Project Report

**Members:**

**M. Hamiz Siddiqui (22K5055)**

**Zuhair Zeshan (22K5062)**

## Abstract

*"Royal Virus"* is a two-player turn-based strategy game where players compete by capturing elemental tiles on an 8x8 grid. Each tile has a nature type (Water, Air, Fire, Earth, or Neutral) with associated point values. The player's objective is to reach a score of 35 or more by strategically moving across the board and capturing high-value tiles. The game integrates intelligent AI using the minimax algorithm with alpha-beta pruning to simulate a challenging opponent. Visually appealing with emoji and color-coded elements, the game encourages strategic thinking and planning.

## Introduction

The game is built using Python and the Pygame library. It provides an engaging and educational experience focused on decision-making and path optimization. Players (one human, one AI) start at opposite corners of the board and take turns moving to adjacent tiles, capturing their nature type and scoring points. The game ends when one player reaches 35 points, no positive tiles remain, or no valid moves are possible.

### Objectives

* Develop an engaging, strategy-based two-player game.
* Implement AI using the minimax algorithm for competitive play.
* Create a colorful and intuitive user interface using emojis and color coding.
* Ensure fairness by balancing the point system across all tile types.

### Problem Statement

Designing a turn-based strategy game that offers balanced gameplay, engaging visuals, and intelligent AI poses challenges in algorithm efficiency, real-time updates, and strategic fairness. *Royal Virus* addresses these challenges by blending heuristic AI decision-making with a visually intuitive layout.

## Methodology

1. **Game Board Design**:  
   The board is an 8x8 grid initialized with random elemental tiles (Water, Air, Fire, Earth, Neutral), each with specific point values:
   * Water: +2 points
   * Air: +3 points
   * Fire: +4 points
   * Earth: +5 points
   * Neutral: 0 points
2. **Player Mechanics**:
   * Player 1 (Human) starts at the top-left corner.
   * Player 2 (AI) starts at the bottom-right corner.
   * On each turn, a player moves to an adjacent tile (up, down, left, or right) to capture it.
3. **AI Strategy**:
   * Uses the **minimax algorithm** with **alpha-beta pruning** to evaluate the best possible move.
   * Evaluates game state up to a specified depth and picks the optimal path to maximize its score while minimizing the opponent’s advantage.
4. **User Interface**:
   * Tiles are displayed using emojis representing each element.
   * Scoreboard displays both players’ current scores.
   * Visual feedback helps users see which tiles have been captured.
5. **Game End Conditions**:
   * A player reaches 35 points.
   * No positive-value tiles are left.
   * No valid moves are available for both players.
6. **Development Tools**:
   * Python 3
   * Pygame (for rendering graphics and handling events)
   * Random module (to initialize board)

## GUI







