

**Tribhuvan University**

**Faculty of Computer Science and Information Technology**

**A PROJECT PROPOSAL ON “TIC-TAC-TOE GAME USING MINIMAX ALGORITHM”**

**Submitted To:**

**Mr. Nishant Phuyal**

Lecturer, Data Analysis Algorithm

Department of CSIT of St. Lawrence college

Chabahil, Kathmandu

**In partial fulfillment of the requirement for the Bachelor in Computer Science and Information Technology**

**Submitted by:**

**Kushal Nepal**

BSC.CSIT 078 Batch, 5th Semester

**Abstract**

This proposal outlines the development process and outcomes of a Tic-Tac-Toe game, which I am working on as part of the practical implementation of the algorithms studied. The game uses the Minimax algorithm to provide an optimal strategy for the AI opponent, ensuring a challenging experience for the player. Tic-Tac-Toe is a classic strategy game that helps in understanding and applying game theory and artificial intelligence concepts.

Keywords: HTML, CSS, JavaScript, Minimax Algorithm, AI, Tic-Tac-Toe

Table of Contents

[**Chapter 1: Introduction** 1](#_Toc170851329)

[1.1. Introduction 1](#_Toc170851330)

[1.2. Problem statement 1](#_Toc170851331)

[1.3. Objective 2](#_Toc170851332)

[1.4. Scope 2](#_Toc170851333)

[**CHAPTER 2: LITERATURE REVIEW** 2](#_Toc170851334)

[**CHAPTER 3: METHODOLOGY** 3](#_Toc170851335)

3.[1. **Planning and Requirements Gathering** 3](#_Toc170851336)

3.[**2.** **Game Design** 3](#_Toc170851337)

3.[**3.** **Development** 3](#_Toc170851338)

3.[**4.** **Testing** 4](#_Toc170851339)

3.[5. **Feedback and Iteration** 4](#_Toc170851340)

3.[6. **Finalization and Deployment** 4](#_Toc170851341)

3.[7. **Post-Deployment Support** 4](#_Toc170851342)

3.[**8.** **Tools and Technologies** 5](#_Toc170851343)

[**Chapter 4: Feasibility study** 5](#_Toc170851344)

[4.1 Technical Feasibility 5](#_Toc170851345)

[4.2. Economic Feasibility 5](#_Toc170851346)

[4.3. Operational Feasibility 6](#_Toc170851347)

[4.4. Schedule Feasibility 6](#_Toc170851348)

[**Chapter 5: System Analysis and Design** 6](#_Toc170851349)

[5.1 Requirement Analysis 6](#_Toc170851350)

[5.2. Data modeling 7](#_Toc170851351)

[**Chapter 6: Implementation** 7](#_Toc170851352)

[6.1 Tools used (CASE Tools, Programming Languages, Database platform). 7](#_Toc170851353)

[6.2 Development plan: 8](#_Toc170851354)

[**Chapter 7: Conclusion** 9](#_Toc170851355)

**List of figures**

[**Figure 1: flowchart for a maze game** 7](#_Toc170851201)

[Figure 2: Gantt chart of maze generation 9](#_Toc170851202)

# **Chapter 1: Introduction**

## Introduction

Tic-Tac-Toe is a simple yet interesting game that involves two players taking turns to mark spaces in a 3×3 grid. The player who succeeds in placing three of their marks in a horizontal, vertical, or diagonal row wins the game. The Minimax algorithm is a recursive method used to choose an optimal move for the AI player, making the game challenging and educational.

## Problem statement

The aim of this proposal is to design and implement a Tic-Tac-Toe game using the Minimax algorithm in a web technology stack (HTML, CSS, and JavaScript). Creating a competitive AI opponent using the Minimax algorithm can present several challenges:

1. Implementing the Minimax algorithm to calculate the best move.
2. Designing an intuitive user interface.
3. Ensuring the game is responsive and runs smoothly on various devices.

## 1.3. Objective

1. To develop a Tic-Tac-Toe game that utilizes the Minimax algorithm for the AI opponent.
2. To design a user-friendly interface.
3. To provide a challenging and engaging experience for the player.

## 1.4. Scope

The scopes of the Tic tac toe game are;

1. The game should be accessible via a web browser.
2. The AI opponent should be able to make optimal moves using the Minimax algorithm.
3. The game should provide real-time feedback and support various screen sizes.

# **CHAPTER 2: LITERATURE REVIEW**

Tic-Tac-Toe is one of the most popular and simplest games in the realm of artificial intelligence and game theory, often serving as an introductory example for teaching fundamental AI concepts. The game has been studied extensively, and numerous algorithms have been applied to ensure optimal gameplay, most notably the Minimax algorithm.

**Historical Context and Development**

The game of Tic-Tac-Toe, also known as Noughts and Crosses, dates back to ancient times, with versions found in Roman and Egyptian history. In modern times, it became a standard example in the study of combinatorial game theory and computer science.

# **CHAPTER 3: METHODOLOGY**

## 3.1. **Planning and Requirements Gathering**

1. Objective: Define the game's goals, target audience, and key features.
2. Requirements: Collect requirements through surveys and analysis of existing Tic-Tac-Toe games.
3. Documentation: Create a detailed project plan outlining milestones, timelines, and resources needed.

## **3.2. Game Design**

1. Grid Layout: Design the 3×3 grid layout.
2. Algorithms: Implement the Minimax algorithm for the AI.
3. User Interface: Design the UI with clear indicators for player and AI moves.
4. Player Controls: Define the control scheme for marking the grid.

## **3.3.Development**

1. Algorithm Implementation: Implement the Minimax algorithm to calculate the best move for the AI.
2. User Interface Development: Develop the UI based on the design mockups.
3. Player Controls: Program the control scheme for player interaction.

## **3.4.** **Testing**

1. Unit Testing: Test individual components for functionality.
2. Integration Testing: Ensure all components work together seamlessly.
3. User Testing: Conduct user testing sessions to gather feedback on usability.
4. Performance Testing: Test the game on different devices and screen sizes.

## 3.5. **Feedback and Iteration**

1. Feedback Collection: Gather feedback from testers and users.
2. Iteration: Make necessary adjustments based on feedback.

## 3.6. **Finalization and Deployment**

1. Polishing: Refine the game's visuals and fix any remaining bugs.
2. Documentation: Create comprehensive documentation for the game.
3. Deployment: Release the game on selected platforms.

## 3.7. **Post-Deployment Support**

1. Maintenance: Provide ongoing support and updates.
2. User Engagement: Engage with the community to keep the game interesting.

## **3.8.Tools and Technologies**

# Development Environment: Use HTML, CSS, and JavaScript.

# Graphics and Design Tools: Use tools like Adobe Photoshop or Illustrator.

# Testing Tools: Employ testing frameworks for automation.

# **Chapter 4: Feasibility study**

## 4.1 Technical Feasibility

1. Tools and Technologies: Development can be done using widely available tools.
2. Skills Required: The development team should have experience with web development and AI algorithms.
3. Algorithm Implementation: The Minimax algorithm is well-documented and feasible to integrate.

## 4.2. Economic Feasibility

1. Development Costs: Minimal costs related to software licenses and hosting.
2. Revenue Potential: The game can generate revenue through ads or a one-time purchase price.
3. Budget Allocation: Allocate budget for essential areas like design, development, and testing.

## 4.3. Operational Feasibility

1. User Engagement: Tic-Tac-Toe is known for its engaging nature.
2. Scalability: The game can be easily scaled to include different difficulty levels.
3. Maintenance: A small development team can manage ongoing maintenance and updates.

## 4.4. Schedule Feasibility

1. Development Timeline: A small, dedicated team can develop the initial version within 2-3 weeks.
2. Milestones: Key milestones include project planning, prototype development, testing, and final release.

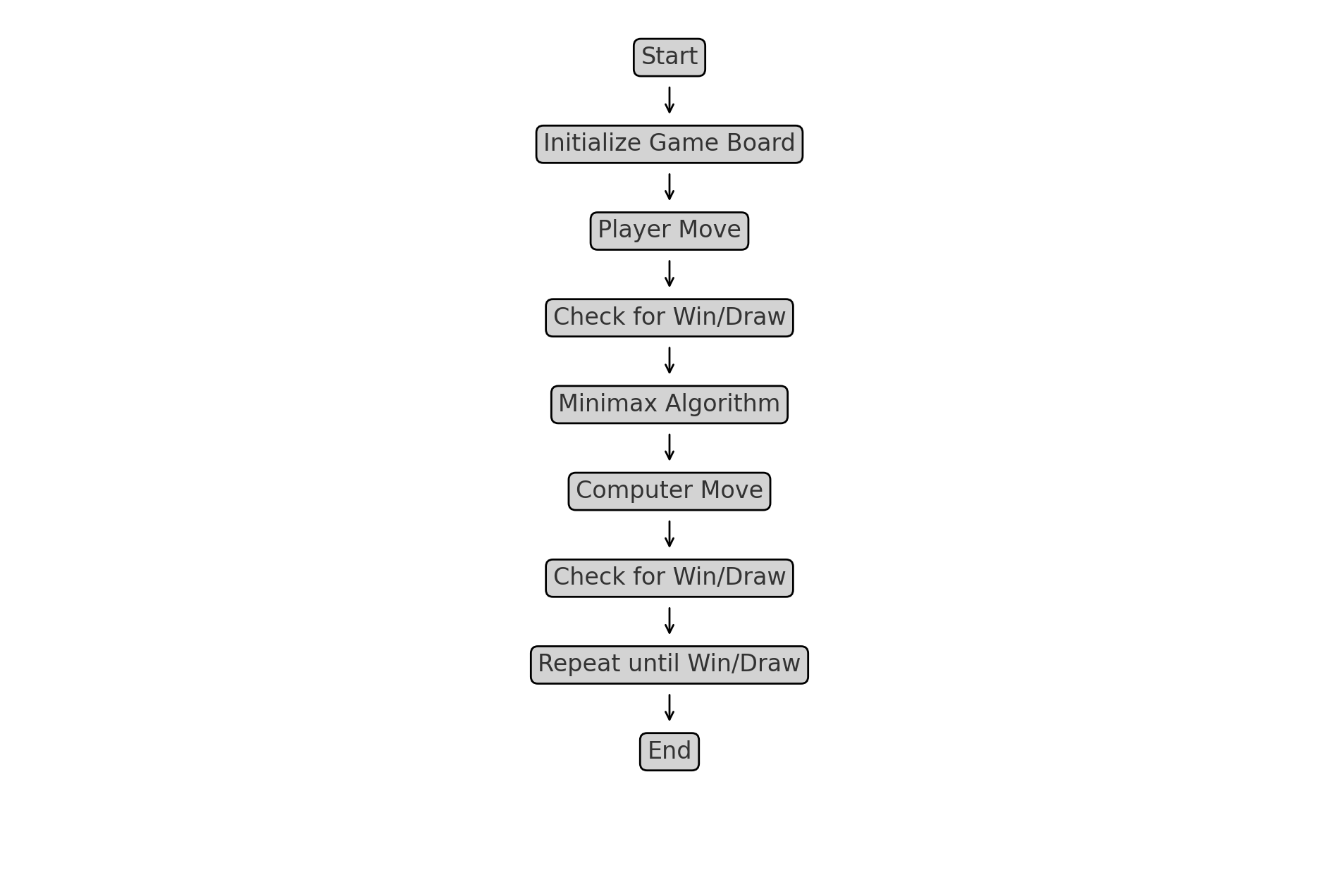
# **Chapter 5: System Analysis and Design**

## 5.1 Requirement Analysis

The system needs to fulfill the functional and non- functional requirements.

1. **Functional requirement**
2. Implement the Minimax algorithm for AI moves.
3. User controls for marking the grid.
4. Visual indicators for player and AI moves.
5. Real-time feedback on game status.
6. **Non- functional requirement**
7. **Performance: The game should run smoothly on multiple devices.**
8. **Usability: The interface should be intuitive and easy to navigate.**
9. **Scalability: The game should handle various difficulty levels.**
10. **Accessibility: The game should be accessible to players of all skill levels.**

## 5.2. Data modeling



**Figure 1: flowchart for a Tic tac toe game**

# **Chapter 6: Implementation**

## 6.1 Tools used (CASE Tools, Programming Languages, Database platform).

The tools that have used in this project are mentioned below:

1. Programming Languages: HTML, CSS, JavaScript
2. IDE: Visual Studio Code

## **6.2 Development plan:**

6.2.1. Phase 1: Planning and Design (2-3 days)

1. Define game rules and requirements.
2. Create wireframes and design mockups.
3. Set up the project repository and development environment.

2. Phase 2: Core Game Mechanics (3 days)

1. Implement the game grid and user interface.
2. Develop the Minimax algorithm for AI moves.
3. Implement input validation and conflict highlighting.

3. Phase 3: Advanced Features (2 day)

1. Add game controls (new game, reset).
2. Implement real-time feedback.

4. Phase 4: Testing and Refinement (2 day)

1. Conduct thorough testing across different browsers and devices.
2. Optimize performance and responsiveness.
3. Fix bugs and refine user experience.

5. Phase 5: Deployment and Documentation (1 day)

1. Deploy the game to a web server.
2. Project Timeline:
3. Total Duration: 7-10days

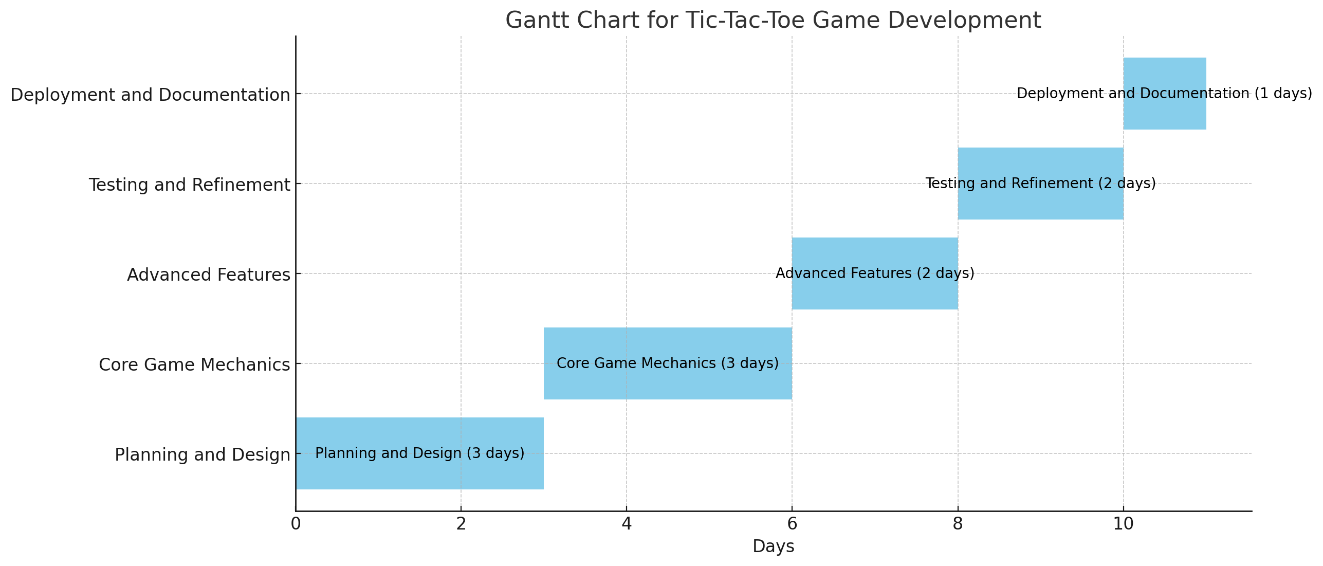


Figure 2: Gantt chart of Tic tac toe game

# **Chapter 7: Conclusion**

In conclusion, developing a Tic-Tac-Toe game using the Minimax algorithm in HTML, CSS, and JavaScript not only reinforces algorithmic understanding but also showcases the versatility of web technologies in creating interactive applications. It blends educational benefits with entertainment, making it a rewarding project for both developers and users. It enhances understanding of algorithms like