**Hackathon Project Phases Template** for the **AutoSage App** project.

# Hackathon Project Phases Template

**Project Title: Playful Al: Intelligent Board Game**

**Opponents and Advisors**

**Team Name: TechGenAI**

**Team Members:**

* B.Mani Teja
* J.sandhya
* D.Rmaya
* E.Akhil
* K.Anil

## Phase-1: Brainstorming & Ideation for Playful AI: Intelligent Board Game Opponents and Advisors

## Objective:

The primary objective of **Playful AI** in board games is to **enhance the player experience** by creating intelligent, adaptive, and immersive opponents or advisors that provide a dynamic, engaging, and personalized gaming experience.

**Key Points:**

**1.Problem Statement:**

The goal is to design a **Playful AI** for board games that **adapts to the player’s skill level**, provides **emotionally intelligent interactions**, offers **strategic guidance and feedback**, and enhances **multiplayer dynamics**.

2.**Proposed Solution:** To address the challenges faced in traditional board games, the proposed solution is the development of a Playful AI that incorporates advanced techniques to adapt to players' abilities, foster emotional engagement, offer personalized feedback, and enhance multiplayer dynamics.

3.**Expected Outcome:** The AI will provide a **dynamic and adaptive challenge**, ensuring players remain engaged regardless of their skill level. Players will feel a sense of accomplishment, as they face an AI that adjusts its difficulty based on their performance.

## Phase-2: Requirement Analysis

**Objective:**

**Playful AI in Board Games** refers to an **artificial intelligence system** designed to enhance the board game experience by providing **adaptive, intelligent, and emotionally aware interactions**. This AI behaves like a dynamic opponent or advisor, adjusting its strategies and behavior based on the player's performance, skill level, and emotional state.

**Key Points:**

1. **Technical Requirements:**

* 1. Programming Language: **Python**

○ Backend: **Game Logic and AI Integration**

○ Frontend: **User Interface (UI) Design**

○ Database: **Data Storage and Management**

1. **Functional Requirements:**

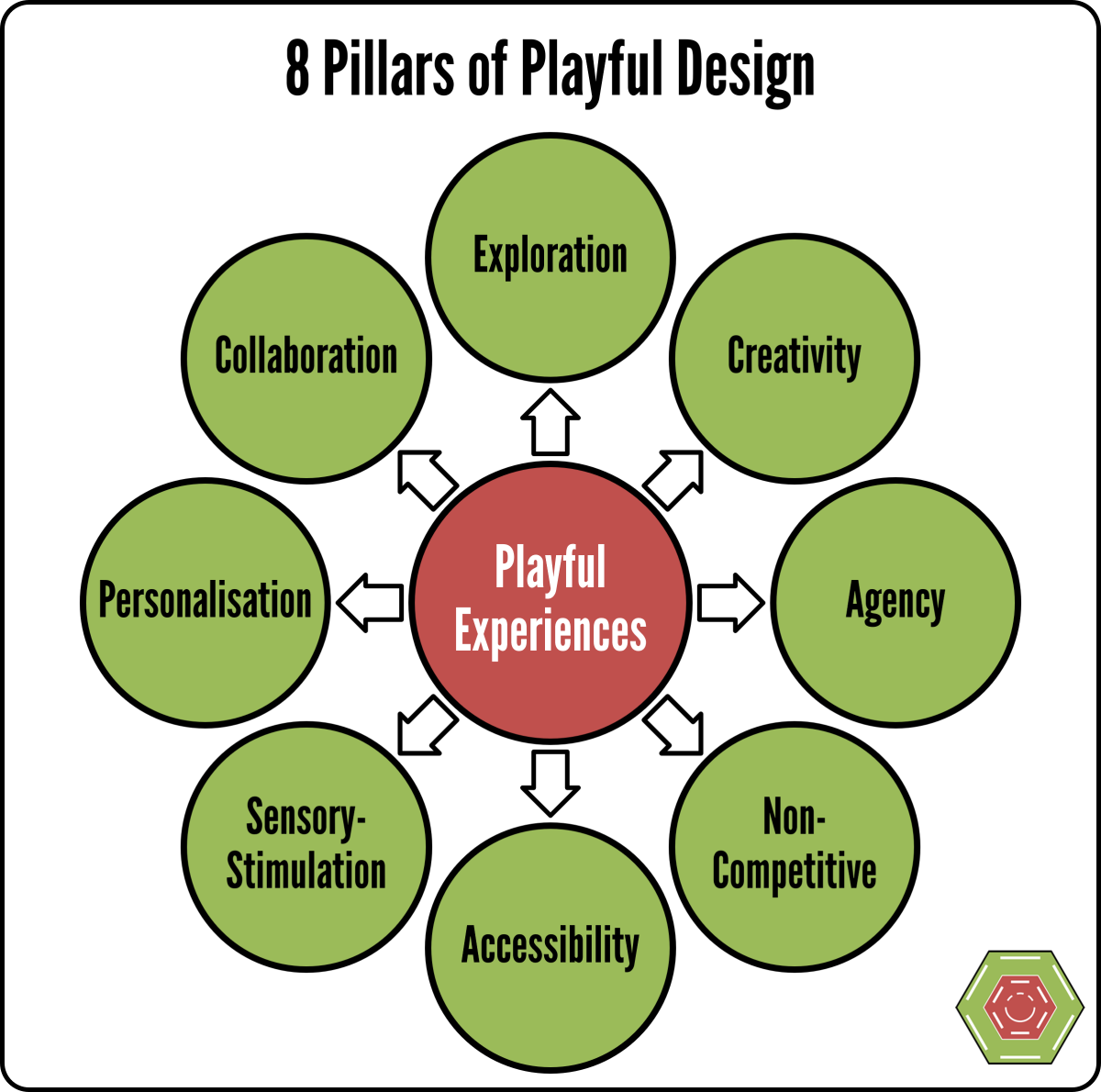
* AI opponents must adapt to different game scenarios and player strategies to maintain engagement.
* The system should offer multiple difficulty settings, adjusting AI behavior to match the selected level.

3. **Constraints & Challenges:**

* Developing "Playful AI: Intelligent Board Game Opponents and Advisors" involves addressing several constraints and challenges to ensure a successful implementation
* Ethical Considerations in AI Usage
* Balancing AI Difficulty

## Phase-3: Project Design

**Objective:**

Develop the architecture and user flow of the application.

**Key Points:**

1. **System Architecture:**

* 1. **User Interaction:** Players interact with the game through a user-friendly interface, initiating gameplay and seeking strategic advice..

○ **Game Logic Controller:** Manages the rules, mechanics, and progression of the

board game, ensuring fair and consistent gameplay.

1. **User Flow:**

Step1: starts a new game or continues an existing one through the main menu.

Step 2: During gameplay, the player makes a move or seeks advice.

Step 3: The AI Engine analyzes the current game state: If acting as an opponent, it calculates and executes its move.If providing advice, it offers strategic suggestions to the player.

Step 4: The Game Logic Controller updates the game state based on the actions taken.

Step 5: The updated game state is rendered on the user interface, and the cycle repeats until the game concludes.

1. **UI/UX Considerations:** 
   * **Visual Feedback:** Provide clear visual cues for AI decisions, player moves, and game state changes to enhance understanding and engagement
   * **Accessibility Options:** Include features such as adjustable text sizes, colorblind modes, and alternative input methods to accommodate diverse player needs.

## Phase-4: Project Planning (Agile Methodologies)

**Objective:**

Break down development tasks for efficient completion.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sprint** | **Task** | **Priority** | **Duration** | **Deadline** | **Assigned To** |
| Sprint 1 | Environment Setup & Tool Integration | 🔴 High | 6 hours  (Day 1) | End of Day  1 | Member 1 |
| Sprint 1 | Basic UI Development | 🟡 Medium | 2 hours  (Day 1) | End of Day  1 | Member 2 |
| Sprint 2 | AI Opponent Implementation | 🔴 High | 3 hours  (Day 2) | Mid-Day 2 | Member 1& 2 |
| Sprint 2 | Advisor Feature Development | 🔴 High | 3 hours  (Day 2) | Mid-Day 2 | Member 1&4 |
| Sprint 3 | UI/UX Enhancements | 🟡 Medium | 1.5 hours  (Day 2) | Mid-Day 2 | Member 2& 3 |
| Sprint 3 | Final Presentation & Deployment | 🟢 Low | 1 hour  (Day 2) | End of Day  2 | Entire Team |

**Sprint Planning with Priorities**

**Sprint 1 – Setup & Integration (Day 1)**

**(**🔴 **High Priority)** **Set up the development environment and install necessary**

**dependencies.**

**(**🔴 **High Priority)** **Integrate AI algorithms for game opponents and advisors.**

**(**🟡 **Medium Priority)** **Build a basic user interface with input fields.**

**Sprint 2 – Core Features & Debugging (Day 2)**

**(**🔴 **High Priority)** Implement AI-driven opponent strategies and advisor functionalities.

**(**🔴 **High Priority)** Debug AI behavior and handle errors in game logic.

**Sprint 3 – Testing, Enhancements & Submission (Day 2)**

**(**🟡 **Medium Priority)** Test AI performance, refine user interface, and fix UI bugs.

**(**🟢 **Low Priority)** Prepare final demonstration and deploy the application..

## Phase-5: Project Development

**Objective:**

Implement core features of the AutoSage App.

**Key Points:**

1. **Technology Stack Used:** 
   * **Frontend:** Utilize a framework like Unity or Godot to develop an interactive user interface for the board game.
   * **Backend:** Implement AI algorithms using reinforcement learning and Monte Carlo Tree Search (MCTS) to simulate intelligent opponents and advisors.
2. **Development Process:**

* **AI Integration:** Develop and integrate AI models capable of learning and adapting strategies for various board games.

1. **Challenges & Fixes:**

* **Challenge:** Complexity in modeling diverse board game strategies.
* **Fix:** Utilize reinforcement learning techniques to enable AI to learn optimal strategies

through self-play and adaptation

## Phase-6: Functional & Performance Testing

**Objective:**

Ensure that the Playful AI: Intelligent Board Game Opponents and Advisors works as expected.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Category** | **Test Scenario** | **Expected Outcome** | **Status** | **Tester** |
| TC-001 | Functional  Testing | AI opponent plays a complete game without errors | AI completes the game following all rules correctly | ✅ Passed | Tester 1 |
| TC-002 | Functional  Testing | AI advisor provides strategy recommendations during gameplay | Advisor offers relevant and effective strategies based on the current game state | ✅ Passed | Tester 2 |
| TC-003 | Performance  Testing | AI response time for move decisions under 500ms | AI makes move decisions within 500ms to ensure smooth gameplay | ⚠ Needs Optimization | Tester 3 |
| TC-004 | Bug Fixes & Improvements | Fixed incorrect AI move selections in specific game scenarios | AI now selects optimal moves in previously problematic scenarios | ✅ Fixed | Develop er |
| TC-005 | Final Validation | Ensure UI is responsive across devices | UI functions correctly on both mobile and desktop platforms | ❌ Failed - UI broken on mobile | Tester 2 |
| TC-006 | Deployment  Testing | Host the game application on the designated platform | Game is accessible online with all features operational | 🚀 Deployed | DevOps |

## Final Submission

* + **Project Report**: A detailed document outlining objectives, methodologies, development processes, challenges faced, and solutions implemented.
  + **Demo Video (3-5 Minutes)**: A concise video demonstrating the application's features, user interface, and AI functionalities.
  + **GitHub/Code Repository Link**: Access to the project's source code via a platform like GitHub, including all necessary files, documentation, and setup instructions.
  + **Presentation**: A structured presentation covering the project's key aspects, such as objectives, design considerations, development processes, and results.