Univerzitet u Sarajevu
Elektrotehnički fakultet
Sarajevo

DSAI
DATA SCIENCE
& ARTIFICIAL INTELLIGENCE

Game of The Amazons

Project Proposal

Course: Artificial Intelligence

Professor: Prof. Dr. Izudin Džafić

Students: Emin Hadžiabdić, Armin Memišević, Muhammed Pašić

Academic year: 2025/26

Student IDs: 19960, 20016, 20109

Sarajevo, November 19th 2025

Table of Contents:

1. Introduction

- **Purpose:** This document outlines how we plan to build the Game of the Amazons from start to finish. It explains the structure of the project, the rules we will implement, the algorithms we will use for move generation and AI, and how the GUI will be designed. The goal is to have a clear plan that shows what we are building and how we will approach each part of the work.
- **Justification:** Amazons is well-suited for our project because it has simple rules but high strategic depth. Each turn combines Queen movement and Arrow placement, creating a large branching factor and non-trivial move generation. The game fits perfectly between medium and complex difficulty, making it an ideal problem for us; challenging enough to demonstrate AI algorithms and heuristics, but manageable to implement within the project timeframe and required skillset.
- **Scope:** The project covers GUI development using natID, graph-based algorithms for move and territory analysis, core game logic for the Amazons ruleset, and AI decision-making through heuristic evaluation and search methods.

2. Objectives

- **Functional Goal:** Create a polished, functional Game of Amazons.
- **Game Rules:** Implement all core rules and mechanics (e.g., movement of Queens, placement of Arrows).
- **Architecture:** Develop a clean, modular architecture.
- **GUI:** Responsive GUI with necessary interactions and visual & sound clarity.

3. System Architecture

- **Overall Principle:** Follow a modular architecture separating core logic from the GUI.
- **Core Components:**
    - **Board/Grid:** Manages the game grid, Queen positions, Arrow positions.
    - **Game State:** Tracks turn order, move history, and valid moves.
    - **Move Generator:** Calculates all legal moves (Queen move + Arrow placement).
    - **Win Checker:** Calculates game end (no valid moves left).
    - **AI:** Generates moves based on search algorithms and heuristics.
- **GUI Components:**
    - **Gui Board:** Renders the grid, Queens, and Arrows.
    - **Gui Controller:** Handles input, menu actions, and turn updates.

4. Algorithms

- **Core Game Logic:**
    - **Line-of-Sight Ray Casting**
      Queens move like chess queens (orthogonal + diagonal), so every move is produced by scanning outward in 8 directions until hitting an obstacle.

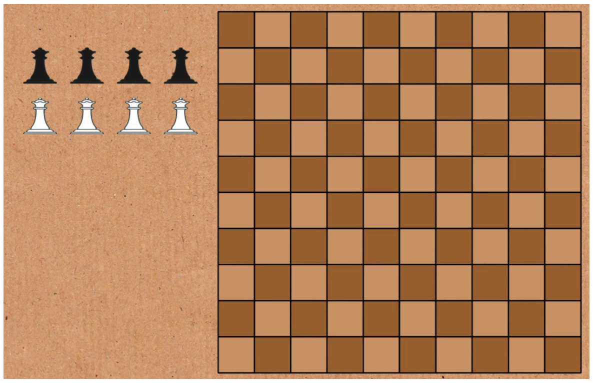
      This ensures correct move generation with minimal overhead.
    - **Move Validation Rule:** A move is legal only if
        - The Queen has an unobstructed path to the target square (ray-cast check).

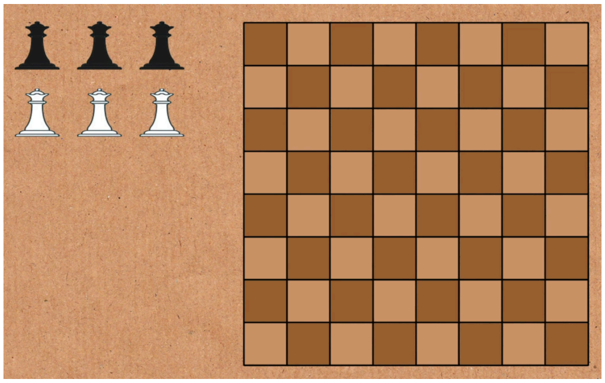
    - The Arrow shot after the movement also has an unobstructed path (second ray-cast).
  - These checks ensure correctness and prevent illegal jumps or passes through blocked tiles.
- **Heuristic Evaluation:** The AI evaluates board states using established Amazons strategy metrics. Our evaluation function includes:
  - **Mobility**
    Count of legal moves for each player.
    More mobility = stronger position.
    This is the primary heuristic in Amazon's AI.
  - **Territory / Zone Control**
    We approximate future territory by flood-filling from each queen and counting reachable tiles.
    The player controlling more accessible areas gains a scoring advantage.
  - **Distance to Opponent / Spatial Influence**
    Optional heuristic: reward positions where queens have more influence over central or strategic zones.

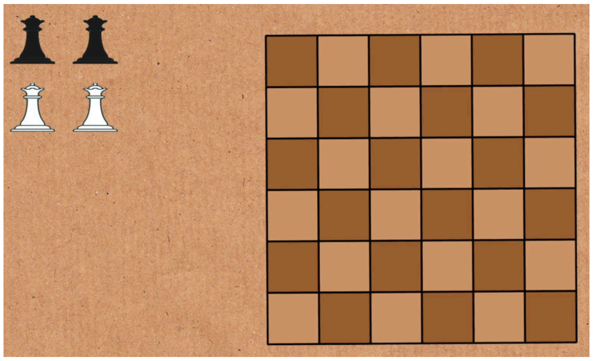
  This gives the AI meaningful guidance during search.
- **Search Algorithm:** Our main AI engine uses:
  - **Minimax Search with Alpha-Beta Pruning**
  - **Features included:**
    - depth-limited search (depth depends on difficulty),
    - alpha-beta pruning for efficiency,
    - evaluation at leaf nodes using heuristics above.
  - **Difficulty levels:**
    - **Easy:** shallow depth (1–2),
    - **Medium:** deeper search (3–4),
    - **Hard:** deepest stable search (5+).

5. GUI Design

- **Board appearance:** Wooden, Black&White, maybe we add more.
- **Interaction:** Interactive Queens, Highlights for valid moves (Queens and Arrows), Selection of moves.
- **Animations:** Shooting Arrow, Field explosion, Moving Queens.
- **Sound Effects:** For all Animations and for Victory/Defeat.
- **Board Size Feature:** Small (Board 6x6), Medium(Board 8x8), Big(10x10).
- **Difficulty Feature:** Easy, Medium, Hard (depth-limited search)
- **Language Feature:** Bosnian, English.
- **GUI Demo Design (wooden style)**







6. Team Responsibilities and Task Distribution

- **Emin Hadžiabdić**
    - Development of the graphical user interface (GUI) using **natID**
    - Integration of GUI elements with the game engine
    - Ensuring smooth user interaction and visual consistency
- **Muhammed Pašić**
    - Implementation of core game rules and logic
    - Validation of legal moves, board updates, and turn management
    - Collaboration with engine developer to ensure correct rule execution
- **Armin Memišević**
    - Development of the main game engine
    - Implementation of the **Minimax algorithm** and **alpha–beta pruning**.
    - Coordination with rule and GUI components to ensure full game functionality