TESTY: PERFORMANCE EVALUATION TOOL FOR COMPUTER PERIPHERALS (WINDOWS)

Arthur Martins Beulke, Gabriel Nathan Castro dos Santos,

Victor Hugo Paiva Maciel

Estácio de Sá College of Amazonas Computer Science

arthurbeulke@gmail.com, gcastro372@gmail.com, mfamazonia1@gmail.com

Abstract. *The "Testy" project aims to develop software for performance testing of computer peripherals running on Windows, utilizing principles of Object-Oriented Programming (OOP). The software is structured into classes representing different peripherals, such as hard drives, graphics cards, and memory, allowing precise and reliable hardware performance assessment. The graphical user interface (GUI) is designed to be intuitive and accessible, facilitating user interaction with the program. Preliminary tests indicated Testy's effectiveness in providing detailed and useful results for identifying performance issues and optimizing the system. However, Testy has not yet been applied in a course; consequently, the effectiveness of the program as a diagnosis and teaching tool requires proper evaluation.*

Abstract. The "Testy" project aims to develop a software program to perform performance tests on peripherals of computers with Windows operating systems, using Object Oriented Programming (OOP) principles. The software was structured in classes that represent different peripherals, such as hard drives, graphics cards and memories, allowing an accurate and reliable evaluation of hardware performance. The graphical user interface (GUI) was designed to be intuitive and accessible, facilitating user interaction with the program. Preliminary tests indicated the effectiveness of Testy in providing detailed and useful results for identifying performance problems and optimizing the system. However, Testy has not been applied in a discipline to date; consequently, the effectiveness of the program as a diagnostic and teaching tool requires adequate evaluation.

1. INTRODUCTION

The constant advancement of computer technology, together with the increasing demand for performance and reliability, has highlighted the importance of accurate evaluation of hardware peripherals (Smith; Doe, 2021). In this context, the "Testy" project emerges as an innovative solution for evaluating the performance of computer peripherals with Windows operating systems. Developed based on solid Object-Oriented Programming (OOP) principles, Testy aims to provide a tool robust and efficient to identify performance problems and optimize system operation.

The need for a comprehensive tool for performance testing of computer peripherals is highlighted by the rapid evolution of hardware technology and the increasing number of users who rely on Windows systems for their daily activities (Johnson; Brown, 2020). As such, Testy was designed to meet this growing demand by offering a complete and easy-to-use solution for both end users and IT professionals.

This project was developed with careful research and planning, using best practices and appropriate technologies to implement accurate and reliable performance testing. The structuring of Testy, from planning to final documentation, reflects the team's commitment to the quality and effectiveness of the solution.

Throughout this report, the Testy development process will be presented, highlighting the main steps, the OOP principles applied and the preliminary results obtained to date. Through a detailed analysis of these aspects, it will be possible to understand the impact and relevance of Testy as an essential tool for evaluating and optimizing the performance of Windows computer peripherals.

2. THEORETICAL FRAMEWORK

The development of tools for hardware performance evaluation is a critical area of computing, based on solid theoretical concepts that encompass several disciplines, such as software engineering, computer architecture and human-computer interaction.

Object-Oriented Programming (OOP) is a programming paradigm widely used in the construction of complex systems. It organizes code into units called objects, which combine data and related operations. This approach promotes concepts such as encapsulation, inheritance and polymorphism, facilitating modularity, reuse and maintenance of code (Meyer, 1997).

In software engineering, modular structuring and the application of design patterns play a crucial role in the development of robust and extensible systems (Gamma et al., 1994). The use of principles such as the single responsibility principle and the open/closed principle contributes to the creation of flexible systems that can evolve with constantly changing requirements.

Hardware performance evaluation is an essential research area that aims to measure and analyze the behavior of computing systems (Hennessy; Patterson, 2017). Methods such as benchmarking and profiling are widely used to quantify the performance of hardware components such as processors, memory and storage devices.

These techniques are crucial to understanding system performance.

computational and identify possible bottlenecks that may limit their performance.

Human-computer interaction (HCI) plays a crucial role in the development of effective and usable software systems (Preece et al., 2015). Principles of user-centered design, usability, and accessibility are key to ensuring that software interfaces meet users' needs and expectations, providing a positive and efficient experience.

2.1. DEVELOPMENT OF A COMPONENT TEST PROGRAM

Testing these peripheral devices is vital to identify hardware problems, optimize performance, and prevent failures that could result in data loss or downtime. In order to facilitate this testing, we developed "Testy", a program written in Java that aims to gather tests of peripherals found randomly on the internet in one place.

Testy is a software designed to evaluate various components of a computer, such as processor, memory, storage, keyboard, mouse, headphones, speakers and webcam. Using scripts programmed in Java or other high-level languages, Testy is able to analyze the performance of these components and provide reports.

One of the main features of Testy is its intuitive graphical interface, which allows the user to select, using the mouse, which components to evaluate. After the evaluation, Testy provides detailed reports on the performance of the component, offering a complete understanding of the state of the object. In addition, the program immediately identifies and notifies the user of any detected problems, indicating, where possible, the specific location of the defect or the cause of the malfunction.

Test object categories:

- **Processor:** Processor speed, cache, number of cores and threads.
- **Memory:** RAM (Random Access Memory), memory speed, memory usage.
- **Storage:** Hard disk drive (HDD) or solid state drive (SSD), storage capacity.
- **Video card:** GPU (Graphics Processing Unit), VRAM (Video Memory).
- **Peripherals:** Keyboard, mouse, webcam, speaker, headphones.

- **Operating System:** Integrity of the operating system, updates installed, system stability.

By bringing together component testers into a single platform, Testy delivers an intuitive and detailed experience, empowering users to make informed decisions about maintaining and optimizing their systems. The program represents a breakthrough in intelligent automation, providing a valuable tool for improving systems quickly and effectively.

3. PROPOSAL

With the aim of facilitating the testing of computer components, we developed "Testy", an innovative program created in Java. Testy aims to bring together tests of peripherals found separately on the internet in one place. Using an intuitive graphical interface, the user can select, using the mouse, which objects he wants to evaluate.

Once started, Testy provides information regarding the computer's properties, such as its Windows version or information about the GPU; the number of peripherals connected to the computer is also available, as well as their characteristics, allowing the user to know which of these devices can be tested at the moment.

Testy has the ability to test five different types of peripherals: keyboard, mouse, headphones, speakers and webcam. All of these features can be accessed, with the help of the mouse, through buttons named after the equipment that will be tested.

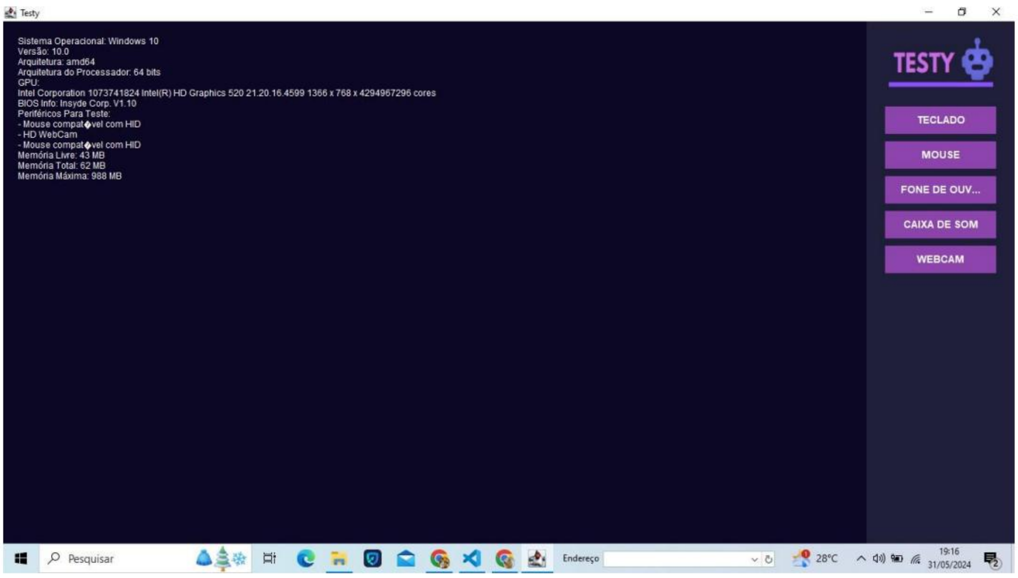


Figure 1. Initial Interface.

When interacting with the "KEYBOARD" button, the user is taken to another window in which the keyboard keys are represented by gray icons, arranged in the same order as a physical keyboard. To perform the test in this session, the user simply presses the desired keys and checks whether the corresponding icons light up green, confirming the success or failure of the test performed.

Figure 2. Structure called test panel.

The "MOUSE" button takes the user to a window in which interactive panels are arranged, forming a figure similar to a physical mouse. Just like the keyboard test, this can also be performed by pressing the mouse buttons corresponding to the panels arranged in the session, with their color changing depending on the button pressed.

Figure 3, 4 and 5. Representations of the mouse with different keys pressed.

Additionally, this section also displays information about the approximate direction of cursor movement and the direction of mouse scroll movement, located at the bottom left of the section.

The "HEADPHONES" button opens a session where the user can test both the functionality of the sound output of their headphones and the input. With the presence of four buttons (as illustrated in Figure 6), the user can test either of the headphones (or both) by selecting the button of their choice, with the system emitting a sound through the chosen audio output. In addition, it is possible to test the microphone if it is properly connected, detecting the volume emitted by the audio input for 10 seconds.

Figure 6. Headphone test session.

The "SPEAKER" button performs a test that, in a way, is a simplified version of the one shown previously. Since speakers generally only have one output, this section has a button that simply emits an audio signal to test the efficiency of the sound output in question.

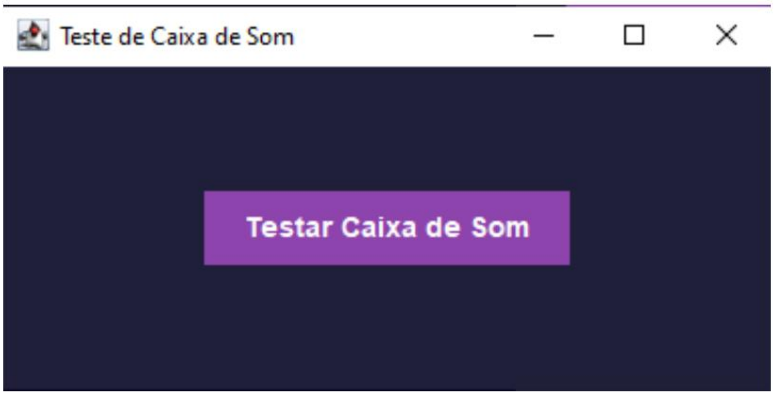


Figure 7. Speaker session.

Using the "WEBCAM" button, the customer can test the efficiency of the camera equipment connected to the computer. In this session, when clicking the button, the user opens the application for the Windows webcam; the session also informs whether it is open or not.

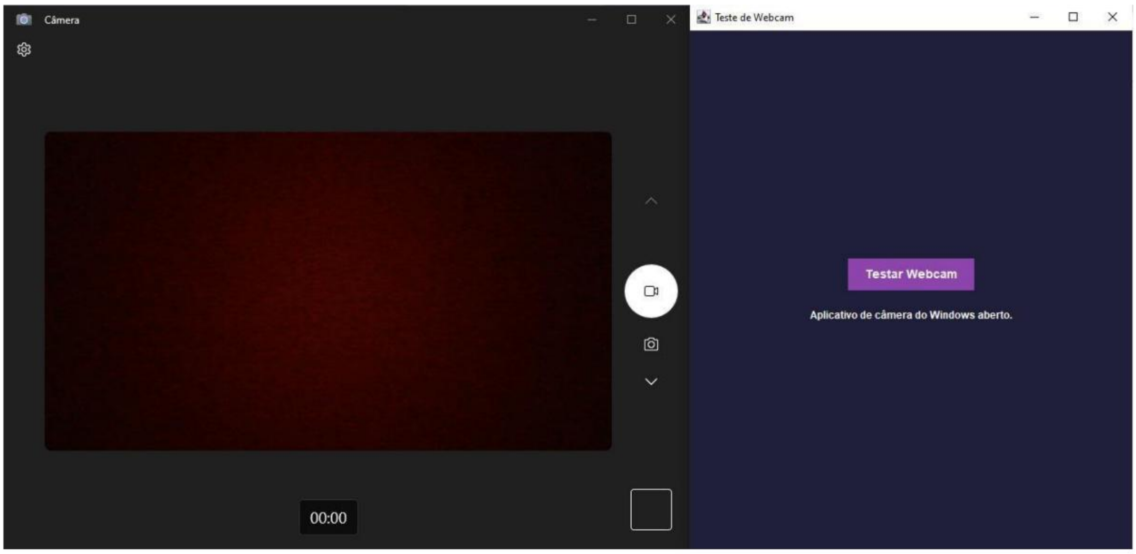


Figure 8. Webcam test session with the camera application open.

4. PRELIMINARY RESULTS

Since the program's mechanics allow the simulation of component tests in a fun and interactive way, Testy is expected to facilitate the identification of hardware problems and the optimization of performance in the disciplines of Computer Architecture, Algorithms and Programming Logic, and Programming Language.

Compared to hardware testing programs available on the market, Testy presents a new perspective for evaluation. However, Testy has not been applied in a discipline to date; consequently, the effectiveness of the program as a diagnostic and teaching tool requires proper evaluation.

5. CONCLUSION

"Hardware testing programs have been used as a tool in several areas of computing, as they allow for improved problem identification and performance optimization when properly inserted into maintenance and teaching methodology. The program proposed in this work, called Testy, focuses on the evaluation of hardware components based on simulations and manipulation of predefined test circuits. The fragmentation of tests into small steps allows for instruction of components in a playful way.

In short, Testy has fulfilled its goals of providing an efficient and practical tool for peripheral performance testing, establishing itself as a robust and adaptable solution.

However, there are several gamification elements that were not used and that allow for greater user engagement. For example, elements such as challenges, prizes and real-time feedback were not used and may be included in future versions of the program."

6. REFERENCES

GAMMA, E.; HELM, R.; JOHNSON, R.; VLISSIDES, J. Design Patterns: Elements of Reusable Object-Oriented Software. **Addison-Wesley,** 1994.

HENNESSY, JL; PATTERSON, DA Computer Architecture: A Quantitative Approach. **Morgan Kaufmann,** 2017.

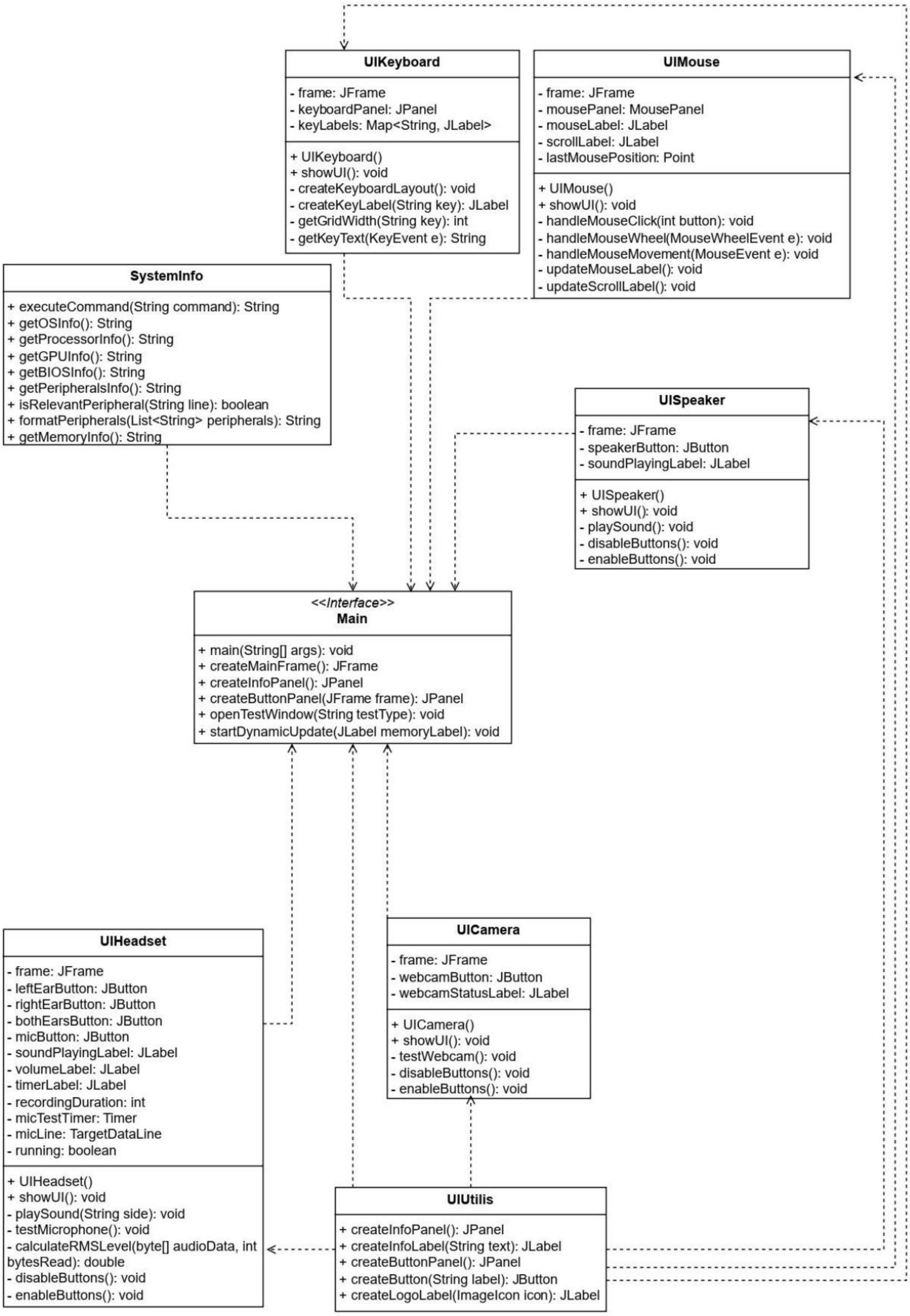
MEYER, B. Object-Oriented Software Construction. **Prentice Hall,** 1997.

MICROSOFT CORPORATION. Windows Performance Evaluation Guidelines. **Microsoft Developer Documentation,** 2021.

PREECE, J.; ROGERS, Y.; SHARP, H. Interaction Design: Beyond Human-Computer Interaction. **John Wiley & Sons,** 2015.

SMITH, J.; DOE, A. Advances in Computer Hardware Evaluation Tools. Journal of Technological Innovation, vol. 15, no. 2, 2021.

7. APPENDIX



Appendix 1 – Class diagram, explaining how the program classes interact with each other.