

Research Article

Squeak:Nurturing Creativity and Innovation with an Open Source Smalltalk Language

Dr. Latika Kharb, Dr. Deepak Chahal

Professor, Jagan Institute of Management Studies, Sector-5, Rohini, Delhi.

Submission: 01/03/2025;

Received: 09/03/2025;

Revision: 26/03/2025;

Published: 02/04/2025

*Corresponding author: Dr. Latika Kharb (latika.kharb@jimsindia.org)

Doi: <https://doi.org/10.61336/jmsr/25-02-10>

Abstract: Squeak is a powerful open-source Smalltalk programming environment that has gained attention for its versatility and dynamic nature in short time span. Squeak provides a comprehensive platform for Smalltalk development, offering a range of features that make it an attractive choice for both novice programmers and experienced developers. One notable aspect of Squeak is its cross-platform compatibility, running seamlessly on various operating systems, including Windows, macOS, and Linux. This versatility ensures that developers can work on their preferred operating system without sacrificing the benefits of the Smalltalk environment. Moreover, Squeak's interactive development environment facilitates rapid prototyping and experimentation, allowing developers to iterate quickly and refine their code efficiently.

Keywords: Smalltalk, object oriented, cross platform, prototyping, IDE.

INTRODUCTION

software development. Renowned for its simplicity, elegance, and dynamic nature, Squeak empowers developers to create robust, maintainable, and innovative solutions [1]. In this article, we explore the key features and advantages of Squeak, shedding light on how this open-source language is fostering creativity and collaboration within the programming community [2].

• Object-Oriented Simplicity:

At the core of Squeak lies the Smalltalk language, celebrated for its object-oriented paradigm. Objects are not just a programming concept in Smalltalk; they are the foundation of the language [3]. Everything in Smalltalk is an object, leading to a clean, intuitive, and consistent syntax that promotes readability and maintainability. Squeak inherits and enhances these characteristics, providing developers with a seamless and elegant coding experience.

• Cross-Platform Compatibility:

Squeak's commitment to versatility is evident in its cross-platform compatibility. Developers can write code in Squeak on one operating system and effortlessly run it on another, including Windows, macOS, and Linux. This flexibility is invaluable for collaborative projects, ensuring that team members can work in their preferred environments without compromising the integrity of the codebase[4].

• Interactive Development Environment (IDE):

Squeak boasts an interactive development

environment that sets it apart. The live coding capabilities allow developers to experiment, test, and modify code in real-time [5]. This iterative process facilitates rapid prototyping and debugging, empowering programmers to refine their solutions efficiently. The interactive nature of Squeak's IDE contributes to a dynamic and engaging development experience.

• Open Source Community Collaboration:

Squeak's open-source nature has led to the formation of a vibrant and collaborative community. Developers around the world contribute to the language's evolution, share insights, and collaborate on projects. This communal approach not only ensures continuous improvements and updates but also provides a supportive environment for learning and mentorship[6].

• Multimedia and Graphical Capabilities:

Squeak extends beyond traditional software development by incorporating multimedia and graphical capabilities. Its built-in support for graphical user interfaces (GUIs) and multimedia elements opens the door to the creation of interactive and visually engaging applications [7] [8]. This feature makes Squeak particularly appealing for projects requiring a blend of functionality and aesthetics.

• Educational Impact:

Squeak has found a significant presence in educational settings. Its simplicity and interactivity make it an ideal language for teaching

programming concepts to beginners. Moreover, the open-source nature of Squeak encourages educational institutions to adopt and adapt the language to suit their specific curriculum needs, fostering a new generation of programmers with a strong foundation in object-oriented thinking.

THE INTERFACE OF SQUEAK

Here's a general overview of the typical UI elements you might encounter in Squeak:

- **Morphic UI:**

Squeak uses a graphical user interface framework called Morphic. Morphic is known for its flexibility and is designed to be highly customizable. In Squeak, everything is a "Morph" — a graphical object that can represent a window, a button, a text editor, or even a complex graphical element. Morphs can be freely manipulated on the screen, providing a dynamic and interactive environment for developers [9].

- **World and Project Organization:**

Squeak's UI often includes a concept of the "World," which acts as the main container for all graphical elements. Within the World, developers can organize projects, windows, and various Morphs. The World serves as the canvas where developers can interact with and manipulate different elements [10].

- **Integrated Development Environment (IDE):**

The Squeak environment integrates the development tools seamlessly into the graphical interface. Developers can open code browsers, inspectors, and debuggers directly from the UI. The ability to edit and execute code in real-time is a hallmark of the Smalltalk development experience, and the UI is designed to facilitate this iterative coding style.

- **Browser-based Code Editing:**

Squeak typically employs a browser-based approach for code editing. Developers use code browsers to navigate and edit classes and methods. The browsers provide a structured view of the codebase, making it easy to explore and modify different aspects of the application [11].

- **Menus and Toolbars:**

Squeak includes menus and toolbars that provide access to various development tools and functionalities. These menus and toolbars are designed to be intuitive and user-friendly, offering quick access to commonly used actions.

- **Graphics and Multimedia Tools:**

Given Squeak's support for multimedia and graphical applications, the UI often includes tools for creating and manipulating graphics. This can range from simple drawing tools to more advanced features for handling multimedia elements [12] [13].

- **Debugger and Profiler:**

Squeak's UI includes debugging and profiling tools that help developers identify and resolve issues in

their code. The debugger, in particular, allows for step-by-step execution and inspection of code during runtime.

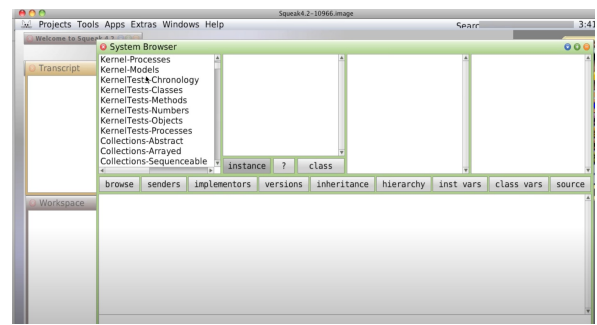


Figure 1: UI screen of Squeak

It's important to note that the unique aspects of Squeak's UI, including the use of Morphic and the interactive development environment, contribute to the distinctive Smalltalk programming experience [14]. For the most accurate and up-to-date information on Squeak's UI, it's recommended to refer to the official Squeak documentation or community resources [15].

CONCLUSION

Squeak, as an open-source Smalltalk language, represents a testament to the power of simplicity, collaboration, and innovation in the world of programming. Its object-oriented approach, cross-platform compatibility, interactive development environment, and support for multimedia applications position it as a valuable tool for developers, educators, and learners alike. As Squeak continues to evolve with the contributions of its dedicated community, it remains a shining example of how open source can propel a programming language into the forefront of creativity and technological advancement.

Conflict of Interest: The authors declare that they have no conflict of interest

Funding: No funding sources

Ethical approval: The study was approved by the Jagan Institute of Management Studies, Sector-5, Rohini, Delhi.

REFERENCES

1. Goldberg, A., & Robson, D. (1983). *Smalltalk 80: The Language and Its Implementation*. Reading, MA: Addison Wesley. ISBN: 0-201-13688-0
2. LaLonde, W., & Pugh, J. (1990). *Inside Smalltalk: Volume 1*. Prentice Hall. ISBN: 0-13-468414-1
3. Kharb, L. (2017). Exploration of Social Networks with visualisation Tools. *American Journal of Engineering Research (AJER)*, 6 (3), 90-93.
4. Kharb, L. (2018, January). A perspective view

- on commercialization of cognitive computing. In 2018 8th International Conference on Cloud Computing, Data Science & Engineering (Confluence) (pp. 829-832). IEEE.
5. Sharp, A. (1997). Smalltalk by Example. McGraw-Hill. <http://stephane.ducasse.free.fr/FreeBooks/ByExample/i>
 6. Kharb, L. (2016). Automated deployment of software containers using dockers. *International Journal of Emerging Technologies in Engineering Research (IJETER)*, 4(10), 1-3.
 7. Kharb, L. (2015). IBM Blue mix: Future development with open cloud architecture. *JIMS8I-International Journal of Information Communication and Computing Technology*, 3(2), 165-168.
 8. Kharb, L., & Kaur, S. Embedding Intelligence through Cognitive Services. *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, ISSN, 2321-9653.
 9. Kharb, L., & Singh, P. (2021). Role of machine learning in modern education and teaching. In *Impact of AI Technologies on Teaching, Learning, and Research in Higher Education* (pp. 99-123). IGI Global.
 10. Woolf, B. (1998). Null Object. In R. Martin, D. Riehle, & F. Buschmann (Eds.), *Pattern Languages of Program Design 3*. Addison Wesley, 5-18.
 11. Kharb, L. (2017). Exploration of social networks with visualization tools. *American Journal of Engineering Research (AJER)*, 6(3), 90-93.
 12. Kharb, L. (2014). Proposing a Comprehensive Software Metrics for Process Efficiency. *International Journal of Scientific and Engineering Research (IJSER)*, 5(6), 78-80.
 - [13]Chahal, D., & Kharb, L. (2019). Smart diagnosis of orthopaedic disorders using internet of things (IoT). *Int. J. Eng. Adv. Technol*, 8, 215-220.
 13. Singh, P., Chahal, D., & Kharb, L. (2020). Predictive strength of selected classification algorithms for diagnosis of liver disease. In *Proceedings of ICRIC 2019: Recent Innovations in Computing* (pp. 239-255). Springer International Publishing.
 14. Singh, R., Singh, P., Chahal, D., & Kharb, L. (2021). "VISIO": An IoT Device for Assistance of Visually Challenged. In *Advances in Electromechanical Technologies: Select Proceedings of TEMT 2019* (pp.949-964). Springer Singapore.
 15. Singh, R., Singh, P., Chahal, D., & Kharb, L. (2021). "VISIO": An IoT Device for Assistance of Visually Challenged. In *Advances in Electromechanical Technologies: Select Proceedings of TEMT 2019* (pp. 949-964). Springer Singapore.