Here will be main page, but for now, the possible game list:

**Checkers**:

* + A strategy board game for two players which involves diagonal moves of uniform game pieces and mandatory captures by jumping over the enemy's pieces.

**Chess**:

* + A classic strategy game played by two players on an 8x8 checkered board, with different pieces each having their own unique movements.

**Connect Four**:

* + A two-player connection game in which the players choose a color and then take turns dropping colored discs into a seven-column, six-row vertically suspended grid.

**Snake**:

* + The player controls a snake, guiding it towards food and away from walls or its own ever-extending body.

**Memory (Concentration)**:

* + A card game where players flip pairs of cards over, trying to find matching pairs with the cards facing down.

**Minesweeper**:

* + A single-player puzzle game where the player uncovers squares on the grid, trying not to uncover a mine.

**Hangman**:

* + A word guessing game where one player thinks of a word and the other tries to guess it by suggesting letters.

**Sudoku**:

* + A logic-based, combinatorial number-placement puzzle where you fill a 9x9 grid with digits so that each column, each row, and each of the nine 3x3 subgrids contain all of the digits from 1 to 9.

**Solitaire (Klondike)**:

* + A single-player card game where the goal is to move all the cards to a foundation, sorting them by suit and in ascending order.

**2048**:

* + A sliding block puzzle game where you combine tiles with the same numbers to achieve a high score.

**Reversi (Othello)**:

* + A strategy board game where players take turns placing disks on the board with their assigned color facing up, with the objective to have the majority of disks turned to display their color when the last playable empty square is filled.

**Four-in-a-Row**:

* + Similar to Connect Four, a game where players aim to connect a line of four horizontal, vertical, or diagonal pieces.

Contents

[**1.** **High-Level Architecture:** 3](#_Toc150922990)

[**2.** **Application Components:** 4](#_Toc150922991)

[**3.** **Development Plan:** 5](#_Toc150922992)

[**4.** **Diagram** 6](#_Toc150922993)

# **High-Level Architecture:**

1. **Core Game Engine**:
   * This module will handle rendering, input management, and basic game loop functionality. It can be designed to be game-agnostic, meaning it provides a framework that each specific game's logic can plug into.
2. **Game Logic**:
   * Separate modules for each game (checkers, chess, and **Connect Four**), containing rules, moves validation, game state management, and AI .
3. **User Interface (UI)**:
   * A module to handle all the user interactions, draw the game board, pieces, and menus. This will call into the core game engine to render the game state on the screen.
4. **Common Utilities**:
   * Shared services like handling high scores, saving/loading game state, and utility functions.

# **Application Components:**

1. **Main Menu**:
   * The entry point of the application where users can select which game to play.
2. **Game Selection**:
   * A screen that lists the available games. Upon selection, it initializes the game logic for the chosen game.
3. **Gameplay Area**:
   * The interface where the game is actually played. It includes the game board and pieces, and it interacts with the game logic modules.
4. **Settings/Options**:
   * Allows users to adjust game settings, such as difficulty levels for AI or game themes.
5. **Results Screen**:
   * Displays the outcome of the game and options to play again or return to the main menu.

# **Development Plan:**

1. **Phase 1 - Setup and Core Development**:
   * Establish the project repository and set up the development environment.
   * Develop the core game engine that can be used by all games.
   * Create the main menu UI.
2. **Phase 2 - Individual Game Logic**:
   * Develop the game logic for checkers, chess, and **Connect Four** in parallel.
   * Each team member can be responsible for one game.
3. **Phase 3 - UI Integration**:
   * Design and integrate the UI for each game with the core engine.
   * Ensure there is a consistent look and feel across all games.
4. **Phase 4 - Testing and Refinement**:
   * Playtest each game and refine the mechanics and UI.
   * Address any bugs or user feedback.
5. **Phase 5 - Finalization**:
   * Implement the settings/options and results screens.
   * Final testing and quality assurance.
6. **Launch**:
   * Prepare for deployment.
   * Release the application.

# **Diagram**

A screen shot of a computer

Description automatically generated