An Open Source Tool for Simulating a Variety of Vision Impairments in Developing Swing Applications

Theofanis Oikonomou[1], Konstantinos Votis[1], Dimitrios Tzovaras[1], and Peter Korn[2]

[1] Informatics and Telematics Institute, 6th Km Charilaou-Thermi Road,
Thermi-Thessaloniki, PO BOX 60361, GR-57001, Greece
{thoikon,kvotis,tzovaras}@iti.gr
[2] Sun Microsystems, Inc., 17 Network Circle, MPK17-101,
Menlo Park, CA 94025, USA
Peter.Korn@Sun.COM

Abstract. A lot of tools have been created lately in order to simulate how a vision impaired or color-blind person would perceive web rich-client applications and content. In this work we propose a simulation tool for non-web Java ™ Swing applications. The aim is to assist the developers in preventing accessibility barriers and improving the overall quality throughout the design and implementation phases of the whole development process.

1 Introduction

It is important to realize that people with disabilities are not just a tiny minority of the population of the European Union. The lowest estimate, based on the extremes of currently defined disablement categories, puts their total number at around 40 Million persons (nearly 11% of the population of the EU[1]). Designing for people with disabilities is becoming an increasingly important topic for a variety of reasons, especially due to the recent legislation in many countries promoting their rights. Consequently, even if people with disabilities want to be independent and do things for themselves by themselves, unfortunately, most Information and Communication Technologies (ICT) applications and systems are not fully accessible today.

Assistive technologies such as screen readers, screen magnifiers, speech recognition systems and Braille terminals help make applications and content accessible to people with disabilities. To accomplish this goal, accessibility standardization activities have been actively performed in various areas, and many institutions and organizations [1,2,3,4,5,6] are introducing new criteria for assessing accessibility [7,8,9].

One of the problems in adopting these standards is the lack of tool support. In order to create documents and applications that are compliant with standards or guidelines, tools for accessibility checking, evaluation and simulation are needed to reduce the burdens on document authors and application/content developers. For the traditional HTML-based web content, various accessibility checking tools have been developed. Currently, sufficient tools do not exist for non-web rich-client applications and content.

[1] http://europa.eu.int/comm/eurostat/Public/datashop/printproduct/EN?catalogue=Eurostat product=3-11012002-EN-AP-EN mode=download

C. Stephanidis (Ed.): Universal Access in HCI, Part I, HCII 2009, LNCS 5614, pp. 135–144, 2009.

To this end we propose a new vision impairment simulation tool for Java ™ Swing applications.

The remaining of this paper is organized as follows: We briefly discuss the vision impairments that were included in our work in Sect. 2. We describe the most representative web rich-client content evaluation and simulation tools and Graphical User Interface (GUI) validation tools in Sect. 3. In Sect. 4 we present the proposed simulation tool followed by some screenshots. Final remarks are discussed in Sect. 5 which concludes this paper.

2 Vision Impairments

We simulate color blindness and various low vision impairments such as loss of central and peripheral vision, blurred vision, extreme light sensitivity and night blindness. These impairments, especially central and peripheral vision loss, have a negative impact on computer use, since modern operating systems employ GUIs which require the use of eye-to-hand coordination to operate the mouse.

The loss of central vision creates a blur or blind spot, but side (peripheral) vision remains intact. This makes it difficult to read, recognize faces, and distinguish most details in the distance. Mobility, however, is usually unaffected because side vision remains intact. Typical examples of central vision loss are cataract and macular degeneration.

Loss of peripheral vision is characterized by an inability to distinguish anything to one side or both sides, or anything directly above and/or below eye level. Central vision remains, however, making it possible to see directly ahead. Typically, loss of peripheral vision may affect mobility and if severe, can slow reading speed as a result of seeing only a few words at a time. This is sometimes referred to as "tunnel vision". Typical examples of peripheral vision loss are glaucoma and retinitis pigmentosa.

Blurred vision causes both near and far to appear to be out of focus, even with the best conventional spectacle correction possible. It may cause difficulties in reading texts with normal formal font size, mobility, everyday activities such as cooking, sewing, cleaning, using computers etc.

Extreme light sensitivity exists when standard levels of illumination overwhelm the visual system, producing a washed out image and/or glare disability. People with extreme light sensitivity may actually suffer pain or discomfort from relatively normal levels of illumination. Difficulties in activities either at dark or at a too bright light or in working in premises that are not properly lit.

Night blindness results in inability to see outside at night under starlight or moonlight, or in dimly lighted interior areas such as movie theaters or restaurants. There may be difficulties in a number of activities performed at dark after daylight such as crossing the street, reading signs etc.

Color blindness is a lack of sensitivity to certain colors. Common forms of color blindness include difficulty distinguishing between red and green, or between yellow and blue. Sometimes color blindness results in the inability to perceive any color. Color blindness is problematic in driving vehicles, reading signs or maps, watching TV, working on a computer, understanding colorful graphics and charts etc.

3 Related Work

In this section we briefly describe the most well known web accessibility evaluation tools, also known as testing or assessment tools (Sect. 3.1). Additionally, we enumerate the most technologically advanced web accessibility simulation tools (Sect. 3.2) and finally we elaborate on some GUI validation tools (Sect. 3.3).

3.1 Web Accessibility Evaluation Tools

Web accessibility evaluation tools check web pages against the two most commonly cited standards for web accessibility [2,10], point out errors or potential problems, and advise you to correct or double check them. Automated accessibility checking tools cannot make firm judgments about the accessibility of everything on a web page. Some of the most well known evaluation tools are:

A-Prompt [12]. A-Prompt is a downloadable software tool which first evaluates the web page to identify barriers to accessibility and then provides assistance in making the necessary repairs. You can check one page or focus on a particular element in a page.

Color Contrast Check [13]. This tool allows specifying a foreground and a background color and determining if they provide enough of a contrast when viewed by someone having color deficits or when viewed on a black and white screen.

Contrast Analyser [14]. It is primarily a tool for checking foreground and background color combinations to determine if they provide good color visibility. It also contains functionality to create simulations of certain visual conditions such as color blindness.

WAVE Web Accessibility Tool [11]. WAVE displays a web page graphically with icons that indicate errors or possible problems. You can even continue to browse within the site while using WAVE.

3.2 Web Accessibility Simulation Tools

Simulation tools present web pages as it would appear under specific circumstances, such as to someone using a text-only browser or to someone with color blindness. Simulation tools can help in some of the areas that accessibility testing tools leave to your judgement. Some of the most representative simulation tools are:

Accessibility Color Wheel [15]. This tool analyzes the contrast of a color pair. It simulates how people with three forms of color blindness might see the colors.

aDesigner [20]. aDesigner is a disability simulator that helps web designers ensure that their pages are accessible and usable by the visually impaired. The tool looks at such elements as the degree of color contrast on the page, the ability of users to change the font size, the appropriateness of alternate text for images, and the availability of links in the page to promote navigability. The tool also checks the pages' compliance with accessibility guidelines. The result of this analysis is a report listing the problems that would prevent accessibility and usability by visually impaired users. In addition, each page is given an overall score. With this information, web developers get immediate feedback and can address these obstacles before the pages are published.

Vischeck Color Blindness Simulation Tool [16]. A color blindness simulator that can be used online or as a Photoshop plug-in. The online tool simulates color blindness on an image that you upload or on a web page that you specify, while the Photoshop plug-in changes the colors of the document you are working on.

VIS [17]. Visual Impairment Simulator is an educational tool that simulates what it is like to use Microsoft Windows® with a visual impairment. When the program runs, it manipulates the images on the user's screen so that it seems like the user has a visual impairment such as color blindness or macular degeneration among others. The user is able to pick which visual impairment to use and the severity of the impairment.

WebAIM Low Vision Simulation Tool [18]. This tool provides an opportunity for web developers to experience a web page using simulated visual disabilities. While it certainly does not simulate low vision itself, it can be used to help understand how visual disabilities can impact web content and how web content can be better designed.

3.3 Graphical User Interface Validation Tools

a11y [23]. The a11y module is an accessibility checker Netbeans [22] plug-in that helps developers to make their GUI forms compliant with accessibility rules effectively and without unnecessary effort. It "listens" for any event in the Form Editor and checks whether all its components meet required criteria. If they don't, adequate message is generated and listed in a table with description or recommendation for the user.

RAVEN [21]. Raven, which stands for Rule-based Accessibility Validation Environment, is an Eclipse tool [24] for inspecting Java^{TM} based GUIs and web pages and validating them for accessibility. It uses an innovative Aspect-Oriented Programming (AOP) technique to inspect the application as it is executed. This tool supports Java AWT or Swing and Eclipse Standard Widget Toolkit (SWT) GUIs, including Eclipse plug-ins.

4 Vision Impairment Simulator

As stated in Sec. 3.3 there is a tool (a11y) that helps the developer during the GUI implementation face by making sure all the accessibility information about every GUI component is set. This guarantees that the information will be available to an impaired user through the aid of an assistive technology. Having that in mind we devised two new modules for the Netbeans Integrated Development Environment (IDE) and one stand-alone application that can give the developer an idea of what would his GUI actually look like to a vision impaired user. In our work we used the Sun Java^{TM} Standard Edition Development Kit (JDK), the Java Accessibility Application Programming Interface (JAAPI) and the Netbeans IDE.

4.1 Preview Design in Vision Impairment Simulator Module

As you design the User Interface (UI), the GUI builder shows the form in the design area of the NetBeans IDE (Fig. 2). This is an accurate but static representation of the UI form. You cannot interact with it as if it were actually running. Static and non-interactive views are fine while you are arranging components, but this view does not help you see how form components will behave as you resize the form, nor can you test the order of tabs. The "Preview Design in Vision Impairment Simulator" module (PreDeVIS) provides a visual design preview feature that allows you to see how the form will display in your application. You can activate the design preview by clicking on the preview icon, a small image of an eye with an arm-chair, at the IDE's Toolbar (Fig. 1).

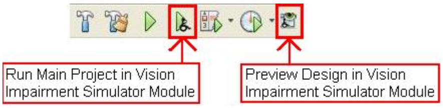

Run Main Project in Vision Impairment Simulator Module
Preview Design in Vision Impairment Simulator Module

Fig. 1. Netbeans toolbar with the two new modules installed

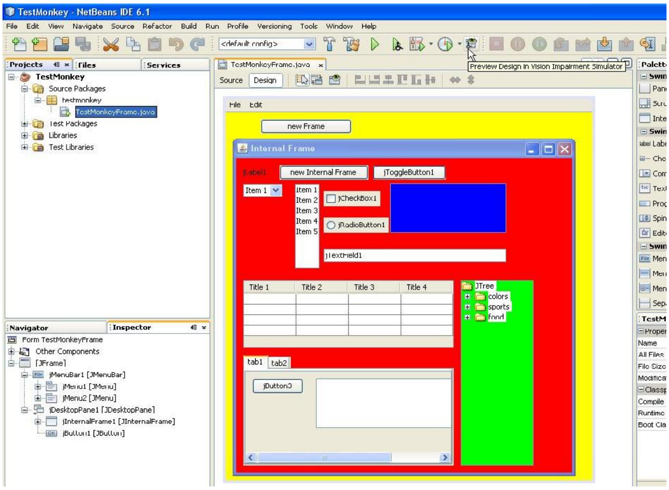
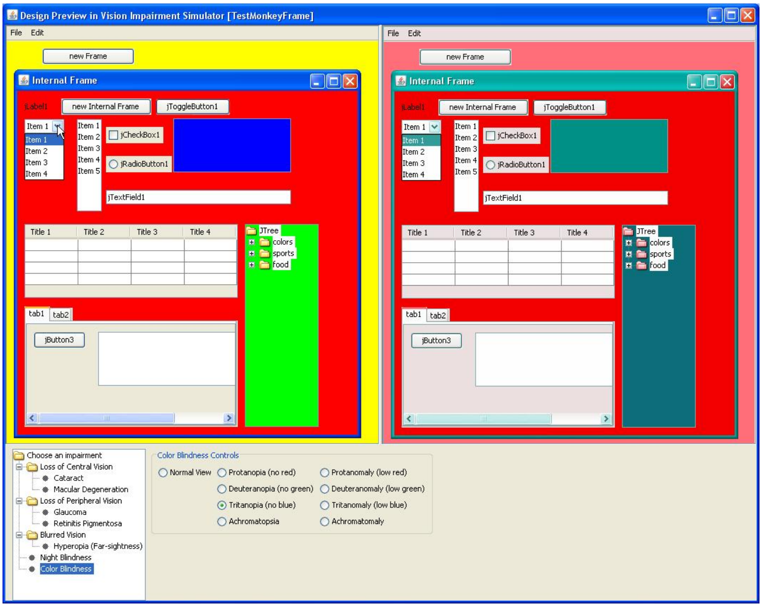


Fig. 2. Netbeans GUI builder

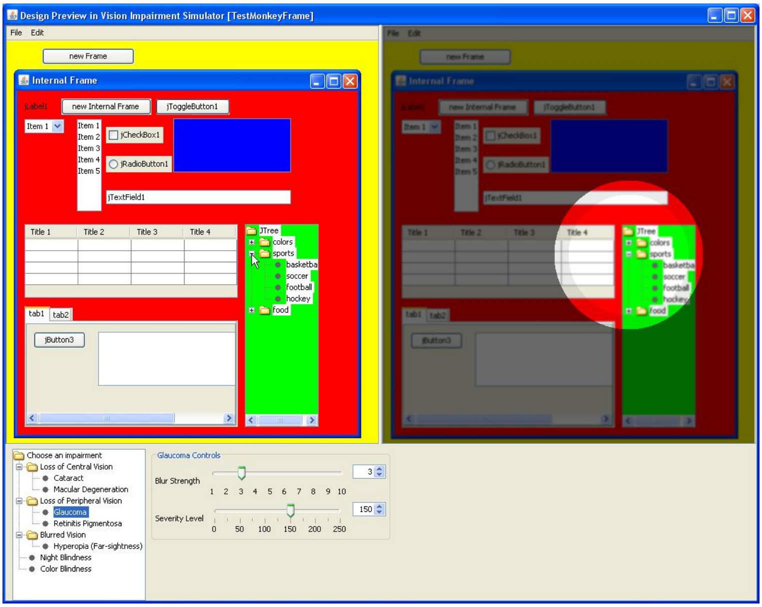
When you click on the preview icon, the GUI builder will activate the form in Preview mode. The preview gives an indication of whether the component alignments and anchors are set the way you want them. Preview mode lets you type information into text fields, tab from field to field, resize the form and simulate various visual deficiencies. We can identify four different regions. The first one, located in the upper left part of the PreDeVIS is the previewed form. The simulated form is located in the upper right part. Any action made in the previewed form is propagated in the simulated form. Furthermore, we can specify which impairment to simulate from the control panel found in the lower left part of the simulator and control various factors regarding the specific impairment from the controls located in the lower right part of



Color Blindness Controls

○ Normal View
○ Protanopia (no red)
○ Protanomaly (low red)
○ Deuteranopia (no green)
○ Deuteranomaly (low green)
● Tritanopia (no blue)
○ Tritanomaly (low blue)
○ Achromatopsia
○ Achromatomaly

Fig. 3. Preview Design module simulating tritanopia



Glaucoma Controls

Blur Strength 3

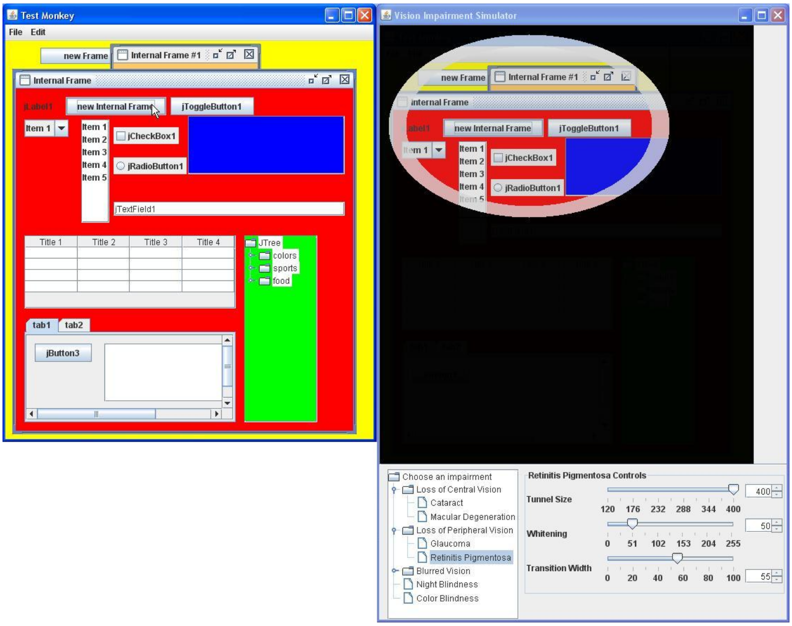
Severity Level 150

Fig. 4. Preview Design module simulating glaucoma

the preview simulator. For example we can choose to simulate how a color blind user with tritanopia [25] would perceive our form when he clicks on a combo-box (Fig. 3) or when a low vision user with glaucoma expands a tree (Fig. 4).

4.2 Run Main Project in Vision Impairment Simulator Module

The developer can use the PreDeVIS module in order to quickly preview the way the form he designs in the Netbeans IDE will look like. This way the functionality of the various GUI components present in the form cannot be tested. The "Run Main Project in Vision Impairment Simulator" module (RunVIS) comes to his aid. You can activate this module by clicking on the RunVIS icon, an image of a green triangle with an arm-chair, at the IDE's Toolbar (Fig. 1). With this module the developer has the ability to explore the application and test if the functionality he has programmatically set to each GUI component actually works. While the application is running new windows, such as dialogs, choosers or frames, may appear due to a user action. The module automatically simulates the window that has the user's focus. Another interesting feature is that the module inherits the Look and Feel (L&F) that was set to the application by the developer. For example the developer can see how a user with retinitis pigmentosa opens an internal frame while the application has the Metal L&F (Fig. 5) or how a user with cataract clicks on a check box while the application has the Windows L&F (Fig. 6). Notice that the Windows high contrast setting is turned on and the "High Contrast White" appearance scheme is activated.



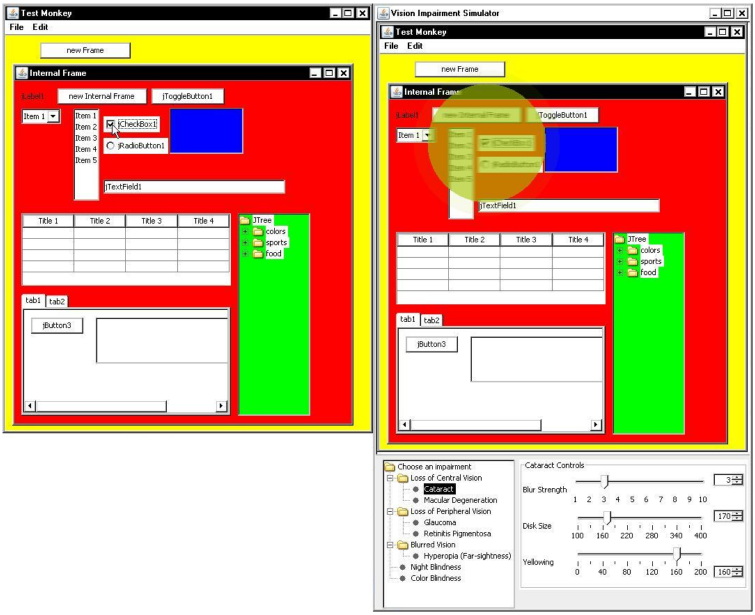
Retinitis Pigmentosa Controls

Tunnel Size 400

Whitening 50

Transition Width 55

Fig. 5. RunVIS module simulating retinitis pigmentosa with Metal L&F



Cataract Controls

Blur Strength 3

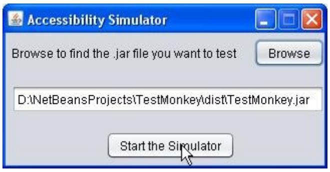
Disk Size 170

Yellowing 160

Fig. 6. RunVIS module simulating cataract with Windows L&F while having the "High Contrast White" appearance scheme activated

4.3 Vision Impairment Simulator Standalone Application

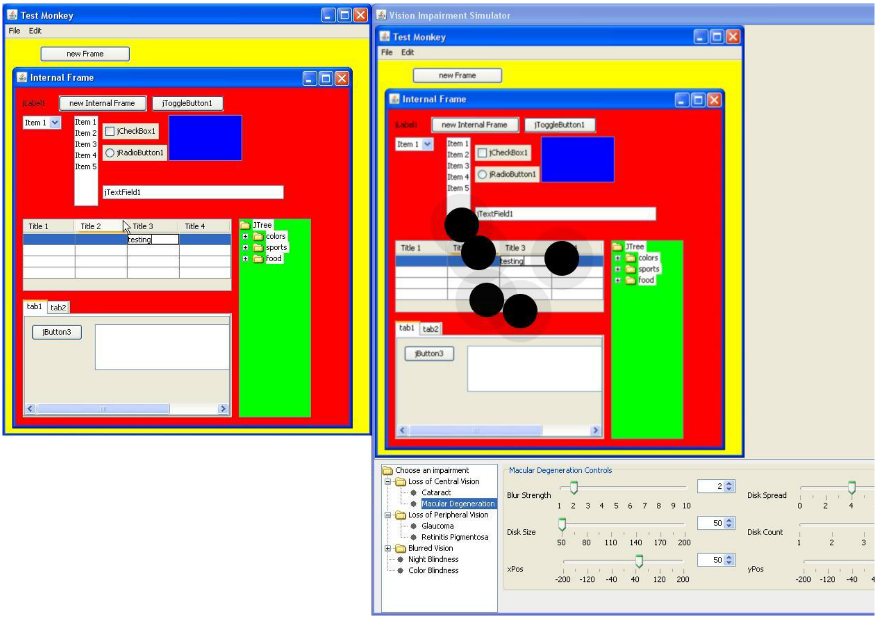
The aforementioned modules are a great help to the developer that give him the ability to quickly check any GUI form and eventually run the application and test its functionality while simulating a variety of vision impairments. These are all done while working with the Netbeans IDE. What happens when this IDE is not available to the developer or when someone else wants to evaluate the work done by the developer? In order to answer these questions we created a stand-alone version of the RunVIS module. With this application anyone, developer or not, can test any GUI application bundled in a jar file. As it is seen in Fig. 7 the user browses for a jar file of his choice and then starts the simulator. The interface is then similar to the RunVIS module and can be used for example to simulate a user with macular degeneration who types some text in a table's cell as can be seen in Fig. 8.



Browse to find the .jar file you want to test

D:\NetBeansProjects\TestMonkey\dist\TestMonkey.jar

Fig. 7. Vision impairment simulator standalone application initial state



Blur Strength: 2

Disk Size: 50

xPos: 50

yPos

Fig. 8. Vision impairment simulator standalone application simulating macular degeneration

5 Conclusions – Future Work

In this paper a tool for simulating various vision impairments in developing Java ™ swing applications is presented. This tool can be used as a part of the Netbeans IDE or as a standalone application, aiding the developers throughout the phases of the whole development process. This way they can overcome accessibility barriers and improve the overall quality of their applications. Feedback from user groups could be used to better simulate the implemented impairments or add more. Finally, the tool could be extended in order to apply the same simulation techniques to JavaFX [26] applications.

Acknowledgments. This work was partially funded by the EC FP7 project ACCESSIBLE - Accessibility Assessment Simulation Environment for New Applications Design and Development, Grant Agreement No. 224145.

References

1. http://www.w3.org
2. http://www.w3.org/WAI
3. http://www.sun.com
4. http://www.ibm.com
5. http://www.oasis-open.org
6. http://www.linux-foundation.org/en/Accessibility
7. http://www.w3.org/TR/aria-roadmap

8. http://java.sun.com/javase/technologies/accessibility/docs/jaccess-1.2/doc
9. http://www.linuxfoundation.org/en/Accessibility/IAccessible2
10. http://www.section508.gov
11. http://wave.webaim.org
12. http://aprompt.snow.utoronto.ca/overview.html
13. http://www.snook.ca/technical/colour_contrast/colour.html
14. http://www.paciellogroup.com/resources/contrast-analyser.html
15. http://gmazzocato.altervista.org/colorwheel/wheel.php
16. http://www.vischeck.com
17. http://vis.cita.uiuc.edu
18. http://www.webaim.org/simulations/lowvision.php
19. http://www.alphaworks.ibm.com
20. http://www.alphaworks.ibm.com/tech/adesigner
21. http://www.alphaworks.ibm.com/tech/raven
22. http://www.netbeans.org
23. http://a11y.netbeans.org
24. http://www.eclipse.org
25. http://en.wikipedia.org/wiki/Color_Blind
26. http://www.sun.com/software/javafx