## 

## 

**Experiment 4:**

**Design Patterns and Unit Testing**

CPE106L (Software Design Laboratory)

Group No.: **10**

Section: **B2**

## **PreLab**



|  |
| --- |
| **Readings, Insights, and Reflection** **<CANDA>**  In the book "Python Projects," developers may see how to use Python and the Model-View-Controller design pattern to construct desktop applications that are simple to administer and manage. On pages 162-173, the first variation, Tic Tac Toe (Console App), is discussed. After that, they go over how to develop a model which depicts the state of the game, a viewpoint that shows the game board, and a controller that controls how the game progresses. Line by line, with thorough descriptions of each action, is the application's code.  Moreover, the GUI app version, described on pages 186–193, demonstrates how to tie events to the widgets and uses the Tkinter library to generate the game's window and widgets. It also details well how to link occurrences, such as push buttons, to these panels. The authors offer a thorough description of the written rules along with how to employ the MVC architecture to execute these, much like they did for the console app version. Ultimately, the Tic Tac Toe projects are a perfect introduction to utilizing Python and the MVC framework to create application software. Developers can discover how to use Python to construct their own desktop apps, incorporating games and other kinds of programs, by completing the precise guidelines in the e-book.  Also, it is apparent when reading the given material that Laura Cassell and Alan Gauld's "Python Projects" is a reliable source of information for developing desktop apps utilizing Python and the Model-View-Controller (MVC) design. Both a console game and a GUI app for Tic Tac Toe are covered in the textbook, along with processed explanations regarding how to apply the MVC pattern in each form.  **<COLLAMAT>**  Upon reviewing the provided content, it becomes clear that "Python Projects" authored by Laura Cassell and Alan Gauld serves as an invaluable guide for individuals seeking to develop desktop applications utilizing Python alongside the Model-View-Controller (MVC) architectural pattern. The book meticulously explores the implementation of two iterations of the classic game Tic Tac Toe: one in a console-based format and the other featuring a graphical user interface (GUI). Throughout the text, readers are presented with detailed, step-by-step instructions elucidating the integration of the MVC pattern into each version.    In the sections spanning pages 162 to 173, the process of constructing a model, view, and controller for the console-based rendition of Tic Tac Toe is thoroughly elucidated. The authors' explanations are comprehensive, enabling developers of varying proficiencies to grasp the intricacies of each code segment and procedural step. Transitioning to the GUI rendition detailed from pages 186 to 193, the text demonstrates the linkage of events to user interface components, leveraging the Tkinter library to craft visually appealing game windows and widgets. Notably, this GUI rendition offers a more immersive and aesthetically pleasing gaming experience compared to its console counterpart. Furthermore, the authors furnish a comprehensive overview of the game's mechanics and the application of the MVC pattern within its framework.    Irrespective of one's proficiency level, "Python Projects" stands as a valuable resource for anyone aspiring to develop desktop applications using Python and the MVC paradigm. Novices can readily follow the book's systematic instructions and elucidations to craft their own desktop applications, while seasoned developers may utilize it as a reference to deepen their understanding. In summary, "Python Projects" represents an indispensable asset for individuals interested in desktop application development with Python and the MVC pattern, with the Tic Tac Toe projects within serving as exemplary illustrations of practical MVC implementation. |
| **<ESTACION>**  Python Projects' Chapter 4 is a remarkable example of the presentation of desktop application development in the form of two Tic Tac Toe games created in Python. Initially, the console based version is built using the Model-Controller-View (MCV) architecture, which helps separate concerns for better manageability and modularity. Here, the focus is on three main layers: the data layer for the storing and retrieving of game data, a core logic layer that handles game algorithms and states, and a user interface layer, developed with the Python cmd module, that allows users to interact with the game. This approach highlights the strong management system required for a text-based game and emphasizes the need for efficiency in command execution for the console programs.    In contrast, the second version of the game incorporates a graphical user interface using Tkinter, the standard GUI toolkit in Python, which provides an easy-to-use development experience and a much better user interface thanks to its improved visual appearance. This GUI version adopts a simplified MCV pattern which is designed specifically for graphical interaction and, therefore, some view and controller responsibilities are integrated into it, thus it provides a better way for the users to interact with the application. The chapter 4 not only emphasizes the Python’s flexibility in building applications with different complexities but also brings the information on how to choose the right architecture that is suitable for the application and the intended experience of the user. This duality in programming design brings a deep educational insight into software development, which is centered on practical skills that can be applied to any software project beyond game development.  **<TRINIDAD>**  The book “Python Projects” shows readers how to use Model-View-Controller (MVC) design and Python itself. The book also shows you how to construct eh MVC pattern in Python for both Tic Tac Toe game’s versions. One being a console app and the other as a GUI app. In Page 162, it describes the rules of the game and walk through the process of creating a MVC. In page 186, it describes how to connect events to the game’s widgets and create the game’s window using the Tkinter toolkit. Overall, regardless of expertise level, "Python Projects" is an invaluable tool for learning how to create desktop apps using Python and the MVC framework.  For individuals who wish to learn how to create desktop applications with Python and the MVC design, the Tic Tac Toe project included in "Python Projects" is an excellent resource. "Python Projects" is a helpful manual for developing many kinds of desktop apps with Python, like games, regardless of your level of programming knowledge. Python developers can improve their skills and make their own desktop programs with the book's comprehensive, step-by-step instructions. |
|  |