**Project Title:** Cosmic Conquest – Interstellar Strategy Board Game

**Submitted By:**
Abdullah Kapadia (22K-4147)

Ali Yahya (22K-4417)

Syed Ansab Iqbal (22K-4520)

**Course:** Artificial Intelligence

**Instructor:** Miss Alina Arshad, Miss Almas Ayesha

**Submission Date:** 22/March/2024

## 1. Project Overview

● **Project Topic:**
Cosmic Conquest is an original strategy board game set in space where players navigate a dynamic network of star systems. The game uses a non-traditional board layout resembling a network graph (or a "web" of interconnected nodes) rather than a standard grid.

● **Objective:**
Develop the game mechanics and build an AI opponent using a modified Minimax algorithm for multi-player (or two-player head-to-head) strategic gameplay. The goal is to balance strategic planning with unpredictability from random events (such as cosmic disturbances) that affect movement between star systems.

## 2. Game Description

● **Original Game Background:**
While inspired by classic territory-control and race games, Cosmic Conquest is not based on any single existing game. Its inspiration comes from traditional strategic board games (like chess) but reimagined with a non-linear network board.

● **Innovations Introduced:**

- **New Board Layout:** Instead of a grid, the board is a network of star systems (nodes) interconnected by routes. Some routes can change dynamically (e.g., "wormholes" or cosmic storms).
- **Random Events:** Periodic cosmic events will alter connections between systems, forcing players to adapt their strategy.

## 3. AI Approach and Methodology

● **AI Techniques to be used:**

- **Modified Minimax Algorithm:** Adapted for non-linear board structures and potential multi-player scenarios.
- **Alpha-Beta Pruning:** To optimize decision-making in complex state spaces.
- **Optional Reinforcement Learning:** For self-play experiments to fine-tune strategies against dynamic board events.

● **Heuristic Design:**

- Evaluate game states by considering factors like control of key star systems, resource reserves, and the connectivity of each player's network.
- Positional advantage will be measured by both static control and flexibility in movement (access to multiple routes).

● **Complexity Analysis:**

- The dynamic board (network graph with random events) introduces challenges in state evaluation and branching factor; expect increased computational complexity compared to standard grid-based games.

- **Unique Movement & Resource Management:** Movement isn't simply based on dice rolls but also on resource tokens that players collect, which they can spend to influence travel or trigger events.
- **Impact on Gameplay:**
    - o Increased replay ability and depth due to dynamic board conditions.
    - o Encourages long-term strategic planning alongside tactical, adaptive decision-making.

## 4. Game Rules and Mechanics

● **Modified Rules:**

- The board is a network graph where nodes represent star systems.
- Players start at designated "home" systems and race to establish control over a set number of key nodes.
- Movement is determined by a combination of resource tokens and dice (or card draws) that add an element of chance.
- Random cosmic events (triggered each round) can open, close, or alter the connections between nodes.

● **Winning Conditions:**

- A player wins by achieving a specific strategic objective—such as dominating a predetermined number of key star systems or accumulating a target resource level.

● **Turn Sequence:**

- Each turn, a player rolls dice (or draws cards) and may spend resource tokens to move between nodes.
- After moves, a cosmic event is triggered which might change the network connections.
- Turns proceed sequentially, with opportunities for negotiation or temporary alliances if multi-player.

## 5. Implementation Plan

● **Programming Language:**
Python

● **Libraries and Tools:**

- **Pygame:** For creating the graphical user interface and displaying the network board.
- **NumPy:** For handling game state data and computations.
- **AI Libraries:** (e.g., TensorFlow or Scikit-learn) if reinforcement learning components are implemented.

● **Milestones and Timeline:**

- **Week 1-2:** Finalize game design, rules, and board layout (graph structure).
- **Week 3-4:** Develop the base game mechanics (movement, resource management, event system).
- **Week 5-6:** Implement and test the AI strategy (Minimax with Alpha-Beta Pruning, design heuristics).
- **Week 7:** Integrate AI with the game and conduct multi-play testing sessions.
- **Week 8:** Final testing, debugging, and report preparation.

## 6. Reference


- Russell, S., & Norvig, P. (2020). Artificial Intelligence: A Modern Approach (4th Ed.).

  Available on [https://www.pearson.com/](https://www.pearson.com/)

- Yannakakis, G. N., & Togelius, J. (2018). Artificial Intelligence and Games. Springer.

  Available on [https://www.springer.com/in](https://www.springer.com/in)

- Watts, D. J. (1999). Small Worlds: The Dynamics of Networks Between Order and Randomness. Princeton University Press.

Available on https://press.princeton.edu/

- Pygame Official Documentation
  Available on https://www.pygame.org/docs/