Documentatie kijk beden voor allemaal links enz

**Klassendiagram: Structuur van de JavaScript-klassen (Damspel)**

**Klassen en hun rollen:**

* **Piece:**
  + Vertegenwoordigt een dambordstuk.
  + Attributen:
    - color: De kleur van het stuk ("Red" of "Black").
    - isKing: Een boolean die aangeeft of het stuk een koning is.
    - element: Het DOM-element dat het stuk op de pagina weergeeft.
  + Methoden:
    - constructor(color): Maakt een nieuw Piece-object met de gegeven kleur.
    - createPieceElement(): Maakt het DOM-element voor het stuk.
    - makeKing(): Promoveert het stuk naar een koning.
* **Board:**
  + Vertegenwoordigt het dambord.
  + Attributen:
    - size: De afmeting van het bord (standaard 8).
    - squares: Een 2D-array van Piece-objecten (of null) die de bezetting van elk veld op het bord weergeeft.
    - element: Het DOM-element dat het bord op de pagina weergeeft.
    - validMoves: Een array met de valide zetten
  + Methoden:
    - constructor(size): Maakt een nieuw Board-object met de gegeven afmeting.
    - createBoard(): Maakt het bord en de velden in het DOM.
    - addPiece(square, row, col, color): Plaatst een stuk op het bord op de gegeven positie.
    - getPiece(row, col): Haalt het stuk op de gegeven positie op.
    - isValidPosition(row, col): Controleert of een rij/kolom-positie binnen de grenzen van het bord valt.
    - clearValidMoves(): Reset de valid moves array en styling.
    - showValidMoves(): Toont de valide zetten op het bord.
* **Game:**
  + Beheert de algehele gang van het spel.
  + Attributen:
    - board: Een Board-object dat het dambord vertegenwoordigt.
    - scoreManager: Een ScoreManager-object voor het bijhouden van de score.
    - currentPlayer: De kleur van de speler die aan de beurt is ("Red" of "Black").
    - selectedPiece: De rij en kolom van het geselecteerde stuk.
    - lastCapturePosition: De rij en kolom van de laatste vangst.
    - gameActive: Een boolean die aangeeft of het spel actief is.
    - hasMadeMove: Boolean die aangeeft of de speler een zet heeft gedaan.
    - turnConfirmationNeeded: Boolean die aangeeft of de zet bevestigd moet worden.
    - turnDisplay: Het DOM-element dat de huidige speler weergeeft.
    - confirmTurnButton: De DOM-elementen om de zet te bevestigen.
    - winModal: Het DOM-element dat het win-modal weergeeft.
    - moveSound: Audio object voor zet geluid.
    - captureSound: Audio object voor vang geluid.
  + Methoden:
    - constructor(): Maakt een nieuw Game-object en initialiseert het bord, de spelers en de UI.
    - setupWinModal(): Zet de win-modal op.
    - checkWinCondition(): Controleert of er een winnaar is.
    - setupEventListeners(): Stelt de event listeners in voor het bord en de knoppen.
    - start(): Start het spel.
    - handleBoardClick(e): Verwerkt klikken op het bord.
    - clearSelection(): Selectie van stukken resetten.
    - checkCaptureInDirection(row, col, dirRow, dirCol): Controleert of er een vangst mogelijk is in een bepaalde richting.
    - checkForCaptures(row, col): Controleert of er zetten zijn om te slaan.
    - highlightValidMoves(row, col): Highlight de valide zetten voor een geselecteerd stuk.
    - movePiece(row, col): Verplaatst een stuk op het bord.
    - confirmTurn(): Bevestigt de zet van de speler.
    - resetGame(): Reset het spel.
    - checkForKingCaptures(row, col): Controleert of een koning een stuk kan slaan.
    - checkKingCaptureInDirection(row, col, dirRow, dirCol): Controleert of een koning in een bepaalde richting kan slaan
* **ScoreManager:**
  + Verantwoordelijk voor het beheren van de scores.
  + Attributen:
    - scores: Object met de scores van elke speler.
    - redScoreElement: DOM element voor rode speler score.
    - blackScoreElement: DOM element voor zwarte speler score.
    - redCapturesElement: DOM element voor rode speler captures.
    - blackCapturesElement: DOM element voor zwarte speler captures.
  + Methoden:
    - constructor(): Constructor.
    - updateScore(color, type): Update de score.
    - resetScores(): Reset de scores.

**Relaties tussen de klassen:**

* Een Game object heeft een Board object en een ScoreManager object.
* Een Board object bevat een 2D-array van Piece objecten (of null).
* Een Piece object is gekoppeld aan een Player (via de kleur).

**flowchart:**

1. **Start spel:** Het spel begint.
2. **Bord initialiseren:** Het dambord wordt gemaakt en de stukken worden op de beginposities geplaatst.
3. **Huidige speler instellen:** De eerste speler (Rood) wordt geselecteerd.
4. **Speler selecteert een stuk?:** Het spel wacht tot de huidige speler een stuk selecteert.
5. **Valide zetten voor stuk tonen:** De mogelijke zetten voor het geselecteerde stuk worden gemarkeerd.
6. **Speler kiest een zet?:** Het spel wacht tot de speler een van de gemarkeerde zetten kiest.
7. **Zet uitvoeren:** De zet wordt uitgevoerd (het stuk wordt verplaatst).
8. **Is er een stuk geslagen?:** Er wordt gecontroleerd of er tijdens de zet een stuk is geslagen.
9. **Score bijwerken:** Als er een stuk is geslagen, wordt de score van de speler die het stuk heeft geslagen, bijgewerkt.
10. **Is de zet geldig?:** Controleer of de gekozen zet aan de regels voldoet.
11. **Is het spel voorbij?:** Er wordt gecontroleerd of het spel voorbij is (bijvoorbeeld als een speler geen stukken meer heeft).
12. **Moet de beurt worden bevestigd?:** Er wordt gecontroleerd of de zet moet worden bevestigd.
13. **Toon Bevestigingsknop:** De knop om de beurt te bevestigen wordt getoond.
14. **Speler bevestigt de beurt?:** Er wordt gewacht op de speler om de beurt te bevestigen.
15. **Schakel speler:** De beurt wordt doorgegeven aan de andere speler.
16. **Toon winnaar:** Als het spel voorbij is, wordt de winnaar getoond.
17. **Reset spel na 2 seconden:** Na 2 seconden wordt het spel gereset en begint het opnieuw.

activityDiagram start --> [Player Selects Piece] [Player Selects Piece] --> {Is Selected Piece Valid?} {Is Selected Piece Valid?} --> [No] --> [End] {Is Selected Piece Valid?} --> [Yes] --> (Determine Valid Moves) (Determine Valid Moves) --> [Player Chooses Move] [Player Chooses Move] --> {Is Chosen Move Valid?} {Is Chosen Move Valid?} --> [No] --> [End] {Is Chosen Move Valid?} --> [Yes] --> (Execute Move) (Execute Move) --> {Is Move a Capture?} {Is Move a Capture?} --> [Yes] --> (Remove Captured Piece) (Remove Captured Piece) --> (Check for More Captures) {Is Move a Capture?} --> [No] --> (End Move) (Check for More Captures) --> {More Captures Available?} {More Captures Available?} --> [Yes] --> [Player Chooses Move] {More Captures Available?} --> [No] --> (End Move) (End Move) --> [End] [End] --> stop

**wire frame**

Afbeelding met plein, schermopname, Rechthoek, schaak

Door AI gegenereerde inhoud is mogelijk onjuist.

<https://trello.com/invite/b/67f96a261df76d127b4b4db1/ATTIb6c0a7a975a497ee946f520c3cfd3216545A51BE/dam-spel> = trello bord

<https://github.com/mosnoord123>

github data beheer