Standard Corsi Test – Java Swing GUI Application Implementation

Tyler Schmidt

tschmidt@uccs.edu

Abstract

This is an example abstract. It goes here in 9 point Times New Roman type. Use of the Abstract Text style is required.

Introduction*[[1]](#footnote-1)*

There are invisible characters after the word Line in the heading above. If you delete them, the copyright line will disappear. The AAAI copyright notice is required. This is example text. It is 10 point Times New Roman. The first sentence after the heading *begins without a paragraph indent. Use of the text style (without alteration) is required.*

*This is example text. The second paragraph is indented 10 points, with no extra space between lines. The text is 10-point Times New Roman.*

Software Language

Java was chosen as the language to create this project in for a variety of reasons, but mostly for its built-in GUI elements under the Abstract Window Toolkit API. Java also features dependencies that can be easily added to any project to expand the capabilities of the software language.

*Java Swing*

*The text below a second-level heading begins without indentation. Use of the subsection heading style is required.*

*This is the second paragraph. It in formatted with the Text-indent style. Use of the Text-Indent style is required.*

*Simple JSON*

*The text below a second-level heading begins without indentation. Use of the subsection heading style is required.*

*This is the second paragraph. It in formatted with the Text-indent style. Use of the Text-Indent style is required.*

*Software Features*

*JFrame Menu System*

*The text below a second-level heading begins without indentation. Use of the subsection heading style is required.*

*This is the second paragraph. It in formatted with the Text-indent style. Use of the Text-Indent style is required.*

*JButtons as Blocks*

*The text below a second-level heading begins without indentation. Use of the subsection heading style is required.*

*This is the second paragraph. It in formatted with the Text-indent style. Use of the Text-Indent style is required.*

Saving & Loading

The text below a second-level heading begins without indentation. Use of the subsection heading style is required.

This is the second paragraph. It in formatted with the Text-indent style. Use of the Text-Indent style is required.

Standard Corsi Forward

The application is able to play the Standard Corsi test, both forward and in reverse. In forward mode, players are tasked to press the blocks in the exact order that the software presses them in. The game then needs to compare the “Computer’s” pressed button sequence and the player’s pressed button sequence and determine if the two sequences match. The way that the game accomplishes this is actually quite simple, every button that appears on the game screen is assigned a 1-9 button *index* that gets put into an array whenever the button is pressed. There are two arrays, one for the “computer player” and one for the “human player”. Once the “Done” button has been pressed, the game then comparers these two arrays to determine if they match or not, if the two arrays are the same, the players wins that round.

Standard Corsi Reversed

The game also features the ability to play the standard corsi test in reverse,

“Classic”, “Pause” & “Modern” Modes

The text below a second-level heading begins without indentation. Use of the subsection heading style is required.

This is the second paragraph. It in formatted with the Text-indent style. Use of the Text-Indent style is required.

Diagnosis

The text below a second-level heading begins without indentation. Use of the subsection heading style is required.

This is the second paragraph. It in formatted with the Text-indent style. Use of the Text-Indent style is required.

Anti-hacking Measures

Hacking and data privacy are some of the biggest concerns facing the software industry in recent years, and video game software is no exception to this. Hackers could potentially want to reverse engineer the software for a number a reasons including cheating, piracy, and gaining access to other people’s sensitive information. To prevent this as much as possible, many steps have been taken in the creation of this game to ensure security.

Encapsulation

Due to the object-oriented design of the Java software language, measures for encapsulating data and variables have already been defined. The game takes advantage of this to hide sensitive variables using encapsulation and private data fields within object classes.

Encryption

Due to the software featuring a saving & loading system that is able to store potentially sensitive data (i.e. address, diagnosis, etc.) the application must attempt to hide this data via encryption.

To achieve this, all data is encrypted with DES encryption before saving to the designated save file. Each player entry within the save file creates a unique encryption key to use with all data that is stored for that specific record. The encryption key is then stored on the file within that specific player record.

Obfuscation

The structure of the save file that is created by the software is a JSONArray of JSON Objects. Therefore, each “value” that is stored in the file has a corresponding variable name that matches to the value. This means that one could look at the outputted save file, and be able to read variable names and their values. These values are encrypted, but their corresponding names are not, meaning that a potential hacker would be able to read the names of the variables saved inside the file. One could simply look at the file and see a variable names “key” and could correctly assume that variable to be an encryption key, and unlock the ability to reverse engineer the software.

To combat this, variables that are written to the save file obfuscated, meaning their names have been changed in a way that gives no meaning for what that variable represents. For example, the variable that stores the encryption key for each player has been renamed to “nahjlo5lke” instead of “key”.

Results

This Is an Example of a Figure Caption.

You must have at least 6 points additional space between your figure or table and the figure or table caption. You must have at least 1 carriage return (12 points additional space) after the caption. You must also have at least 1 carriage return of space (12 points additional space) before a figure or table that does not begin at the top of a page. Figures and tables must not intrude into *any* margin or gutter (this includes boxes and rules and frames).

Conclusion

This Is an Example of a Figure Caption.

You must have at least 6 points additional space between your figure or table and the figure or table caption. You must have at least 1 carriage return (12 points additional space) after the caption. You must also have at least 1 carriage return of space (12 points additional space) before a figure or table that does not begin at the top of a page. Figures and tables must not intrude into *any* margin or gutter (this includes boxes and rules and frames).

References

Aaai.org. (2019). *Formatting Instructions Word*. [online] Available at: http://www.aaai.org/Publications/Author/formatting-instructions-word.pdf [Accessed 7 Oct. 2019].

Bodnar, J. (2019). *Java Swing first programs - JFrame, JPanel, JButton, GroupLayout, JLabel*. [online] Zetcode.com. Available at: http://zetcode.com/tutorials/javaswingtutorial/firstprograms/ [Accessed 7 Oct. 2019].

Gupta, L. (2019). *Java read json and write json example - JSON.simple tutorial*. [online] HowToDoInJava. Available at: https://howtodoinjava.com/library/json-simple-read-write-json-examples/ [Accessed 7 Oct. 2019].

H. (2019). *How to append data on a existing Json file using Java?* [online] Stack Overflow. Available at: https://stackoverflow.com/questions/50402343/how-to-append-data-on-a-existing-json-file-using-java [Accessed 7 Oct. 2019].

1. Copyright © 2019, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved. [↑](#footnote-ref-1)