Wentworth Institute of Technology

Arcane Chess

Software Requirements Specification

1.2

Group 04

David Costa

Ibukunoluwa Folajimi

Bryce Parkinson

Nathaly Phrasavath

10/9/2025

Contents

Revision History

Date	Version	Description	Author
9/25/2025	1.0	Initial Draft of SRS	David Costa, Ibukun Folajimi, Bryce Parkinson, Nathaly Phrasavath
9/28/2025	1.1	Revised SRS Remove design related content	Ibukun Folajimi
10/16/2025	1.2	Tweaks based on peer feedback, minor punctuation fixes	David Costa

1. Introduction

Chess is a classic board game that has been played around the world for thousands of years. In more recent times, there have been games which take the original game and put a spin on it. Whether they add a First-Person Shooter aspect to it or use lasers and mirrors to win pieces on the board, there have been entertaining and interesting developments in the community. We plan to add a new version of the game to the public repertoire which will involve trading card game aspects similar to the classic *Magic the Gathering*.

During gameplay, each player will get to choose whether to play a card, or to move a piece every turn. This dynamic will allow for strategy, where a player either chooses to get a leg-up in positional development, or take the gamble of losing a turn for a potentially more powerful investment. Cards will all be available for deckbuilding before actual gameplay, allowing for even more strategy, while partially limiting the "RNG" aspect of getting random cards. Cards can have a variety of effects including allowing pieces to move in new ways; altering the state of the gameboard; punishing the other player; giving strong temporary bonuses; or giving the player access to entirely new pieces.

1.1. Problem statements

- Balancing the relative strength of cards and pieces may prove difficult given the goal of 32 cards and the timeframe of 1 semester.
- Unforeseen server problems must be accounted for. For example, if the server becomes non-functional, gameplay must be paused until functionality is fully restored or the offending client's time-out (90 seconds).
- Ensuring scalability of the game design so that it supports multiple concurrent two-player matches.
- Designing win conditions, avoiding situations where matches become prolonged, boring, or stalemated.
- Preventing excessive reliance on randomness by allowing players to pre-build their decks, while still preserving unpredictability and excitement.

1.2. Proposed solution

We propose Arcane Chess, a hybrid online game that combines classic chess rules with strategic card-based attributes.

- Players shall choose to either play a card or move a piece on their turn.
- Cards shall affect the gameboard in the following ways:
  - Transforming pieces into new pieces or other pieces of equal value.
  - Transforming a pawn on the closest rank into a queen for 1 move.
  - Placing a new, immovable "cursed effigy" piece on the board that limits the other player's movement until it is captured.
  - A one-time-use card that switches kings upon either player getting checkmate.

  - Summoning a free pawn-like piece on a randomly selected safe square, that cannot be promoted upon reaching the back rank.
  - Creating a 1 tile “safe zone” border around the chessboard, that can be moved into but does not allow for capturing pieces when moving into, nor when moving out of.
- The game shall be web-based, with Python handling game logic and server processes, and JavaScript for the user interface.

1.3. Novelty

This project introduces a new gameplay layer by merging two strategic systems:

- Classic chess for structure and familiarity.
- Card mechanics introduce variability, unpredictability, and strategic depth.

Unlike other chess variants, Arcane Chess will focus on balance, ensuring no card will dominate the gameboard, while maintaining the core chess principles like checkmate or stalemate.

2. Requirements

2.1. Functional requirements

- The system shall provide a visual web interface for players to interact with using their cursor.
- Once a player connects to the game server, they shall build their deck of 16 cards from the available 32. Once the player is satisfied with the deck, the player shall click a button to enter the matchmaking queue.
- Once two players have joined the queue, players shall be placed into a match together.
- Players shall be able to view an enlarged view of each of their cards by hovering their cursor on the card.
- All player actions shall be performed by selecting a piece or card using the cursor, then selecting either where the piece shall move, or which piece/location the card shall affect.
- After performing an action, the player’s screen shall be updated according to the action they performed:
  - Visual effects such as a color change or sprite alteration shall indicate the effect of a card.
  - A piece being moved shall appear in its new location.
- When pieces are taken by an opponent, they shall appear along the top of the game view.
- Pieces taken by the player shall appear along the bottom of the view.
- Cards held in hand shall appear along the right of the view.

- The game state shall include a timer for each player which begins at 15 minutes and counts down for each player while it is their turn. If a player’s timer reaches 0, they lose the game automatically.
- When a game concludes, both players shall be prompted to either rematch, or rejoin the queue to find a new opponent.

2.2. Non-functional requirements

- The system shall update the opponent’s screen no more than 5 seconds after a player’s move has been registered.
- The server which manages the game state shall have a maximum capacity of 20 concurrent games.
- The server should validate each player's turn, through confirmation of movements and/or played cards based on the previous state of the gameboard and players hands.
- The system shall have a dedicated URL for players to access the game.
- The server shall be hosted on a team member’s machine during the duration of the project.
- The game shall be accessible from any web browser regardless of the user’s machine or operating system.