EMERTXE TRAINING PROJECT DOCUMENTATION FRAMEWORK

REQUIREMENTS & DESIGN DOCUMENT

Module – Python Programming

Tic Tac Toe

EMERTXE

Contents

1 Abstract

Tic-tac-toe is a noughts and crosses or Xs and Os paper-and-pencil game for two players. Two players representing X and O respectively will take turns marking the spaces in a 3×3 grid. The player who succeeds in placing three of their marks in a diagonal, horizontal, or vertical row is the winner. It is a solved game with a forced draw assuming best play from both players. The aim of this project is to implement the Tic-tac-toe game application using Python. The game logic can be built using core python, and Tkinter will be used to develop the application's Graphical User Interface (GUI).

The app can be built in two phases. The first one being a standalone game played in a computer and another one being a network based game played between different computers in the same network. Requirements of both is given below.

2 Requirements

Here are the requirements for standalone and network versions of the game.

The standalone app can be played by 2 humans on the same computer

- Each player can take turns to play the game
- Player can click on the grid to make a move
- One player gets X and other player O
- On each move, the game status is checked
- If a player wins, the game ends and the winner status is displayed
- The game can also end in a draw
- After the game ends, a new game can start

The Features of Network App

- The server app listens for and accepts clients connections.
- The server mainly acts as a gateway.
- It will receive player info, and game moves and forward the details among opponents.
- Design of the client server protocol is a key aspect of developing this project.
- Two separate apps will be developed -Client & Server app
- The client app provides the user interface for connecting to server and playing the game

3 Sample Output

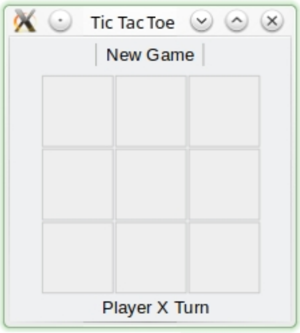


Fig 3 1: The GUI when the game is launched

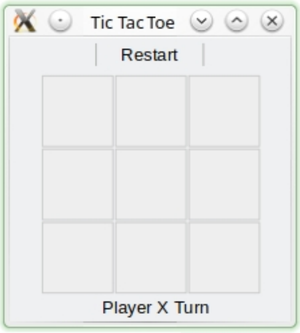


Fig 3 2: When the new game is started, Observe players turn notification

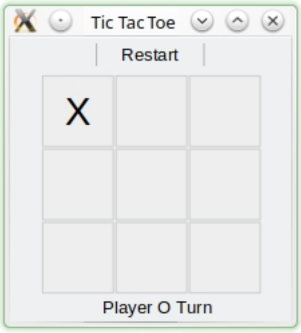
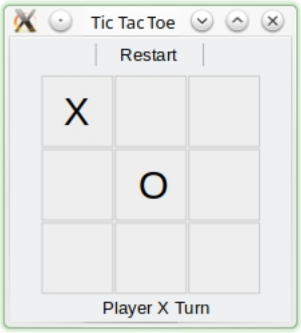


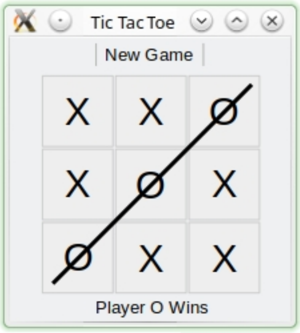
Fig 3 3: Player O turn



X		
	O	

Player X Turn

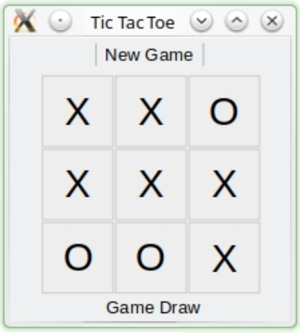
Fig 3 4: Player X turn



X	X	O
X	O	X
O	X	X

Player O Wins

Fig 3 5: Player X wins the game



X	X	O
X	X	X
O	O	X

Game Draw

Fig 3 6: Game draw between both players

4 Artifacts

4.1 References

- Qt - Automotive Suite
- Qt for Embedded Systems
- Qt based GUI for Raspberry Pi