Minesweeper Application

Overview

This application will contain a version of minesweeper, with standard difficulty options of

1. Beginner (10 x 10 grid, 10 mines)
2. Intermediate (16 x 16 grid, 40 mines)
3. Expert (30 x 16 grid, 99 mines)

There will also be an ‘AI’ functionality, where the user can play against the AI on the same, or different difficulties, with the boards displayed side by side.

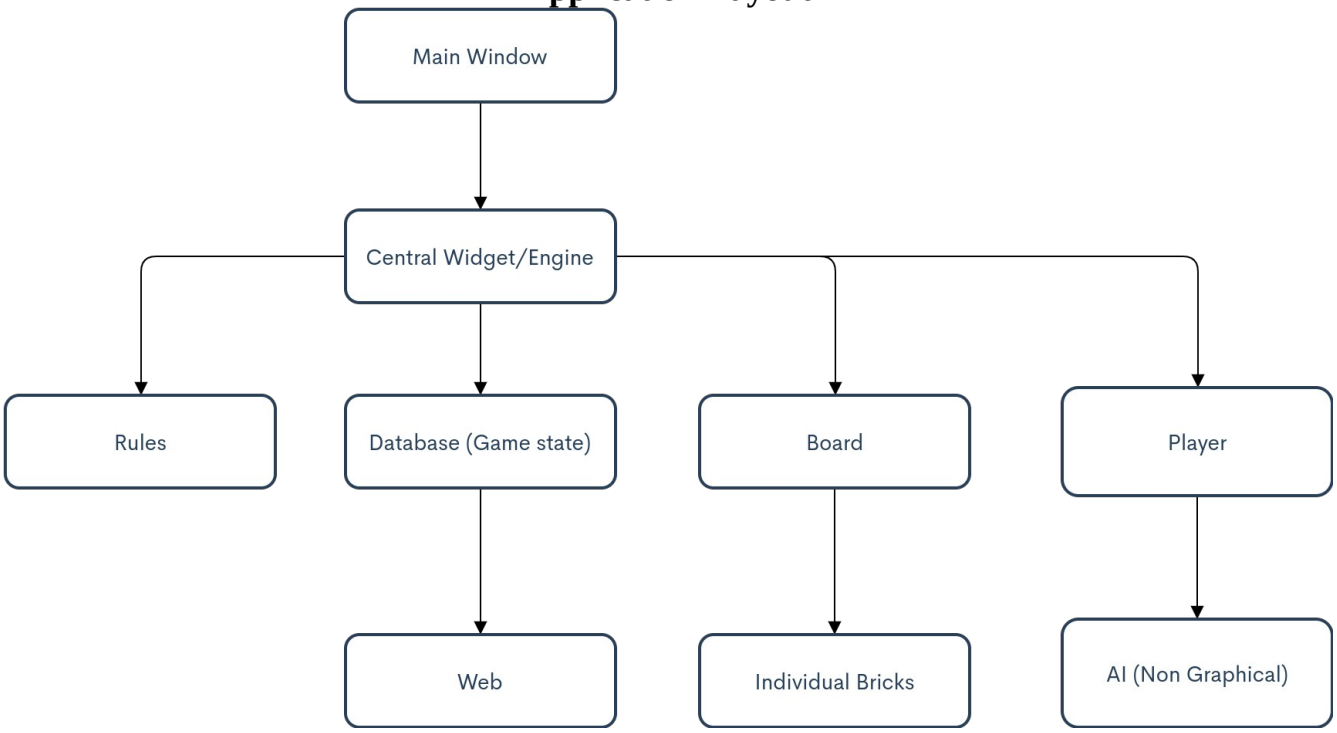
The AI can play on beginner, intermediate, or expert boards, and the user will also have the option of setting the skill level of the AI

The user will also be able to watch the AI play the game, without playing themselves.

The current game state, as well as statistics about each game and players will be saved to a database.

A web page will display the high-scores for human players, AI players, and both combined. High scores will be based on board size/difficulty and elapsed time.

Application Layout



Classes

- Main Window:
  - Windows to begin game (Single/Multiplayer window, Difficulty window)
  - Menus:
    - File: Load, Save, New, Quit
    - Edit: Preferences
    - Help: Rules/Link to website
- Central Widget / Engine:
  - Get moves from player/ receives moves from board
  - Verifies every moves with rules/game state
  - Once move is verified initiates change in game state and board
  - Contains Board class, rules class, and players class
- Board:
  - Contains grid layout of current game state
  - Each item in grid is a Brick
  - Contains all graphics for altering board and appearance of bricks on board
  - Gets signals from Brick clicks

- Rules:
  - Contains valid actions, and logic to determine validity of actions
    - Only one brick can be clicked at a time
    - If bricks have already been clicked they cannot be clicked again
    - ...
- Player:
  - Human or AI
    - If player is AI, contains logic for solving game
      - Limits information available to AI
    - If player is Human contains player profile, high scores, other statistics about game play for that player
- Brick:
  - Contains information about square, including visibility, coordinates, bomb/no-bomb, bombs touching, flagged/no flag, coordinates

Database

Each Player's information and game information will be stored separately whether the player is the computer or a person.

Game state – One to one relationship with game info

Coordinates (either custom type or list)	Bomb (bool)	Visibility (char varying)	Near mines (int)	Mine # ? (int)	Game ID (int)	Game-StateID (int)

Game Info- One to one relationship with Game state | One to one relationship with Player Info

Game ID (int)	Difficulty (char varying)	Time Elapsed (int)	Mines Left (int)

Player Info – One to one relationship with Game info

Player ID (int)	Game ID (int)	Player Name (char varying)	Human/AI (char varying)