**Lab 11- Develop Tic Tac Toe Game using Unity UI components**

1. Open **Unity Hub** and create a **new 2D project**.
2. In the **Hierarchy**, right-click and go to **UI > Panel** (this will act as a container for the game board). Rename it to **GameBoard**.
3. Set the **Rect Transform** so that it covers the screen properly.
4. Create a **Grid Layout** for the game board:- Select **GameBoard**. In the **Inspector**, click **Add Component**. Search for **Grid Layout Group** and add it. Set the **Cell Size** to (100, 100) and **Spacing** to (5,5). Set the **Child Alignment** to **Middle Center**. Check **Control Child Size** (Width & Height).
5. Right-click **GameBoard**, go to **UI > Button**. Rename it as **Cell\_0** (for the first cell). Resize it to (100x100). Change the button text to **""** (empty). **Duplicate** (Ctrl + D) the button **8 times** to create 9 cells. Arrange them inside the **GameBoard**. This is your Tic-Tac-Tac-Toe Game board.
6. Right-click in the **Hierarchy** > **UI > Text - TextMeshPro**. Rename it to **GameStatusText**. Position it at the top of the screen. Change the **Text** property to "Player X's Turn".
7. Right-click in the **Hierarchy** > **UI > Button**. Rename it **ResetButton**. Adjust its size and position. Change its **Text** to "Restart".
8. In the **Project** panel, create a new folder called **Scripts**. Inside the **Scripts** folder, right-click and select **Create > C# Script**. Name it **GameManager**. Open the script and replace the content with:

*using UnityEngine;*

*using UnityEngine.UI;*

*using TMPro;*

*public class GameManager : MonoBehaviour*

*{*

*public Button[] cells; // Array to hold the game cells*

*public TextMeshProUGUI gameStatusText;*

*public Button resetButton;*

*private string currentPlayer = "X";*

*private string[] board = new string[9]; // Represents the board state*

*private bool gameActive = true;*

*void Start()*

*{*

*resetButton.onClick.AddListener(ResetGame);*

*for (int i = 0; i < cells.Length; i++)*

*{*

*int index = i;*

*cells[i].onClick.AddListener(() => OnCellClicked(index));*

*board[i] = "";*

*}*

*}*

*void OnCellClicked(int index)*

*{*

*if (!gameActive || !string.IsNullOrEmpty(board[index]))*

*return;*

*board[index] = currentPlayer;*

*cells[index].GetComponentInChildren<TextMeshProUGUI>().text = currentPlayer;*

*if (CheckWin())*

*{*

*gameStatusText.text = currentPlayer + " Wins!";*

*gameActive = false;*

*return;*

*}*

*if (CheckDraw())*

*{*

*gameStatusText.text = "It's a Draw!";*

*gameActive = false;*

*return;*

*}*

*currentPlayer = (currentPlayer == "X") ? "O" : "X";*

*gameStatusText.text = currentPlayer + "'s Turn";*

*}*

*bool CheckWin()*

*{*

*int[,] winPatterns = {*

*{ 0, 1, 2 }, { 3, 4, 5 }, { 6, 7, 8 }, // Rows*

*{ 0, 3, 6 }, { 1, 4, 7 }, { 2, 5, 8 }, // Columns*

*{ 0, 4, 8 }, { 2, 4, 6 } // Diagonals*

*};*

*for (int i = 0; i < winPatterns.GetLength(0); i++)*

*{*

*int a = winPatterns[i, 0], b = winPatterns[i, 1], c = winPatterns[i, 2];*

*if (board[a] != "" && board[a] == board[b] && board[a] == board[c])*

*{*

*return true;*

*}*

*}*

*return false;*

*}*

*bool CheckDraw()*

*{*

*foreach (string cell in board)*

*{*

*if (cell == "") return false;*

*}*

*return true;*

*}*

*void ResetGame()*

*{*

*currentPlayer = "X";*

*gameStatusText.text = "Player X's Turn";*

*gameActive = true;*

*for (int i = 0; i < cells.Length; i++)*

*{*

*board[i] = "";*

*cells[i].GetComponentInChildren<TextMeshProUGUI>().text = "";*

*}*

*}*

*}*

1. Create an **Empty GameObject** in the Hierarchy. Rename it to **GameManager**. Attach the **GameManager** script to it.
2. Assign UI elements in the **Inspector**-Drag **all 9 buttons** into the **cells** array. Assign the **GameStatusText** to the **gameStatusText** field. Assign the **ResetButton** to the **resetButton** field.
3. Play test.