Research on the Applicability of Makecode Arcade from the Perspective of Resource Allocation

Qingpeng Xu[1](✉) and Yingxin Liu[2]

[1] Normal College, Shenyang University, Shenyang, China
books-vi@outlook.com
[2] Graduate School, Dadong District, Shenyang University, No. 21, Wanghua South Street, Shenyang 110044, China

Abstract. Programming is an important information technology skill in K-12 education and a necessary means to promote the development of core literacy in the new curriculum standard of information technology. However, the effective implementation of programming teaching highly relies on software and hardware configuration. In the current situation of imbalanced distribution of educational resources in China, it is particularly necessary to fully consider the issue of resource allocation in programming education. This study takes exploring low configuration requirements and high compatibility programming tools as the starting point, focusing on the low-cost characteristics and compatibility advantages of Makecode Arcade, providing valuable reference for programming tool selection in resource disadvantaged schools.

Keywords: programming tools · low-cost · Makecode Arcade · BBPE

1 Introduction

In March 2022, the Ministry of Education of China officially issued the "Information Technology Curriculum Standards for Compulsory Education (2022 Edition)" (hereinafter referred to as the "Curriculum Standards"), which established the independent status of the information technology discipline and put forward new requirements for curriculum objectives and content. In the Curriculum Standards, information awareness, computational thinking, digital learning and innovation, and information social responsibility are included in the core competencies of the discipline, closely integrated with the development of the information society, and more in line with the search for information talents in Chinese society.

Programming, as a fundamental skill to enhance core competencies, has also been put forward with higher requirements. In the Curriculum Standards, programming learning is no longer limited to traditional computer coding, but has expanded to more scenarios such as drone performances and small-scale system simulations. This means that schools need to invest more resources to meet the requirements of the Curriculum Standards.

D. Kumar et al. (Eds.): IEIT 2023, AHSSEH 10, pp. 741–747, 2023.
https://doi.org/10.2991/978-94-6463-230-9_88

However, for a long time, a value orientation of "urban center" has been formed in China that ignores regional and urban-rural differences [1], and there is a siphon effect of "advantageous schools" [2], leading to the concentration of high-quality educational resources towards "demonstration schools" and "key schools" [3]. Therefore, under the existing mechanism of education investment distribution, the current situation of imbalanced distribution of educational resources between regions, urban-rural areas, and schools has emerged, for information technology disciplines that heavily rely on educational resource investment, this situation greatly hinders the development of the discipline.

Based on the above factors, more effective measures should be taken to alleviate the current problems while further improving the mechanism for coordinating education funds. To ensure the effective implementation of programming teaching in resource disadvantaged schools, it is particularly important to choose programming tools with low configuration requirements and high compatibility. This article takes this as the starting point and focuses on the low-cost characteristics and compatibility advantages of Makecode Arcade, providing valuable reference for the selection of programming tools in resource disadvantaged schools.

2 Block-Based Programming Environment (BBPE)

2.1 Characteristics of BBPE

Currently, programming learning is widely applied in STEM/STEAM education and the intervention process of computational thinking, and has achieved many positive results. It is worth noting that "has low threshold and high ceiling" is one of the guiding principles when creating programming environments for children. It essentially means that a student can produce a working game quickly (low threshold), and a real game that is playable and exhibits sophisticated behavior, e.g., complex AI (high ceiling) [4]. A programming environment for dragging graphical code blocks was born under this demand. Thomas W use the term Block-Based Programming Environment (BBPE) to refer to those environments that allow users to construct and execute computer programs by composing atomic blocks of code together to produce program structure [5].

Compared to programming languages based on text commands, in addition to their block interfaces, many BBPEs share the following characteristics as well [5]:

- They target novice programmers, often younger children during primary or secondary education.
- Their programs reflect the syntax and structure of existing programming languages.
- They situate programming in a multi-media context, with a focus on cultural relevance. Users can integrate art, music and interactivity into their projects, leading to the creation of games, stories and apps.

2.2 Mainstream BBPE

In the past decade or so, different types of BBPE products have flourished and developed, this comes from the advantage that they provide a programming environment that is more

in line with children's cognitive development. Therefore, various types of BBPEs are widely used in programming learning in the K-12 stage and have played an extremely significant positive role in computational thinking intervention and STEM/STEAM education.

Scratch is one of the most famous BBPE, developed by researchers at the MIT in 2007.As one of the earliest BBPEs, Scratch has complete and powerful functions, and has a large number of program development resources on the Internet. It is favored by students and teachers around the world and is widely used in children's programming teaching. MIT App Inventor [6], which allows users to design and program Android apps in a web application. It has been evaluated in K-12 classrooms and summer camps, suggesting it is a powerful, motivational and accessible tool [7], which can serve as a bridge to textual coding in Java [8]. In addition, Tinkercad, Alice, and Kodu each have their own advantages and have played a positive role in many programming teaching practices.

The earliest BBPE in Chinese Mainland is ArduBlock developed by Xinchejian in Shanghai. It runs based on Arduino. The way of implementing programming based on drag and drop blocks is a prominent feature of ArduBlock witch is different from the Arduino text programming environment. Kittenblock is a BBPE developed by Shenzhen Xiaomiao Technology based on Scratch 3.0. As a parallel replacement for Scratch in China, it is consistent in interface design and functionality with Scratch.Kittenblock is based on web programs and runs for free with a large number of users. In addition, Linkboy, Huibiancheng, Mind +, and others developed by Chinese companies or teams also have a certain influence.

In summary, a series of BBPE represented by Scratch and App Inventor have achieved positive results in various programming teaching practices. In many studies, BBPE has been proven to be positive and effective in promoting students' problem-solving through programming and the development of computational thinking.

2.3 Application Issues of BBPE Under the Influence of Software and Hardware Environmental Factors in Chinese Mainland

A wide variety of BBPE have their own advantages and pertinence. However, the application of BBPE is restricted by factors such as school network access and digital equipment configuration in Chinese Mainland. Especially for schools with limited resources, there are common problems such as low hardware configuration performance, outdated computer operating system versions, and a lack of professional software. These schools often do not have the digital equipment conditions to use BBPE for programming teaching.

For example, Scratch's web server cannot be accessed normally in Chinese Mainland. And, although Scratch provides multiple operating system clients, it does not support old devices and systems enough. These problems limit Scratch's applications. As a localized replacement product for Scratch, Kitenblock has not been specifically optimized for mobile devices, resulting in a poor experience on tablets or phones. The server of App Inventor cannot be accessed normally in Chinese Mainland. In addition, App Inventor is only for the development of Android applications, and needs to rely on virtual machines or Android physical devices. Schools lacking corresponding software and hardware are difficult to use this platform for programming teaching. Similarly, Tinkercad can run

entirely on the web, but it also lacks optimization for mobile terminals, and its servers in Chinese Mainland are often inaccessible.

3 Makecode Arcade

Microsoft MakeCode is a free web-based open source platform developed by Microsoft Corporation, aimed at creating an attractive computer science learning experience and laying the foundation for practical programming. It includes two BBPEs: MakeCode for micro: bit and MakeCode Arcade. Among them, Arcade is a web-based programming environment designed to create retro style electronic games for gaming handheld devices. It is easy to use and has extremely low hardware costs. Arcade supports multiple microcontrollers, including a 2D game engine and easy-to-use sprite and music editor. It includes three main functional modules: simulator, block editor, and JavaScript editor.

Interactive simulators provide students with real-time feedback on the running status of their programs, making code testing and debugging easy and feasible. Block editors are designed for first-time code writing students, who can drag and drop colored command modules onto the workspace to visually build programs. JavaScript editors are aimed at students with a certain programming foundation, who can use a fully functional JavaScript editor that provides code snippets, tool tips, and error detection tools. In addition, Arcade is not an editor created for specific hardware, but rather a hardware specification has been released, and many companies are building devices compatible with Arcade [9].

4 Advantages of Makecode Arcade

4.1 Extremely Low Usage Cost

In a small amount of research on Arcade, some researchers have noticed an important feature of Arcade - low cost. Moskal et al. strongly emphasize the low cost characteristics of Arcade, pointing out that there is no game creation platform that has both the same low-barrier to entry and low-cost hardware as the BBC micro: bit. Qualitative and quantitative evaluation demonstrates that Arcade enables a modern and fully web-based programming experience for low-cost microcontroller-based gaming handhelds [10]. Specifically, Arcade's low cost is mainly reflected in three aspects:

- Arcade is completely free and open source, without limiting the use of any features;
- Running on the web greatly reduces system environment limitations and hardware performance requirements;
- Allowing third-party innovation and the creation of new game controllers seamlessly integrated with the Arcade software stack, the acquisition cost of expansion devices is lower, and Arcade can simulate or debug programs on different devices entirely through its built-in emulator, so it is not necessary to purchase expansion devices.

4.2 Excellent Cross Platform Compatibility

Due to the diversity of computer hardware configurations and systems in different schools, there may be varying degrees of obstacles when installing applications or driving devices, and the compilation of programs that rely on internet services may also be affected by network conditions. Arcade is fully based on web program implementation functionality, allowing it to run on different old system environments and outdated hardware devices, with strong cross platform compatibility. It can run on iOS devices based on A9 processors, low configuration Android devices, and even smart TVs in the living room. Moreover, Once the web app has loaded into browser, it remains resident and operational (even if the browser is closed and reopened, or network connectivity is lost) [9]. This is extremely friendly for schools with poor internet conditions.

4.3 Convenient Multi Device Collaboration and Sharing

Arcade can achieve multi end synchronization of data by logging into a Microsoft account, and can complete the same project across platforms on different devices or system environments under the same account. Users can share their programs in a variety of ways. For example, MakeCode can store the user's encrypted program in the cloud and generate a URL (that contains the decryption key) to share with other users. Arcade's convenient multi device collaboration and sharing function provides strong support for hybrid learning, group cooperative learning and other ways. Especially for the differences in electronic hardware configuration between different families, Arcade highlights its strong compatibility. Students only need a device that supports browser in their home programming learning, such as smart TV in the living room or low configuration Android device, Therefore, there is no need to incur additional costs to purchase specific types of hardware.

5 Disadvantages of Makecode Arcade

The MakeCode Arcade interface, like Scratch, has the ability to edit sprite characters. However, compared to Scratch, it is more limited in terms of what can be created [11]. Compared to other BBPEs, Arcade does not have outstanding advantages in terms of functionality. However, from a cost perspective alone, it can make it easier to promote programming learning, especially in developing countries with imbalanced educational investment. Although some researchers have proposed from the perspective of computational thinking intervention that the lack of equipment in developing countries can be addressed through the use of "unplugged" activities that are independent of digital infrastructure [12], programming teaching that is completely detached from digital devices lacks practical significance. Therefore, the low-cost characteristics of Arcade have practical value in developing countries, and it is worth trying to bridge the gap in programming teaching between regions.

6 Practical Research on Using Arcade for Programming Teaching

Currently, there is still relatively little research on using Arcade for programming teaching. In a study on primary school students' programming learning using Arcade, researchers concluded that Arcade is an interesting environment for teaching programming, and the environment is suitable for teaching programming - it contains basic programming concepts [13]. Begel et al. designed, developed, and ran a 13-day, remote video game coding camp for incoming college first-year students with ASD. Students used the MakeCode Arcade development environment to build their games and Zoom to remotely collaborate with their teammates. In summative interviews, students reported improved programming skills, increased confidence in communication, and better experiences working with others [14]. Although a small amount of practical research on Arcade is not yet sufficient to confirm its positive role in programming learning, Acrade possesses the general characteristics of BBPE. Based on the results of previous practical research on various types of BBPE, the effectiveness of Acrade in programming learning seems to be taken for granted.

7 Conclusion

The characteristics and advantages of Arcade are significant. It is completely free, open source, and has complete simulator functions, which makes the purchase of third-party hardware unnecessary. Therefore, schools using Arcade for programming teaching do not need to invest more funds in hardware. Arcade's fully web-based and offline running services have extremely low requirements for system environment and hardware performance, which gives it excellent cross platform compatibility. This largely ensures that schools with outdated equipment and poor network conditions can also carry out programming teaching. In the process of programming learning at home, students are also less affected by differences in the software and hardware configurations of electronic devices between families. Arcade's convenient network collaboration and sharing functions enable it to play an active role in hybrid teaching and cooperative learning. In a word, although Arcade is not perfect, its characteristics and advantages can alleviate the programming teaching problems under the current unbalanced distribution of educational resources in Chinese Mainland to a certain extent, and have high practical value for promoting the development of information technology disciplines and education fairness.

Acknowledgments. This work was supported by the 2023 Liaoning Provincial Science Public Welfare Research Fund (soft science Research Program) 2023H4/10700057.

References

1. Huang Jing. The Imbalance of Educational Resource Allocation and Its Impact on Social Equity [J]. *Theory and Contemporary*, 2009, 05:29-32

2. Yao Yong-qiang. The Study on the Transformation of Compulsory Education Equilibrium Development Mode [D]. Central China Normal University,2014.
3. Cao Meiqi. Reflection on the Practice of Basic Education Group Running Schools [J]. *Teaching & Administration*, 2018 (10): 9-12
4. Repenning A, Webb D, Ioannidou A. Scalable game design and the development of a checklist for getting computational thinking into public schools[C]//Proceedings of the 41st ACM technical symposium on Computer science education. 2010: 265–269.
5. Price T W, Barnes T. Comparing textual and block interfaces in a novice programming environment[C]//Proceedings of the eleventh annual international conference on international computing education research. 2015: 91–99.
6. S. Pokress and J. Veiga. MIT App Inventor: Enabling personal mobile computing. In *Workshop on Programming for Mobile and Touch*, 2013.
7. R. Morelli, T. de Lanerolle, and P. Lake. Can android app inventor bring computational thinking to k-12. In *ACM technical symposium on Computer science education (SIGCSE'11)*, 2011.
8. A. Wagner, J. Gray, J. Corley, and D. Wolber. Using app inventor in a K-12 summer camp. In *Proceeding of the 44th ACM technical symposium on Computer science education*, 2013.
9. Ball T, Chatra A, de Halleux P, et al. Microsoft MakeCode[J]. 2019.
10. Moskal M, Ball T, Chatra A, et al. Web-based Programming for Low-cost Gaming Handhelds[C]//The 16th International Conference on the Foundations of Digital Games (FDG) 2021.2021:1–12.
11. Chesterman M. Game Making and Coding Fluency in a Primary Computing Context[M]//*Teaching Coding in K-12 Schools: Research and Application. Cham: Springer International Publishing*, 2023: 171-187.
12. Israel-Fishelson R, Hershkovitz A. Studying interrelations of computational thinking and creativity: A scoping review (2011–2020)[J]. *Computers & Education*, 2022,176:104353.
13. Voštinár P. MakeCode Arcade: Interesting environment for programming 2D games[C]//2021 IEEE World Conference on Engineering Education (EDUNINE). IEEE, 2021: 1–6.
14. Begel A, Dominic J, Phillis C, et al. How a Remote Video Game Coding Camp Improved Autistic College Students' Self-Efficacy in Communication[C]//Proceedings of the 52nd ACM Technical Symposium on Computer Science Education. 2021: 142–148.