2023). Icy Science. [online] [accesat 06.02.2024] Disponibil: https://ro.theastrologypage.com/herman-hollerith

[7] *Istoria IBM - Profilul unei fabrici de calculatoare Giant*. (n.d.). [online] [accesat 06.05.2024] Disponibil: https://ro.eferrit.com/istoric-ibm/

[8] *Microsoft is born - Stories*. (2020, December 7). Stories. [online] [accesat 06.02.2024] Disponibil: https://news.microsoft.com/announcement/microsoft-is-born/#:~:text=Inspired%20by%20the%20January%20cover,8800%2C%20an%20early%20personal%20computer

[9] Softlead. (2016). *Ce sunt mediile de dezvoltare (IDE)? Cum le folosim pentru a crea aplicații software de succes? softlead.ro*. [online] [accesat 06.02.2024] Disponibil: https://softlead.ro/noutati-it-c/ce-sunt-mediile-de-dezvoltare--ide--cum-le-folosim-pentru-a-crea-aplicatii-software-de-succes.html

[10] TUNG, L. (2021, February 1*). Visual Studio Code: How Microsoft's "any OS, any programming language, any software" plan is paying off. ZDNET*. [online] [accesat 06.02.2024] Disponibil: https://www.zdnet.com/article/visual-studio-code-how-microsofts-any-os-any-programming-language-any-software-plan-is-paying-off/

[11] *.NET programming languages | C#, F#, and Visual Basic*. (n.d.). [online] Microsoft. [accesat 06.02.2024] Disponibil: https://dotnet.microsoft.com/en-us/languages

[12] *What is C# Programming? A Beginner's Guide | Pluralsight*. [online] (2022, November 22). [accesat 06.02.2024] Disponibil: https://www.pluralsight.com/blog/software-development/everything-you-need-to-know-about-c-#:~:text=When%20was%20C%23%20created%3F,a%20history%20for%20popular%20creations

[13] NASTASE, R. (2022). *Programare Orientata pe Obiecte: ce este si ce rol are? RamonNastase.ro*. [online] [accesat 06.02.2024] Disponibil: https://ramonnastase.ro/blog/programare-orientata-pe-obiecte-poo/

[14] *Ce este un limbaj de programare vizuală (vpl)?* - definiție din techopedia - Dezvoltare 2023. (2023). Icy Science. [online] [accesat 06.02.2024] Disponibil: https://ro.theastrologypage.com/visual-programming-language

[15] CHEN, V. (2022). *An Intro Guide to Visual Programming* (2022). *Bubble Blog | What You Need to Know About Building With No-code*. [online] [accesat 06.05.2024] Disponibil: https://bubble.io/blog/visual-programming/#:~:text=Early%20visual%20programming%20involved%20lots,GRaIL%20(Graphical%20Input%20Language)

[16] SANDE, S. (2020). *Retro Apple: How HyperCard Changed the World One Stack at a Time. Rocket Yard*. [online] [accesat 06.02.2024] Disponibil: https://eshop.macsales.com/blog/64930-hypercard-one-stack-at-a-time/

[17] Contributor, T. (2019). *Visual Basic (VB). WhatIs.com*. [online] [accesat 06.02.2024] Disponibil: https://www.techtarget.com/whatis/definition/Visual-Basic-VB#:~:text=Visual%20Basic%20(VB)%20is%20an,defining%20their%20behavior%20and%20appearance

[18] *Our Story — Scratch Foundation*. (n.d.). Scratch Foundation. [online] [accesat 06.02.2024] Disponibil: https://www.scratchfoundation.org/our-story

[19] *Unreal Engine | The most powerful real-time 3D creation tool*. (n.d.). Unreal Engine. [online] [accesat 06.02.2024] Disponibil: https://www.unrealengine.com/en-US

[20] *Blueprint Overview*. (n.d.). Unreal Engine 4.27 Documentation. [online] [accesat 06.02.2024] Disponibil: https://docs.unrealengine.com/4.27/en-US/ProgrammingAndScripting/Blueprints/Overview/#:~:text=A%20Blueprint%20Class%2C%20often%20shortened,assets%20in%20a%20content%20package

[21] HIGHTON, A. (2022, December 20). *How much money has Fortnite made? - Charlie INTEL*. Charlie INTEL. [online] [accesat 06.02.2024] Disponibil: https://www.charlieintel.com/how-much-money-has-fortnite-made/218252/#:~:text=How%20much%20money%20has%20Fortnite%20made%20in%202022%3F,has%20likely%20made%20in%202022

[22] TECHNOLOGIES, U. (n.d.-b). *Unity visual scripting*. Unity. [online] [accesat 06.02.2024] Disponibil: https://unity.com/features/unity-visual-scripting

[23] Figure 1. Valerie. (2022, October 27). Are Visual Programming Languages Worth Learning? - Dare To Be Better [online] [accesat 06.02.2024] https://medium.com/dare-to-be-better/can-you-learn-programming-with-visual-programming-languages-f6cf83c17c70

[24] Figure 2. Sim, E. (2022, June 25). What's hot in Enterprise IT/VC #295. *By Ed Sim*. [online] [accesat 06.02.2024] https://www.whatshotit.vc/p/whats-in-enterprise-itvc-295?utm_source=substack&utm_medium=email

[25] Figure 3. Bay, J. W., & Bay, J. W. (2023). The Best Programming Languages for Games (According to Employers). *Game Industry Career Guide*. [online] [accesat 06.02.2024] https://www.gameindustrycareerguide.com/best-programming-languages-for-games/

[26] Figure 4. PiratePaprika. (n.d.). *Reddit - Dive into anything*. [online] [accesat 06.02.2024] https://www.reddit.com/r/ProgrammerHumor/comments/11el53a/try_visual_programming_they_said_it_will_be_fun/

[27] Figure 5. Program-Ace. (2023, February 2). *Unity vs. Unreal: Choosing the Best Engine | Program-Ace*. [online] [accesat 06.02.2024] https://program-ace.com/blog/unity-vs-unreal/