Journal of Advanced Academic Research and Studies 高等学术研究期刊
Vol. 1, No.3, March 2024 (pp. 55-62)
http://www.nlbaeai.org/

ISSN 3006-4007 (Print)
ISSN 3006-4015 (Online)

Strategy for Designing and Developing Preschool Education Software

Fangqi,Yi[a] **Sukh-Ochir Enkh-Amar**[b]

[a] Department of Inclusive Education, Shinhan University, South Korea
[b] International Exchange Office, Shinhan University, South Korea

Received 13 September 2023, Revised 10 December 2023, Accepted 1 March 2024

Abstract

Purpose – The purpose of this paper is to explore the strategies for the design and development of preschool education software, and to provide guidelines for the developers of preschool education software to produce more effective and interactive educational software.
Design/Methodology/Approach – Based on the development suitability of preschool education and the principle of combining education and entertainment, this paper analyzed the complete software development process, from needs analysis to testing and evaluation, and put forward the implementation strategy.
Findings – This paper studies the key principles and processes in the design and development of preschool education software, and how to implement the software through appropriate strategies, so as to improve the practical application value and education effect of educational software.
Research Implications – Research significance: This study provides guidance and reference for the design and development of preschool education software. It has significance in helping developers better understand the characteristics and needs of preschool education so as to design software products that are more in line with children's developmental characteristics and educational goals, promoting technological innovation and updating of teaching methods in the field of preschool education.

Keywords: Preschool education, Software design, Software development, Educational technology
JEL Classifications: L2,L8,O3

[a] First Author, E-mail: yifangqi335@gmail.com
[b] Second Author, E-mail: enkh7351@gmail.com

Ⅰ. Introduction

With technological advancements, the field of preschool education increasingly relies on digital tools to support teaching and learning activities. Designing and developing efficient preschool education software is paramount for fostering the cognitive and social skills development of young children. Effective software design not only aligns with fundamental educational principles but also considers the developmental characteristics and learning needs of young children. This thesis thoroughly discusses the key principles and processes involved in these design and development phases, along with implementing the software through suitable strategies to enhance their practical application value and educational effectiveness.

Ⅱ. Design Principles of Preschool Education Software

1. Principle of Developmental Appropriateness

The principle of developmental appropriateness, rooted in educational theory, underscores the necessity for software design to align with preschoolers' developmental stages and cognitive abilities. This principle mandates that software developers thoroughly grasp the physiological and psychological developmental traits of preschoolers, ensuring educational content and interactive design cater to their age-specific characteristics and learning needs. When implementing this principle, careful consideration is given to children's attention spans, cognitive capacities, and the appropriateness of sensory stimuli. For instance, educational software targeted at 3 to 5-year-olds should feature straightforward, intuitive user interfaces and avoid complex navigation pathways. Additionally, tasks and challenges should be designed to be sufficiently simple, enabling preschoolers to operate independently with minimal adult intervention. Moreover, the principle emphasizes the practical application of educational content, fostering comprehensive development through specific, relevant learning activities. Activities and games within the software should ignite children's curiosity and innate desire for exploration while providing an optimal level of cognitive challenge to support growth across language, mathematics, social, and emotional domains. Achieving this principle necessitates close collaboration between developers and educational experts to strike the right balance between entertainment and education, as depicted in Table 1.

Table 1. Special Principles of Preschool Education Software Design

	Principles of Preschool Education
General Principles	Principle of Respecting Children's Personality, Dignity, and Legal Rights. Principle of Promoting Comprehensive Children Development. Principle of Inclusivity with Attention to Individual Differences. Principle of Maximizing Children, Family, and Social Resources. Principle of Developmental Appropriateness
Special Principles	Principle of Integrating Care with Education Principle of Play-Based Activities Principle of Educational Activity and Activity Diversity Principle of Leveraging the Overall Educational Function of Daily Activities

2. Principle of Integrating Education with Entertainment

The principle of blending education with entertainment, often referred to as "edutainment," stands as a cornerstone when crafting preschool education software. This principle holds that presenting educational material in an entertaining format can significantly boost children's motivation to learn and engagement, fostering a natural and enjoyable learning experience. In practice, developers must devise captivating storylines and interactive games, framing educational goals as engaging activities rather than conventional learning. For instance, role-playing games can impart basic mathematical skills while virtual travel narratives can introduce children to and deepen their understanding of concepts in natural sciences. Furthermore, this principle underscores the importance of multisensory learning. By integrating visual, auditory, and tactile elements, software can effectively support children's cognitive development and skill acquisition. Animations, sound effects, and interactive feedback should be designed to captivate children's attention without overly distracting them, ensuring the seamless delivery of educational content. Successfully merging education with entertainment requires close collaboration among educational psychologists, curriculum developers, and game designers to ensure that the resulting software not only meets educational objectives but also offers significant entertainment value (Wang, wangfen, 2015).

Ⅲ. Development Process of Preschool Education Software

1. Requirement Analysis and Topic Selection

In the development of preschool education software, requirement analysis encompasses not only a thorough understanding of educational content but also an investigation into the psychological and behavioral traits of young children. During this phase, the development team collaborates with educational experts, psychologists, and practicing teachers to gather data through observations and interviews, among other methods, in order to analyze children's learning motivation, cognitive developmental stages, and learning preferences. Moreover, requirement analysis must consider regulatory requirements and technical feasibility to ensure that the software complies with relevant educational policies while also harnessing the latest technology to enhance teaching effectiveness. Identifying and defining key issues in early childhood education, such as language development, foundational mathematics, and social skills, is pivotal in this process, as it directly influences topic selection and subsequent design. Through systematic requirement analysis, the core functions and educational objectives of the software can be established, offering clear guidance for the development of educational content and learning activities.

2. Design of Teaching Content and Objectives

The teaching objectives should be precise and specific, closely linked to the developmental stage of the young children, while also ensuring that the teaching activities comprehensively cover the key learning areas recommended in the curriculum. The design team needs to develop specific content that aligns with the teaching objectives, including teaching materials, activities, interactive games, etc. This content should be both highly educational and engaging to spark children's interest in learning and enhance their sense of participation. During the development of teaching content, as outlined in Table 2, each teaching activity is clearly matched with its corresponding teaching objectives, ensuring that each component effectively meets the learning needs of the children. Table 1 illustrates the relationship between teaching content and objectives, enabling the development team to clearly grasp the design intent and expected outcomes of each teaching activity,

thereby ensuring that each activity effectively contributes to the achievement of teaching objectives. Furthermore, the design of teaching content should also incorporate assessment methods and feedback mechanisms, allowing teachers and parents to monitor children's learning progress and outcomes, thus facilitating necessary teaching adjustments and optimizations (Zhang, Binglin, 2014).

Table 2. Correspondence between Teaching Content and Objectives

Types of Teaching Activities	Specific Content	Teaching Objectives
Interactive Games	Number Matching Game	Improve children's numerical cognition abilities
Role-playing	Zoo Daily Management Simulation	Enhance children's understanding of animal species and sense of responsibility
Storytelling	Go Shopping: A Mathematical Adventure Story	Teach basic mathematical calculations and currency usage
Drawing Activities	Expressing Emotions through Color	Develop children's artistic expression and color recognition abilities
Music Interaction	Teaching Basic Knowledge of Music and Rhythm	Cultivate children's sense of musical rhythm and creativity

3. Interaction Design and User Experience Optimization

The process of interaction design encompasses several aspects, including user interface (UI) design, user experience (UX) design, usability testing, and adaptive adjustments. UI design involves attention to visual elements like color usage, icon design, and layout configuration to suit the visual and cognitive development levels of young children. For instance, vibrant colors and large icons can better capture children's attention, while simplified layouts aid in their comprehension and navigation. UX design delves into children's feelings and behavioral feedback during software usage, ensuring that the interaction logic aligns with their learning habits and psychological traits. This may involve streamlining complex commands via touchscreen operations and incorporating voice prompts and animated feedback to enhance learning motivation. Usability testing is a practical process for evaluating and refining user experience, typically through methods like prototype testing, user trials, and scenario simulations to identify and address design issues iteratively. These tests should be conducted in authentic usage environments with real user groups to gather precise usage data and feedback. Lastly, adaptive adjustments ensure a consistent user experience across different devices and operating systems, including responsive design and cross-platform support, catering to children's usage habits on various devices like tablets and smartphones.

4. Technical Implementation and Platform Selection

Technical implementation and platform selection involve software programming, architecture design, and the final deployment of the platform. During the technical implementation phase, the development team must choose the appropriate technology stack and development tools based on previous requirement analysis and design specifications. Common programming languages such as JavaScript and Python, along with development frameworks like React or Angular, are widely used in modern educational software development. Additionally, selecting the right database management system, such as MongoDB or MySQL, and backend services like Node.js, are crucial for ensuring software performance and

scalability. Platform selection entails deciding on which devices and operating systems the software will run, taking into account the device usage habits and access permissions of the target user group. For instance, if the target users primarily use iOS devices, releasing the app on Apple's App Store would be a priority, while Google Play and other Android app stores would be preferable if Android devices are more common among the user group. Furthermore, given the specific nature of preschool education software, the development team must also prioritize security and privacy protection to comply with relevant laws and regulations such as the Children's Online Privacy Protection Act (COPPA) and safeguard children's personal information. Through careful technical and platform choices, software can meet educational needs in terms of functionality, ensure stable and reliable performance, and adhere to legal requirements, thereby providing children with a safe, effective, and accessible learning environment (Zhang, Lilong, 2012).

5. Testing, Evaluation, and Iterative Optimization

Testing, evaluation, and iterative optimization constitute the ultimate stage of the preschool education software development process. The primary purpose of this phase is to verify the software's functionality and performance through systematic testing, evaluate its teaching effectiveness, and make necessary iterative improvements based on feedback. In this stage, the primary task is to conduct thorough software testing, including functional testing, performance testing, security testing, and compatibility testing, to ensure stable operation on different devices and platforms and to check for any potential security vulnerabilities. Additionally, particular attention is paid to whether the educational content and interaction design of the software meet the predetermined teaching objectives and are suitable for the cognitive and operational abilities of children. The evaluation process often requires integrating quantitative and qualitative research methods, including surveys, interviews, observations, and user testing, to provide direct data on the software's teaching effectiveness and user experience. Based on this data, the development team can identify deficiencies and improvement potentials in the software, enabling iterative development to continuously optimize the teaching content, user interface, and technical implementation. Emphasizing the iterative optimization process is crucial for maintaining the software's long-term competitiveness and adapting to rapidly changing technological environments, ensuring that the software continues to meet the demands of the education market and emerging educational concepts. Furthermore, a continuous user feedback loop is a key driver of iterative optimization. By establishing effective feedback mechanisms, the development team can promptly obtain user opinions and suggestions, enabling the software product to be more refined and better serve educational objectives and children's learning development.

Ⅳ. Strategies for Improving the Quality of Preschool Education Software

1.Resource Integration and Interdisciplinary Collaboration

Resource integration and interdisciplinary collaboration involve sharing resources and exchanging expertise to enhance the educational quality and technological level of software. This strategy relies on the collaboration of experts from various fields, including educational psychologists, early childhood education specialists, software engineers, interface designers, and child development experts, who collectively participate in the design, development, and evaluation of the software. Resource integration first involves diversifying teaching content, which encompasses not only the transmission of academic knowledge but also areas such as music, art, and physical activity to promote the comprehensive development of children. Table 3 illustrates the main participants and their contributions to interdisciplinary collaboration. Through this collaborative

approach, educational software can meet children's developmental needs on different levels while making educational activities more diverse and engaging. Additionally, collaboration extends beyond internal team experts to external educational institutions, research institutions, and families, forming a broad support network. Through regular seminars and feedback meetings, stakeholders can promptly exchange the latest research findings and educational concepts, inspiring each other and collectively driving innovation and progress in educational software (Wu, Aihong, 2010).

Table 3. Participants in Interdisciplinary Collaboration and Their Contributions

Professional Field	Participants	Contribution
Educational Psychology	Educational psychologists	Design teaching content that meets the developmental needs of young children.
Software Engineering	Programmers	Provide technical solutions to ensure stable software performance.
Interface Design	Designers	Develop user interfaces that are friendly and easy to use.
Child Development	Early childhood education specialists	Offer support for children's growth and learning theories.
Arts and Music	Artists, musicians	Enrich teaching content, enhance interactivity, and fun.

2.Enhancing Children's Engagement

Enhancing engagement in educational activities significantly increases children's motivation and educational outcomes. To achieve this, software development incorporates elements that capture children's attention and interest, including vivid visual effects, interactive game design, and teaching content tailored to their cognitive development stages. Implementing this strategy requires in-depth research into children's interests and preferences to ensure software content is closely related to their daily lives and can stimulate their curiosity. For instance, integrating role-playing games, simulation activities, or interactive stories allows children to learn new knowledge while engaging in play. Additionally, software is designed to allow children to choose different learning paths based on their learning pace and interests, providing personalized learning experiences. This not only increases children's sense of involvement but also promotes their active learning abilities through self-directed learning. Furthermore, educational software features high interactivity, utilizing touchscreen operations, voice feedback, and dynamic feedback to maintain children's interest in learning. These interactive features help children achieve a sense of immediate accomplishment and satisfaction during operation, further enhancing their learning motivation. Effective engagement enhancement strategies are based on principles from child psychology and education, continuously adjusted through ongoing user testing and feedback, to continually optimize software design, ensuring it truly captivates and sustains children's attention during actual use, achieving the desired educational outcomes.

3.Interaction Between Parents and Educators

The interaction between parents and educators not only enhances the application effectiveness of educational software but also optimizes the educational content and functionality of the software. In practical cases, a preschool education software named "Jia Yuan Qiao" exemplifies this well. The software allows educators to upload course content and students' learning progress, while parents can receive this information in real-time and provide feedback or suggestions based on

their children's learning progress. Additionally, the software includes parent forums and a Q&A module, enabling parents to engage in in-depth communication with teachers regarding their children's educational issues. Through this approach, parents not only become witnesses to their children's learning process but also active participants and facilitators. Statistical data indicates that in schools using "Jia Yuan Qiao," parental involvement increases by an average of 30%, and with the collaborative efforts of educators and parents, the learning outcomes of preschool children generally improve. This model emphasizes the bidirectional communication and mutual involvement between parents and educators, effectively integrating home education with school education, optimizing the allocation and utilization of educational resources, and enhancing the quality of education.

4.The Application of Implicit Learning and Gamified Education

Implicit learning and gamified education involve integrating learning content into games, allowing children to naturally acquire knowledge and skills while having fun. This learning approach not only enhances the enjoyment of learning but also improves its effectiveness. For example, an educational software designed specifically for preschool children, called "Le Xue Yuan," teaches colors, numbers, and basic logic through various interactive games. Each game in the software is based on research in educational psychology, ensuring that the game design not only captures children's attention but also achieves educational objectives. In "Le Xue Yuan," children learn by solving puzzles, completing memory games, or participating in role-playing activities. These game designs take into account children's age characteristics and cognitive development stages, ensuring the suitability of educational content and the scientificity of teaching methods. Furthermore, the concept of implicit learning emphasizes learning in informal and non-coercive environments, allowing children to develop problem-solving abilities and innovative thinking unconsciously. This gamified learning approach has been proven to effectively enhance children's learning motivation and long-term learning outcomes, making it particularly important for fostering children's autonomy and creativity in learning (Yu, Linchong, 2000).

5.Establishment of Continuous Updates and Feedback Mechanism

In the design and development process of preschool education software, establishing a continuous update and feedback mechanism is crucial. This mechanism not only provides ongoing motivation for software improvement but also ensures the software remains adaptable to changing educational needs and technological advancements. Effective feedback mechanisms involve collecting data and opinions from users (children, parents, and educators) and using this information to guide the iterative updates of the software. For example, through regular online surveys, user behavior analysis, and direct user feedback collection, the development team can identify shortcomings in the software and promptly respond to specific user needs. Additionally, employing advanced data analysis techniques such as machine learning and pattern recognition enables automated analysis of user behavior, thereby gaining insights into usage patterns and potential issues. These analytical insights support the development team in precisely adjusting teaching content and interface design during version iterations to enhance user interaction and learning effectiveness.

Ⅴ. Conclusion

When designing and developing preschool education software, following principles of developmental appropriateness, combining entertainment with education, and establishing feedback mechanisms are crucial. Through these strategies, the software is not only supporting children's learning but also providing an environment that combines learning with fun.

Integrating interdisciplinary resources and promoting children's participation enhances the richness and interactivity of the learning experience. The continuous updating of the software and effective integration of user feedback ensure its long-term applicability and educational effectiveness. Therefore, meticulously designed preschool education software can significantly improve the quality of education and meet the needs of modern education.

References

Wang, Wangfen (2015) "Strategies for Design and Development of Multimedia Teaching Software", Information and Computers (Theoretical Edition), (07), 60+62.

Zhang, Binglin (2014) "Strategies for Design and Development of Preschool Education Software", Preschool Education Research, (09), 22-30. DOI:10.13861/j.cnki.sece.2014.09.003.

Zhang, Lilong (2014), Design and Development of Interactive Mathematics Teaching Software for Preschool Children (Master's Thesis), Wuhan: Central China Normal University.

Wu, Aihong (2010), Design and Development of "Preschool English Education" Online Course Based on Blended Learning (Master's Thesis), Baoding: Hebei University.

Yu, Linchong (2000) "Development of Educational Software and Its Development Strategies", Modern Computer , (05), 88-89.