

# **INFINI Tic-Tac-Toe**

Module 3 Final Project



**In partial fulfillment of the requirements in  
Computer Programming Concepts 2 (IT101-2)**

**Faller, Amiel Ed Angelo  
Catienza, Michel Eirene I.**

Bachelor of Science in Entertainment and Multimedia Computing

**Engr. Martzel P. Baste**  
Course Instructor

June 2024

# Introduction

## Problem Scenario:

The youth of today are known to always be surrounded by cell phones and television screens which makes them the generation that is most exposed to digital media. As times are changing, their reliance on devices, media and games to gather knowledge is very high. With the many advantages digital media gives us, many are still unaware or remain oblivious to their excessive consumption of energy and media. In order to help in increasing their knowledge and raise awareness, implementing societal issues in certain media and games have become the norm.

## Sustainable Development Goals (SDGs)

The game project can raise awareness to the following SDGs:

- SDG 12 - Responsible Consumption and Production  
This game project can raise awareness regarding the consumption and conservation of electricity, with the game's concept of having unlimited moves but limited marks (represented by light bulbs) acting as a means to limit energy consumption.
- SDG 13 - Climate Action  
Limiting the consumption of electricity can aid in the efforts against climate change by reducing the need for the production of energy and reducing light pollution.

## Project Objectives

The game project aims to increase the replayability of Tic-Tac-Toe while staying true to the classic game mechanics and implementing a modern twist. Furthermore, the visuals and GUI of the application will be utilized as a medium of raising awareness to various issues being experienced by the world today.

## Target Market

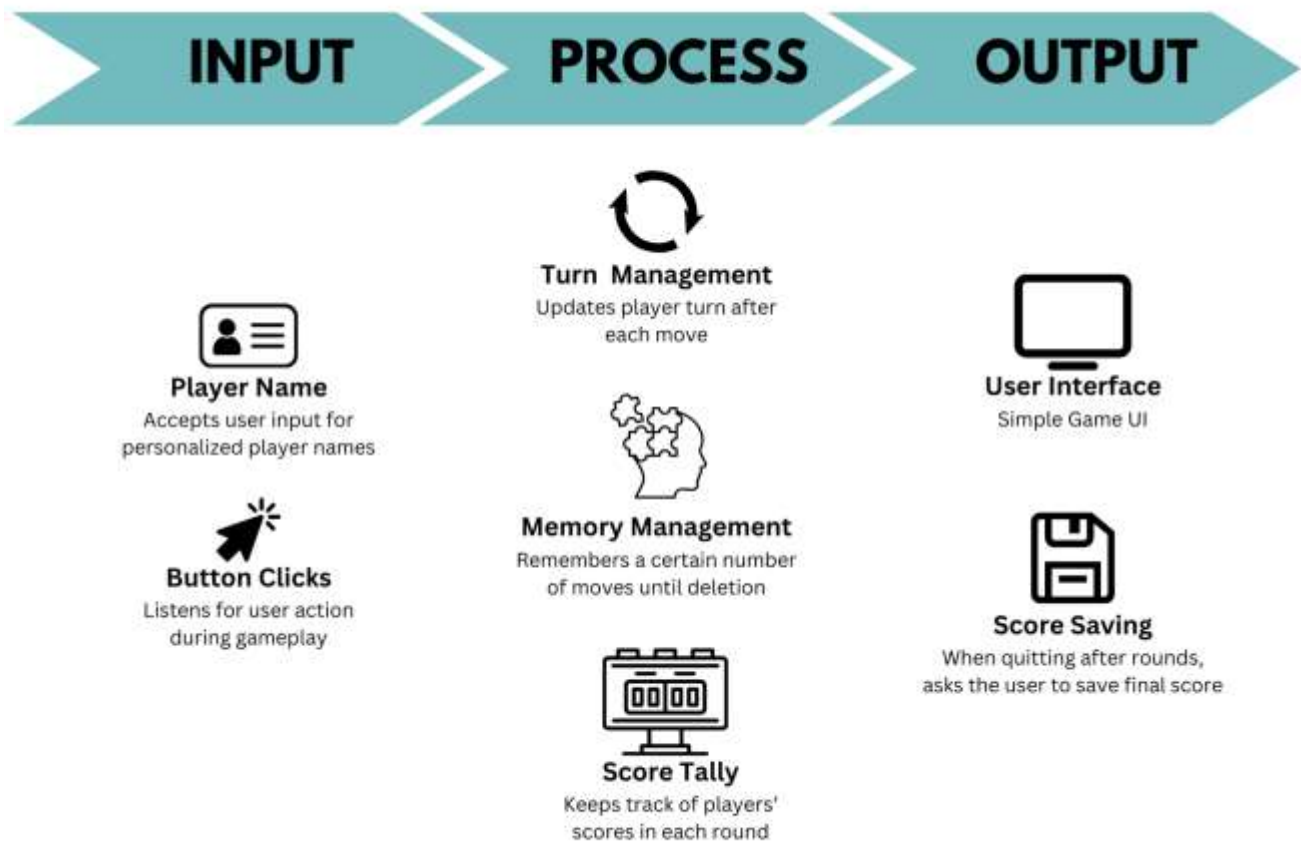
Based on the classic Tic-Tac-Toe game, INFINI Tic-Tac-Toe can be played by people as young as 8 years old and above.

## Similar Application

Many versions of Tic-Tac-Toe exist on the internet, many of which also have an infinite feature which has been implemented in different ways. Some include bigger play fields similar to Connect Four, others give infinite amounts of game resets. In order to differentiate from these similar applications, a version of infinite Tic-Tac-Toe was created that keeps its 3x3 playing field while implementing a game loop that can last longer than the classic game.

## Conceptual

## Framework



## Scope and Limitations

The requirements for the game project are rather low, meaning most if not all computers can enjoy the gameplay of INFINI Tic-Tac-Toe including computers with low specs. While the project is a simple game with a simple UI, the game project features a score saving system, its core mechanic of deleting old moves after a certain number of turns, and a score tally that continues when the users wish to play again. However, the game lacks online multiplayer support and the choice to play single player, as a bot or AI would be difficult to implement. Originally, there was also supposed to be more visuals to be added to the game, but due to time constraints and lack of experience with mixing panels, labels, buttons, and images into one frame, we were unable to do so.

## Project Definition

This game project primarily utilizes the Java programming language including its libraries, packages, and methods. Among these include:

**Java Abstract Window Toolkit (AWT).** The game project utilizes the `java.awt` import to create some of the UI of the game panel. This also was used to call event handlers such as `ActionListener` in order to check whether or not the buttons have been pressed, and what action to take when pressed.

**Java.util Library.** `Java.util` is the standard library for Java. In the game project, it was used for the memory aspect of the code, specifically with `ArrayList`, in order to remember the users' button input at a certain threshold before deleting it to keep the game going.

**Java Swing Library.** Used to also create the UI of the application, with the assistance of the Java AWT library.

**Javax.io Library.** Used for File CRUD, however, only the file save option was used in this project.

**Game Assets.** The assets used for the game application were created with Canva and Piksel.

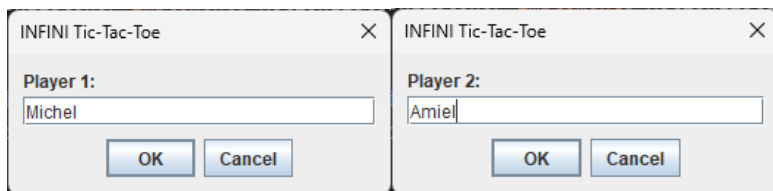
## Project Solution

INFINI Tic-Tac-Toe is a game that introduces a layer of strategy that goes beyond the one-round experience of classic Tic-Tac-Toe. Its infinite nature limits the players' ability to offensive and defensive strategies, allowing for longer playtime and an endurance of wits.

## System Prototyping

### *Player Names*

The program will begin by asking the names of the two players, which will be used later for the score tally, save file, and player name display..



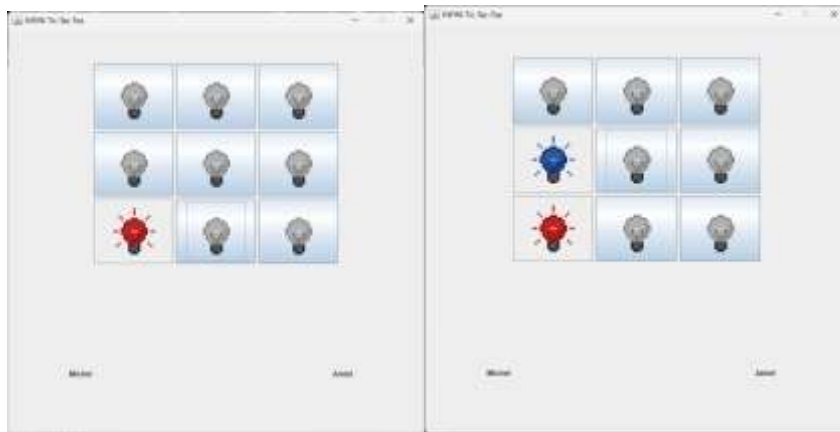
### *Play Area - Start Game*

Directly after pressing "OK", the game will begin at this screen. This screen consists of the standard 3x3 Tic-Tac-Toe grid, and two JPanels that display the players' names.



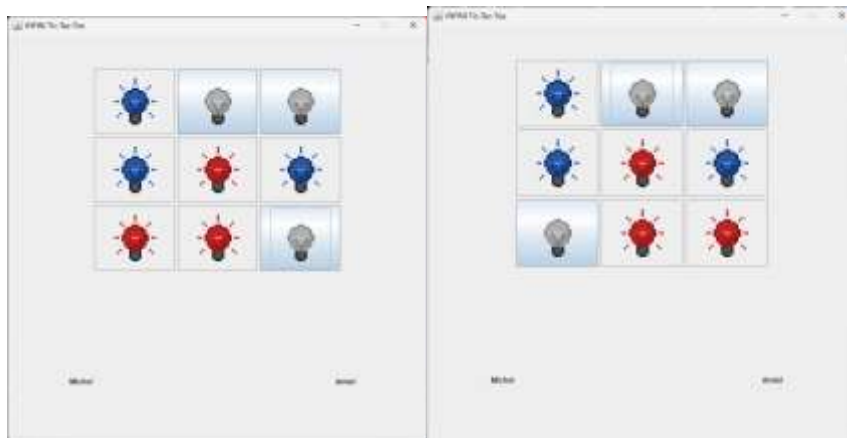
### *Play Area - Player Moves*

Each time a player presses any of the buttons, the buttons' display will change by order of players, which is represented by a Red light bulb and Blue light bulb in that particular order.



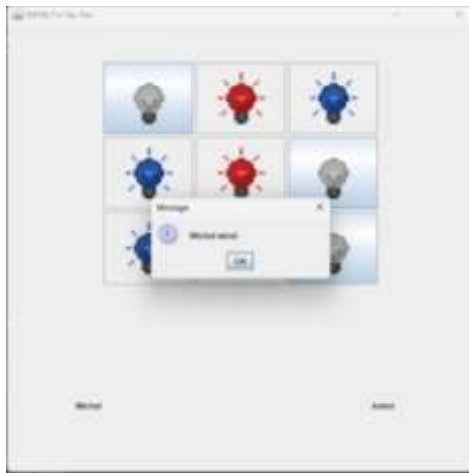
### ***Play Area - Infinity Mechanic***

Once both players have played all three of their bulbs on the board, their next move will now delete their oldest recorded move and replace it with a turned off light bulb.



### ***Winning Screen***

Once a player is determined by the code, a dialog box will appear on screen to confirm that someone has won the game.



### ***Continue Screen***

The next screen after the winning screen will give the player the option to continue their game or to exit. By pressing "Yes", the score tally will be added by one based on whoever won that round, and the grid will clear all light bulbs.



### ***Save File Screen***

By pressing “No” in the continue screen, the players will be redirected here. In this screen, the total tallied score will be displayed and saved with the file name of the users’ choosing. The saved file will be situated at the src folder as a .txt file.



## Source Code



**Used Libraries.** Imports necessary libraries

```
1  import javax.swing.*;
2  import java.awt.*;
3  import java.awt.event.*;
4  import java.io.*;
5  import java.util.ArrayList;
```

**Class and Member variables.** List of the variables used for the code.

```
7  ▶ public class tictactoe {
8      private JFrame frame; 17 usages
9      private JPanel gamePanel; 5 usages
10     private JPanel player1Panel; 6 usages
11     private JPanel player2Panel; 6 usages
12     private JButton[] buttons = new JButton[9]; 41 usages
13     private boolean xTurn = true; 4 usages
14     private ArrayList<Integer> memory= new ArrayList<>(); 5 usages
15
16     private String p1, p2, score; 8 usages
17     private int scoreP1 = 0, scoreP2 = 0; 6 usages
18
19     ImageIcon dark = new ImageIcon(getClass().getResource( name: "assets/LIGHT_OFF.png")); 2 usage
20     ImageIcon p1 = new ImageIcon(getClass().getResource( name: "assets/LIGHT_RED.png")); 2 usages
21     ImageIcon p2 = new ImageIcon(getClass().getResource( name: "assets/LIGHT_BLUE.png")); 2 usages
```

**Game UI.** This generates all the buttons, panels, labels, frame, pane, and the images used for the game.

```

30 public class Game {
31     // Create the JFrame
32     JFrame frame = new JFrame("INFINI Tic-Tac-Toe");
33     frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
34     frame.setResizable(false);
35     frame.setSize(800, 700);
36     frame.setLocationRelativeTo(null);
37
38     // Create the JPanel
39     JPanel gamePanel = new JPanel();
40     gamePanel.setSize(800, 700);
41     gamePanel.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10));
42
43     // Create the JButton array
44     for (int i = 0; i < 9; i++) {
45         JButton button = new JButton(""); // set to blank
46         button.setSize(new Dimension(100, 100));
47         button.setFont(new Font("Arial", Font.PLAIN, 40));
48         button.addActionListener(new ButtonClickListener());
49         gamePanel.add(button[i]);
50     }
51
52     // Create the JOptionPane
53     JOptionPane p1 = JOptionPane.showInputDialog(frame, "Player 1", "INFINI Tic-Tac-Toe", JOptionPane.PLAIN_MESSAGE);
54     JOptionPane p2 = JOptionPane.showInputDialog(frame, "Player 2", "INFINI Tic-Tac-Toe", JOptionPane.PLAIN_MESSAGE);
55
56     // Create the JLabel
57     JLabel player1Label = new JLabel("Player 1");
58     player1Label.setSize(100, 50);
59     player1Label.setFont(new Font("Arial", Font.PLAIN, 20));
60     player1Label.add(new JLabel(p1));
61     player1Label.setLocation(20, 530);
62
63     // Create the JLabel
64     JLabel player2Label = new JLabel("Player 2");
65     player2Label.setSize(100, 50);
66     player2Label.setFont(new Font("Arial", Font.PLAIN, 20));
67     player2Label.add(new JLabel(p2));
68     player2Label.setLocation(20, 580);
69
70     // Add the JLabel to the JFrame
71     frame.add(player1Label);
72     frame.add(player2Label);
73     frame.add(gamePanel, BorderLayout.CENTER);
74     frame.setVisible(true);
75 }

```

**checkForWinner() Method.** This method checks each row, diagonal, and column for a completed set of bulbs in order to determine a winner.

```

64 public void checkForWinner() {
65     // Check rows
66     for (int i = 0; i < 9; i += 3) {
67         if (buttons[i].getText().equals(buttons[i+1].getText()) && buttons[i].getText().equals(buttons[i+2].getText()) && !buttons[i].isEnabled()) {
68             if (buttons[i].getText().equals("X")) {
69                 JOptionPane.showMessageDialog(frame, "Player 1 wins!");
70                 scoreP1++;
71             } else if (buttons[i].getText().equals("O")) {
72                 JOptionPane.showMessageDialog(frame, "Player 2 wins!");
73                 scoreP2++;
74             }
75             resetGame();
76             return;
77         }
78     }
79
80     // Check columns
81     for (int i = 0; i < 3; i++) {
82         if (buttons[i].getText().equals(buttons[i+3].getText()) && buttons[i].getText().equals(buttons[i+6].getText()) && !buttons[i].isEnabled()) {
83             if (buttons[i].getText().equals("X")) {
84                 JOptionPane.showMessageDialog(frame, "Player 1 wins!");
85                 scoreP1++;
86             } else if (buttons[i].getText().equals("O")) {
87                 JOptionPane.showMessageDialog(frame, "Player 2 wins!");
88                 scoreP2++;
89             }
90             resetGame();
91             return;
92         }
93     }
94 }

```

```

77 // Check diagonals
78 if (buttons[0].getText().equals(buttons[4].getText()) && buttons[0].getText().equals(buttons[8].getText()) && !buttons[0].isEnabled()) {
79     if (buttons[0].getText() == "X") {
100         JOptionPane.showMessageDialog(Frame, message: P1 + " wins!");
101         scoreP1++;
102     } else if (buttons[0].getText() == "O") {
103         JOptionPane.showMessageDialog(Frame, message: P2 + " wins!");
104         scoreP2++;
105     }
106     resetGame();
107     return;
108 }
109 if (buttons[2].getText().equals(buttons[4].getText()) && buttons[2].getText().equals(buttons[6].getText()) && !buttons[2].isEnabled()) {
110     if (buttons[2].getText() == "X") {
111         JOptionPane.showMessageDialog(Frame, message: P1 + " wins!");
112         scoreP1++;
113     } else if (buttons[2].getText() == "O") {
114         JOptionPane.showMessageDialog(Frame, message: P2 + " wins!");
115         scoreP2++;
116     }
117     resetGame();
118     return;
119 }
120 }
121 }

```

***resetGame() Method.*** This method contains the last few screens of the project, which are the Continue screen and Save File screen.

```

122 public void resetGame() {
123     int result = JOptionPane.showConfirmDialog(ContinuationScreen, "Continue?", "Continue?", JOptionPane.YES_NO_OPTION);
124     if (result == JOptionPane.YES_OPTION) {
125         int turn = new Integer(getLast().getMove()).intValue().equals("X") ? 1 : 2;
126         for (int i = 0; i < 9; i++) {
127             buttons[i].setText("");
128             buttons[i].setEnabled(true);
129             buttons[i].setBackground(null);
130             buttons[i].setFocusable(true);
131             memory.clear();
132         }
133         turn = turn;
134         return;
135     } else {
136         int score = new Integer(scoreP1 + scoreP2);
137         score = JOptionPane.showMessageDialog(ContinuationScreen, "Score: " + P1 + " = " + scoreP1 + " + " + scoreP2 + " = " + score + "\n" +
138             "P1 = " + " " + scoreP1 + " + " + scoreP2 + " = " + scoreP1 + " + " + scoreP2 + " = " + scoreP2);
139         File file = new File(score);
140         try {
141             FileWriter fw = new FileWriter(file);
142             fw.write("Score: " + P1 + " = " + scoreP1 + " + " + scoreP2 + " = " + scoreP1 + " + " + scoreP2 + " = " + scoreP2);
143             fw.close();
144         } catch (IOException e) {
145             e.printStackTrace();
146         }
147         return;
148     }
149 }
150 }

```

***ButtonClickListener Class and Memory mechanic.*** This class handles the button click events in the game project, determines when to switch turns, and contains the Memory part of the code that stores the buttons clicked and deletes the oldest moves after 6 moves

in total have been made.

```
153 private class ButtonClickListener implements ActionListener { 1 usage
154     private int index; 2 usages
155
156     public ButtonClickListener(int index) { 1 usage
157         this.index = index;
158     }
159
160
161     @Override
162     public void actionPerformed(ActionEvent e) {
163         // Handle button click
164         JButton button = (JButton) e.getSource();
165
166         // Change the appearance of the button depending on the current turn.
167         if (xTurn) {
168             button.setText("X");
169             button.setDisabledIcon(p1);
170             button.setIcon(p1);
171             button.setHorizontalTextPosition(JButton.CENTER);
172             button.setVerticalTextPosition(JButton.CENTER);
173         } else {
174             button.setText("O");
175             button.setDisabledIcon(p2);
176             button.setIcon(p2);
177             button.setHorizontalTextPosition(JButton.CENTER);
178             button.setVerticalTextPosition(JButton.CENTER);
179         }
180         button.setEnabled(false);
181         xTurn = !xTurn;
182
183         if (memory.size() >= 6) {
184             int first = memory.getFirst();
185             buttons[first].setText("");
186             buttons[first].setEnabled(true);
187             buttons[first].setBackground(null);
188             buttons[first].setIcon(dark);
189             memory.removeFirst();
190         }
191         memory.add(index);
192
193         // Check the winner ONLY after modifying button state memory.
194         checkForWinner();
195     }
196 }
```

## APPENDIX

## References

<https://unstats.un.org/sdgs/report/2023/>

<https://unstats.un.org/sdgs/report/2023/Goal-12/>

<https://unstats.un.org/sdgs/report/2023/Goal-13/>

<https://codewithcurious.com/projects/tic-tac-toe-game-using-java-swing/>

## CURRICULUM VITAE

### PERSONAL INFORMATION

Name: Michel Eirene I. Catienza  
Birthday: March 15, 2003  
Age: 21  
Program: CCIS – BS EMC  
Section: A125  
Address: Doña Carmen St., Dacoville Subd., Dumoy, Davao City  
Contact: 0916 755 7363  
Email: [catienzamicheleirene@gmail.com](mailto:catienzamicheleirene@gmail.com) | [meCatienza@mcm.edu.ph](mailto:meCatienza@mcm.edu.ph)



## EDUCATIONAL BACKGROUND

College: Mapua Malayan Colleges Mindanao

Senior High School: Mapua Malayan Colleges Mindanao

Junior High School: Seven Seas Academy

Elementary: Seven Seas Academy; Christian Values School; Concepcion Elementary School

## KEY SKILLS

- Java Programming
- C++ Programming
- Python Programming
- Story Writing
- Animation

## CURRICULUM VITAE

### PERSONAL INFORMATION

Name: Amiel Ed Angelo A. Faller

Birthday: November 13, 2000

Age: 23

Program: CCIS – BS EMC

Section: A125

Address: Blk.1 Lot 4 Garden Villas, Elenita Heights Subd., Cat. Grande, Davao City

Contact: 09995413057

Email: [amieledangelo.faller@gmail.com](mailto:amieledangelo.faller@gmail.com) | [aeafaller@mcm.edu.ph](mailto:aeafaller@mcm.edu.ph)



## **EDUCATIONAL BACKGROUND**

College: Mapua Malayan Colleges Mindanao

Senior High School: Ateneo de Davao University

Junior High School: Ateneo de Davao University

Elementary: Ateneo de Davao University

## **KEY SKILLS**

- Java Programming
- C++ Programming
- Python Programming
- 3D Modeling
- Photoshop

## **Documentation**

## The Sustainable Development Goals Report 2023: Special Edition



**2023**  
The Sustainable Development Goals Report  
Special edition

**10 July 2023:** The Sustainable Development Goals Report 2023: Special Edition provides a powerful call to action, presenting a candid assessment of the SDGs based on the latest data and estimates. While highlighting the existing gaps and urging the world to redouble its efforts, the report also emphasizes the immense potential for success through strong political will and the utilization of available technologies, resources, and knowledge. Together, the global community can re-gate progress towards achieving the SDGs and create a brighter future for all.

According to the report, the impacts of the climate crisis, the war in Ukraine, a weak global economy, and the lingering effects of the COVID-19 pandemic have revealed weaknesses and hindered progress towards the Goals. The report further warns that while lack of progress is universal, it is the world's poorest and most vulnerable who are experiencing the worst effects of these unprecedented global challenges. It also points out areas that need urgent action to rescue the SDGs and deliver meaningful progress for people and the planet by 2030.

**Download Report:** [العربية](#) | [中文](#) | [English](#) | [Français](#) | [Русский](#) | [Español](#)

**Related Links**

[SDG Report 2023](#)

[Explore the Report](#)

[Key Messages](#)

[Fact Sheets](#)

[The Gender Snapshot 2023](#)

[Global Database](#)

[Progress Chart 2023](#)

[Progress Chart Technical Note](#)

[SDG Extended Report](#)

[UN Data Commons for the SDGs](#)



### Responsible consumption and production

- The material footprint per capita in high-income countries is 10 times the level of low-income countries. The world is also seriously off track in its efforts to halve per capita food waste and losses by 2030.
- Global crises triggered a resurgence in fossil fuel subsidies, nearly doubling from 2020 to 2021.
- Reporting has increased on corporate sustainability and on public procurement policies, but has fallen when it comes to sustainable consumption and monitoring sustainable tourism.
- Responsible consumption and production must be integral to recovery from the pandemic and to acceleration plans of the Sustainable Development Goals. It is crucial to implement



Workers sort plastics at a recycling plant in Côte d'Ivoire. Plastic pollution threatens the coastal zone, tourism, fishing and tourism. Overhauling the economic model governing plastics is essential to address this crisis.

© UNEP/Divulsi Group

policies that support a shift towards sustainable practices and decouple economic growth from resource use.

#### Regional inequalities in material footprints highlight consumption disparities

Between 2000 and 2019, global domestic material consumption—the amount of raw materials directly used for production processes in a country—increased by 66 per

cent. Excess of domestic material consumption over material footprint, 2019 (percentage)





## Climate action

- With a climate cataclysm looming, the pace and scale of current climate action plans are wholly insufficient to effectively tackle climate change. Increasingly frequent and intense extreme weather events are already impacting every region on Earth. Rising temperatures will escalate these hazards further, posing grave risks.
- The Intergovernmental Panel on Climate Change (IPCC) emphasises that deep, rapid and sustained reductions in greenhouse gas (GHG) emissions are essential in all sectors, beginning now and continuing throughout this decade. To limit global warming to 1.5°C above pre-industrial levels, emissions must already be decreasing and need to be cut by almost half by 2030, just seven years away.
- Urgent and transformative action is critical, going beyond mere plans and promises. It requires raising ambition, covering



Two men wade through flood water in Pakistan after torrential monsoon rains dumped up to five times the 30-year average rainfall. Such events are now more likely and more severe due to climate change.

© UNICEF/Pavel Zisk

entire economies and moving towards climate-resilient development, while outlining a clear path to achieve net-zero emissions. Time is running out, and immediate measures are necessary to avoid catastrophic consequences and secure a sustainable future for generations to come.

### Urgent global greenhouse gas emission reductions are needed to avert 1.5°C tipping point

The latest IPCC synthesis report unequivocally states that human activities

world will be for current and future generations will depend on the choices we make.



Home Projects Books Blog Interview Us

Browse Handwritten Notes

#### Buy Handwritten Notes



Complete CSS Handwritten Notes

Java Projects, projects by Dr. Othman

## Tic Tac Toe Game Using Java Swing



### Introduction:

In this article, we will demonstrate how to build a Tic Tac Toe Game using Java Swing library. This Tic Tac Toe Game allows you to play a game easily using a graphical user interface (GUI) application. This game is developed with the Java Swing library which offers a set of components that can be used to build a graphical user interface (GUI). This game has an easy and simple design with buttons to play the game. To play a game, you can click on the buttons and start playing the Tic Tac Toe Game.

### Explanation:

This Tic Tac Toe Game is a graphical user interface (GUI) game that allows you to easily play the game using buttons. You can easily quickly play the game using the system's simple design. The Tic Tac Toe Game has a simple design with buttons. The player can click on the buttons to start playing the game.

#### More Java Projects

- Calendar Application using Java Swing
- Flipped Tic Tac Toe App Using Java Swing
- Calculator - Text Based Note Taking Application
- Library Management System Using Java Swing
- Flight Clerk Using Java Swing
- Currency Converter Using Java Swing
- Report Card Using Java Swing
- Hangman Game Using Java Swing
- Signin Clerk Using Java Swing