Quoridor Game - Dart API Documentation

Generated from dartdoc

Date: December 21, 2025

Table of Contents

Game > Logic > Computer > Player

Classes

```
ComputerPlayer
```

AI player that uses minimax with alpha-beta pruning to select moves.

Class: ComputerPlayer class

AI player that uses minimax with alpha-beta pruning to select moves. Generates all possible successor states (pawn moves and wall placements), evaluates them using a weighted heuristic, and chooses the best move.

Constructors

```
ComputerPlayer ({ required Pawn pawn , required Map < int , List < int > > validMoves ,
required Map < int , String > wallPos , required String wallColor , required int
myInitRow , required int oppInitRow })
```

Properties

```
hashCode → int
```

The hash code for this object. no setter inherited

```
isSpecialAdded ↔ bool
```

Whether special jump/diagonal moves have been temporarily added. getter/setter pair inherited

```
myInitRow ↔ int
```

The starting row for this player (goal row for opponent). getter/setter pair inherited

```
numOfWalls ↔ int
```

Number of walls this player has left to place. getter/setter pair inherited

```
oppInitRow ↔ int
```

The starting row for the opponent (this player's goal row). getter/setter pair inherited

```
pawn ↔ Pawn
```

The pawn controlled by this player. getter/setter pair inherited

```
runtimeType → Type
```

A representation of the runtime type of the object. no setter inherited

```
selectedCol ↔ int
```

Column of the currently selected square (-1 if none). getter/setter pair inherited

```
selectedRow ↔ int
```

Row of the currently selected square (-1 if none). getter/setter pair inherited

```
validMoves ↔ Map < int , List < int > >
```

Shared reference to the board's adjacency list. getter/setter pair inherited

```
wallColor ↔ String
```

Color identifier for this player's walls ('r' or 'b'). getter/setter pair inherited

```
wallPos ↔ Map < int , String >
```

Shared reference to placed wall positions. getter/setter pair inherited

Methods

```
addingWallEffect ( GameState currentGameState , int firstWall , int secondWall , int
middleWall , int offset ) → MoveData
```

Attempts to place a wall by removing edges from the graph, then validates paths. inherited

```
addSpecialCase ( GameState currentGameState , bool isMyRowMax ) → void
```

Temporarily adds jump or diagonal moves when facing the opponent. inherited

```
addWallsToList ( GameState currentGameState , int firstWall , int secondWall , bool isRow )
→ MoveData
```

Validates and places a wall spanning two adjacent segments. inherited

```
calculateHeuristic ( bool isMax , GameState current ) → double
```

Computes a weighted heuristic score for the given position.

```
generateAllStates ( GameState currentState ) → List < GameState >
```

Generates all legal successor states from the current position.

```
minimax ( GameState currentGameState , int currentLevel , int maxLevel , bool isMax ) →
GameState
```

Recursively evaluates a game state using minimax with alpha-beta pruning.

```
noSuchMethod ( Invocation invocation ) → dynamic
```

Invoked when a nonexistent method or property is accessed. inherited

```
play ( int row , int col , int oppCurrRow , int oppCurrCol ) → MoveData
```

Computes and executes the best move using minimax search. override

```
removeSpecialCases ( GameState currentGameState , int row , int col , bool isMyRowMax ) → void
```

Cleans up the temporary jump/diagonal edges added by addSpecialCase . inherited

```
toString ( ) → String
```

A string representation of this object. inherited

Game > Logic > Human > Player

Classes

```
HumanPlayer
```

Player implementation that handles human touch/click input.

Class: HumanPlayer class

Player implementation that handles human touch/click input. Manages two-tap wall placement and validates pawn movement based on tapped coordinates and current selection state.

Constructors

```
HumanPlayer ({ required Pawn pawn , required Map < int , List < int > > validMoves ,
required Map < int , String > wallPos , required String wallColor , required int
myInitRow , required int oppInitRow })
```

Properties

```
hashCode → int
```

The hash code for this object. no setter inherited

```
isSpecialAdded ↔ bool
```

Whether special jump/diagonal moves have been temporarily added. getter/setter pair inherited

```
myInitRow ↔ int
```

The starting row for this player (goal row for opponent). getter/setter pair inherited

```
numOfWalls ↔ int
```

Number of walls this player has left to place. getter/setter pair inherited

```
oppInitRow ↔ int
```

The starting row for the opponent (this player's goal row). getter/setter pair inherited

```
pawn ↔ Pawn
```

The pawn controlled by this player. getter/setter pair inherited

```
prevSelectedWall ↔ int ?
```

Previously selected wall segment index (for two-tap wall placement). getter/setter pair

```
runtimeType → Type
```

A representation of the runtime type of the object. no setter inherited

```
selectedCol ↔ int
```

Column of the currently selected square (-1 if none). getter/setter pair inherited

```
selectedRow ↔ int
```

Row of the currently selected square (-1 if none). getter/setter pair inherited

```
validMoves ↔ Map < int , List < int > >
```

Shared reference to the board's adjacency list. getter/setter pair inherited

```
wallColor ↔ String
```

Color identifier for this player's walls ('r' or 'b'). getter/setter pair inherited

```
wallPos ↔ Map < int , String >
```

Shared reference to placed wall positions. getter/setter pair inherited

Methods

```
addingWallEffect ( GameState currentGameState , int firstWall , int secondWall , int
middleWall , int offset ) → MoveData
```

Attempts to place a wall by removing edges from the graph, then validates paths. inherited

```
addSpecialCase ( GameState currentGameState , bool isMyRowMax ) → void
```

Temporarily adds jump or diagonal moves when facing the opponent. inherited

```
addWallsToList ( GameState currentGameState , int firstWall , int secondWall , bool isRow )
→ MoveData
```

Validates and places a wall spanning two adjacent segments. inherited

```
checkIfSelected ( int row , int col , int oppCurrRow , dynamic oppCurrCol ) → MoveData
```

Handles taps on playable squares (pawn selection and movement).

```
checkIfWallIsSelected ( int row , int col , int oppCurrRow , int oppCurrCol ) → MoveData
```

Handles taps on wall segments (two-tap wall placement).

```
noSuchMethod ( Invocation invocation ) → dynamic
```

Invoked when a nonexistent method or property is accessed. inherited

```
play ( int row , int col , int oppCurrRow , int oppCurrCol ) → MoveData
```

Processes a tap at the given board coordinates. override

```
removeSpecialCases ( GameState currentGameState , int row , int col , bool isMyRowMax ) → void
```

Cleans up the temporary jump/diagonal edges added by addSpecialCase . inherited

```
toString ( ) → String
```

A string representation of this object. inherited

Game > Logic > Player

Classes

```
Player
```

Base player class that holds shared pawn state, wall inventory, and move helpers.

Constants

```
boardCol
```

```
boardRow
```

```
kDebugMode
```

Class: Player class abstract

Base player class that holds shared pawn state, wall inventory, and move helpers. Provides core logic for wall placement validation and special jump/diagonal moves. Subclasses implement play to handle user input or AI decisions.

Constructors

```
Player ({ required Pawn pawn , required Map < int , List < int > > validMoves , required
Map < int , String > wallPos , required String wallColor , required int myInitRow ,
required int oppInitRow })
```

Creates a player with the given configuration.

Properties

```
hashCode → int
```

The hash code for this object. no setter inherited

```
isSpecialAdded ↔ bool
```

Whether special jump/diagonal moves have been temporarily added. getter/setter pair

```
myInitRow ↔ int
```

The starting row for this player (goal row for opponent). getter/setter pair

```
numOfWalls ↔ int
```

Number of walls this player has left to place. getter/setter pair

```
oppInitRow ↔ int
```

The starting row for the opponent (this player's goal row). getter/setter pair

```
pawn ↔ Pawn
```

The pawn controlled by this player. getter/setter pair

```
runtimeType → Type
```

A representation of the runtime type of the object. no setter inherited

```
selectedCol ↔ int
```

Column of the currently selected square (-1 if none). getter/setter pair

```
selectedRow ↔ int
```

Row of the currently selected square (-1 if none). getter/setter pair

```
validMoves ↔ Map < int , List < int > >
```

Shared reference to the board's adjacency list. getter/setter pair

```
wallColor ↔ String
```

Color identifier for this player's walls ('r' or 'b'). getter/setter pair

```
wallPos ↔ Map < int , String >
```

Shared reference to placed wall positions. getter/setter pair

Methods

```
addingWallEffect ( GameState currentGameState , int firstWall , int secondWall , int
middleWall , int offset ) → MoveData
```

Attempts to place a wall by removing edges from the graph, then validates paths.

```
addSpecialCase ( GameState currentGameState , bool isMyRowMax ) → void
```

Temporarily adds jump or diagonal moves when facing the opponent.

```
addWallsToList ( GameState currentGameState , int firstWall , int secondWall , bool isRow )
→ MoveData
```

Validates and places a wall spanning two adjacent segments.

```
noSuchMethod ( Invocation invocation ) → dynamic
```

Invoked when a nonexistent method or property is accessed. inherited

```
play ( int row , int col , int oppCurrRow , int oppCurrCol ) → MoveData
```

Executes a player action (pawn move or wall placement) at the given position.

```
removeSpecialCases ( GameState currentGameState , int row , int col , bool isMyRowMax ) → void
```

Cleans up the temporary jump/diagonal edges added by addSpecialCase .

```
toString ( ) → String
```

A string representation of this object. inherited

Helper > Game > State

Classes

```
GameState
```

Immutable snapshot of a board position used by both UI and minimax search.

Functions

```
maxState
```

Returns the GameState with the higher heuristic value.

```
minState
```

Returns the GameState with the lower heuristic value.

Class: GameState class

Immutable snapshot of a board position used by both UI and minimax search. Represents the complete state needed to evaluate moves and generate successors. The AI clones this state for each simulation to avoid mutating the live game.

Constructors

```
GameState ({ required Map < int , List < int > > validMoves , required Map < int , String > wallPos , required int myRow , required int myCol , required int oppRow , required int oppCol , required double alpha , required double beta , int numOfWalls = 10 , int oppNumOfWalls = 10 })
```

Creates a game state with the specified board configuration.

Properties

```
alpha ↔ double
```

Alpha value for alpha-beta pruning. getter/setter pair

```
beta ↔ double
```

Beta value for alpha-beta pruning. getter/setter pair

```
hashCode → int
```

The hash code for this object. no setter inherited

```
heuristicValue ↔ double ?
```

Computed score for this position (set by minimax). getter/setter pair

```
myCol ↔ int
```

Current column of the active player (0-16, even only). getter/setter pair

```
myRow ↔ int
```

Current row of the active player (0-16, even only). getter/setter pair

```
numOfWalls ↔ int
```

Number of walls remaining for the active player. getter/setter pair

```
oppCol ↔ int
```

Current column of the opponent (0-16, even only). getter/setter pair

```
oppNumOfWalls ↔ int
```

Number of walls remaining for the opponent. getter/setter pair

```
oppRow ↔ int
```

Current row of the opponent (0-16, even only). getter/setter pair

```
runtimeType → Type
```

A representation of the runtime type of the object. no setter inherited

```
validMoves ↔ Map < int , List < int > >
```

Adjacency list of valid pawn movements between cells. getter/setter pair

```
wallPos ↔ Map < int , String >
```

Maps cell indices to wall colors ('r' for red, 'b' for blue). getter/setter pair

Methods

```
clone ( { double ? heuristicOverride , double ? alphaOverride , double ? betaOverride }) →
GameState
```

Creates a deep copy of this game state for simulation.

```
noSuchMethod ( Invocation invocation ) → dynamic
```

Invoked when a nonexistent method or property is accessed. inherited

```
toString ( ) → String
```

A string representation of this object. inherited

Class: GameState class

Immutable snapshot of a board position used by both UI and minimax search. Represents the complete state needed to evaluate moves and generate successors. The AI clones this state for each simulation to avoid mutating the live game.

Constructors

```
GameState ({ required Map < int , List < int > > validMoves , required Map < int , String > wallPos , required int myRow , required int myCol , required int oppRow , required int oppCol , required double alpha , required double beta , int numOfWalls = 10 , int oppNumOfWalls = 10 })
```

Creates a game state with the specified board configuration.

Properties

```
alpha ↔ double
```

Alpha value for alpha-beta pruning. getter/setter pair

```
beta ↔ double
```

Beta value for alpha-beta pruning. getter/setter pair

```
hashCode → int
```

The hash code for this object. no setter inherited

```
heuristicValue ↔ double ?
```

Computed score for this position (set by minimax). getter/setter pair

```
myCol ↔ int
```

Current column of the active player (0-16, even only). getter/setter pair

```
myRow ↔ int
```

Current row of the active player (0-16, even only). getter/setter pair

```
numOfWalls ↔ int
```

Number of walls remaining for the active player. getter/setter pair

```
oppCol ↔ int
```

Current column of the opponent (0-16, even only). getter/setter pair

```
oppNumOfWalls ↔ int
```

Number of walls remaining for the opponent. getter/setter pair

```
oppRow ↔ int
```

Current row of the opponent (0-16, even only). getter/setter pair

```
runtimeType → Type
```

A representation of the runtime type of the object. no setter inherited

```
validMoves ↔ Map < int , List < int > >
```

Adjacency list of valid pawn movements between cells. getter/setter pair

```
wallPos ↔ Map < int , String >
```

Maps cell indices to wall colors ('r' for red, 'b' for blue). getter/setter pair

Methods

```
clone ( { double ? heuristicOverride , double ? alphaOverride , double ? betaOverride }) →
GameState
```

Creates a deep copy of this game state for simulation.

```
noSuchMethod ( Invocation invocation ) → dynamic
```

Invoked when a nonexistent method or property is accessed. inherited

```
toString ( ) → String
```

A string representation of this object. inherited

Class: GameState class

Immutable snapshot of a board position used by both UI and minimax search. Represents the complete state needed to evaluate moves and generate successors. The AI clones this state for each simulation to avoid mutating the live game.

Constructors

```
GameState ({ required Map < int , List < int > > validMoves , required Map < int , String >
wallPos , required int myRow , required int myCol , required int oppRow , required int
oppCol , required double alpha , required double beta , int numOfWalls = 10 , int
oppNumOfWalls = 10 })
```

Creates a game state with the specified board configuration.

Properties

```
alpha ↔ double
```

Alpha value for alpha-beta pruning. getter/setter pair

```
beta ↔ double
```

Beta value for alpha-beta pruning. getter/setter pair

```
hashCode → int
```

The hash code for this object. no setter inherited

```
heuristicValue ↔ double ?
```

Computed score for this position (set by minimax). getter/setter pair

```
myCol ↔ int
```

Current column of the active player (0-16, even only). getter/setter pair

```
myRow ↔ int
```

Current row of the active player (0-16, even only). getter/setter pair

```
numOfWalls ↔ int
```

Number of walls remaining for the active player. getter/setter pair

```
oppCol ↔ int
```

Current column of the opponent (0-16, even only). getter/setter pair

```
oppNumOfWalls ↔ int
```

Number of walls remaining for the opponent. getter/setter pair

```
oppRow ↔ int
```

Current row of the opponent (0-16, even only). getter/setter pair

```
runtimeType → Type
```

A representation of the runtime type of the object. no setter inherited

```
validMoves ↔ Map < int , List < int > >
```

Adjacency list of valid pawn movements between cells. getter/setter pair

```
wallPos ↔ Map < int , String >
```

Maps cell indices to wall colors ('r' for red, 'b' for blue). getter/setter pair

Methods

```
clone ( { double ? heuristicOverride , double ? alphaOverride , double ? betaOverride }) →
GameState
```

Creates a deep copy of this game state for simulation.

```
noSuchMethod ( Invocation invocation ) → dynamic
```

Invoked when a nonexistent method or property is accessed. inherited

```
toString ( ) → String
```

A string representation of this object. inherited

Class: GameState class

Immutable snapshot of a board position used by both UI and minimax search. Represents the complete state needed to evaluate moves and generate successors. The AI clones this state for each simulation to avoid mutating the live game.

Constructors

```
GameState ({ required Map < int , List < int > > validMoves , required Map < int , String > wallPos , required int myRow , required int myCol , required int oppRow , required int oppCol , required double alpha , required double beta , int numOfWalls = 10 , int oppNumOfWalls = 10 })
```

Creates a game state with the specified board configuration.

Properties

```
alpha ↔ double
```

Alpha value for alpha-beta pruning. getter/setter pair

```
beta ↔ double
```

Beta value for alpha-beta pruning. getter/setter pair

```
hashCode → int
```

The hash code for this object. no setter inherited

```
heuristicValue ↔ double ?
```

Computed score for this position (set by minimax). getter/setter pair

```
myCol ↔ int
```

Current column of the active player (0-16, even only). getter/setter pair

```
myRow ↔ int
```

Current row of the active player (0-16, even only). getter/setter pair

```
numOfWalls ↔ int
```

Number of walls remaining for the active player. getter/setter pair

```
oppCol ↔ int
```

Current column of the opponent (0-16, even only). getter/setter pair

```
oppNumOfWalls ↔ int
```

Number of walls remaining for the opponent. getter/setter pair

```
oppRow ↔ int
```

Current row of the opponent (0-16, even only). getter/setter pair

```
runtimeType → Type
```

A representation of the runtime type of the object. no setter inherited

```
validMoves ↔ Map < int , List < int > >
```

Adjacency list of valid pawn movements between cells. getter/setter pair

```
wallPos ↔ Map < int , String >
```

Maps cell indices to wall colors ('r' for red, 'b' for blue). getter/setter pair

Methods

```
clone ( { double ? heuristicOverride , double ? alphaOverride , double ? betaOverride }) →
GameState
```

Creates a deep copy of this game state for simulation.

```
noSuchMethod ( Invocation invocation ) → dynamic
```

Invoked when a nonexistent method or property is accessed. inherited

```
toString ( ) → String
```

A string representation of this object. inherited

Class: GameState class

Immutable snapshot of a board position used by both UI and minimax search. Represents the complete state needed to evaluate moves and generate successors. The AI clones this state for each simulation to avoid mutating the live game.

Constructors

```
GameState ({ required Map < int , List < int > > validMoves , required Map < int , String >
wallPos , required int myRow , required int myCol , required int oppRow , required int
oppCol , required double alpha , required double beta , int numOfWalls = 10 , int
oppNumOfWalls = 10 })
```

Creates a game state with the specified board configuration.

Properties

```
alpha ↔ double
```

Alpha value for alpha-beta pruning. getter/setter pair

```
beta ↔ double
```

Beta value for alpha-beta pruning. getter/setter pair

```
hashCode → int
```

The hash code for this object. no setter inherited

```
heuristicValue ↔ double ?
```

Computed score for this position (set by minimax). getter/setter pair

```
myCol ↔ int
```

Current column of the active player (0-16, even only). getter/setter pair

```
myRow ↔ int
```

Current row of the active player (0-16, even only). getter/setter pair

```
numOfWalls ↔ int
```

Number of walls remaining for the active player. getter/setter pair

```
oppCol ↔ int
```

Current column of the opponent (0-16, even only). getter/setter pair

```
oppNumOfWalls ↔ int
```

Number of walls remaining for the opponent. getter/setter pair

```
oppRow ↔ int
```

Current row of the opponent (0-16, even only). getter/setter pair

```
runtimeType → Type
```

A representation of the runtime type of the object. no setter inherited

```
validMoves ↔ Map < int , List < int > >
```

Adjacency list of valid pawn movements between cells. getter/setter pair

```
wallPos ↔ Map < int , String >
```

Maps cell indices to wall colors ('r' for red, 'b' for blue). getter/setter pair

Methods

```
clone ( { double ? heuristicOverride , double ? alphaOverride , double ? betaOverride }) → GameState
```

Creates a deep copy of this game state for simulation.

```
noSuchMethod ( Invocation invocation ) → dynamic
```

Invoked when a nonexistent method or property is accessed. inherited

```
toString ( ) → String
```

A string representation of this object. inherited

Class: GameState class

Immutable snapshot of a board position used by both UI and minimax search. Represents the complete state needed to evaluate moves and generate successors. The AI clones this state for each simulation to avoid mutating the live game.

Constructors

```
GameState ({ required Map < int , List < int > > validMoves , required Map < int , String > wallPos , required int myRow , required int myCol , required int oppRow , required int oppCol , required double alpha , required double beta , int numOfWalls = 10 , int oppNumOfWalls = 10 })
```

Creates a game state with the specified board configuration.

Properties

```
alpha ↔ double
```

Alpha value for alpha-beta pruning. getter/setter pair

```
beta ↔ double
```

Beta value for alpha-beta pruning. getter/setter pair

```
hashCode → int
```

The hash code for this object. no setter inherited

```
heuristicValue ↔ double ?
```

Computed score for this position (set by minimax). getter/setter pair

```
myCol ↔ int
```

Current column of the active player (0-16, even only). getter/setter pair

```
myRow ↔ int
```

Current row of the active player (0-16, even only). getter/setter pair

```
numOfWalls ↔ int
```

Number of walls remaining for the active player. getter/setter pair

```
oppCol ↔ int
```

Current column of the opponent (0-16, even only). getter/setter pair

```
oppNumOfWalls ↔ int
```

Number of walls remaining for the opponent. getter/setter pair

```
oppRow ↔ int
```

Current row of the opponent (0-16, even only). getter/setter pair

```
runtimeType → Type
```

A representation of the runtime type of the object. no setter inherited

```
validMoves ↔ Map < int , List < int > >
```

Adjacency list of valid pawn movements between cells. getter/setter pair

```
wallPos ↔ Map < int , String >
```

Maps cell indices to wall colors ('r' for red, 'b' for blue). getter/setter pair

Methods

```
clone ( { double ? heuristicOverride , double ? alphaOverride , double ? betaOverride }) →
GameState
```

Creates a deep copy of this game state for simulation.

```
noSuchMethod ( Invocation invocation ) → dynamic
```

Invoked when a nonexistent method or property is accessed. inherited

```
toString ( ) → String
```

A string representation of this object. inherited

Class: GameState class

Immutable snapshot of a board position used by both UI and minimax search. Represents the complete state needed to evaluate moves and generate successors. The AI clones this state for each simulation to avoid mutating the live game.

Constructors

```
GameState ({ required Map < int , List < int > > validMoves , required Map < int , String > wallPos , required int myRow , required int myCol , required int oppRow , required int oppCol , required double alpha , required double beta , int numOfWalls = 10 , int oppNumOfWalls = 10 })
```

Creates a game state with the specified board configuration.

Properties

```
alpha ↔ double
```

Alpha value for alpha-beta pruning. getter/setter pair

```
beta ↔ double
```

Beta value for alpha-beta pruning. getter/setter pair

```
hashCode → int
```

The hash code for this object. no setter inherited

```
heuristicValue ↔ double ?
```

Computed score for this position (set by minimax). getter/setter pair

```
myCol ↔ int
```

Current column of the active player (0-16, even only). getter/setter pair

```
myRow ↔ int
```

Current row of the active player (0-16, even only). getter/setter pair

```
numOfWalls ↔ int
```

Number of walls remaining for the active player. getter/setter pair

```
oppCol ↔ int
```

Current column of the opponent (0-16, even only). getter/setter pair

```
oppNumOfWalls ↔ int
```

Number of walls remaining for the opponent. getter/setter pair

```
oppRow ↔ int
```

Current row of the opponent (0-16, even only). getter/setter pair

```
runtimeType → Type
```

A representation of the runtime type of the object. no setter inherited

```
validMoves ↔ Map < int , List < int > >
```

Adjacency list of valid pawn movements between cells. getter/setter pair

```
wallPos ↔ Map < int , String >
```

Maps cell indices to wall colors ('r' for red, 'b' for blue). getter/setter pair

Methods

```
clone ( { double ? heuristicOverride , double ? alphaOverride , double ? betaOverride }) →
GameState
```

Creates a deep copy of this game state for simulation.

```
noSuchMethod ( Invocation invocation ) → dynamic
```

Invoked when a nonexistent method or property is accessed. inherited

```
toString ( ) → String
```

A string representation of this object. inherited

Helper > Helper > Func

Functions

`bfs`

Finds the shortest path from a starting position to any cell in the target row.

`calculateRowCol`

Converts a flattened board index into (row, col) coordinates.

`copyWallPos`

Creates a shallow copy of the wall positions map.

`deepCopyValidMoves`

Creates a deep copy of the validMoves adjacency list.

Helper > Move > Data

Enums

`MoveData`

Results for a player's attempted action.

Main

Classes

```
MyApp
```

Root widget of the Quoridor application.

Functions

```
main
```

Entry point for the Quoridor game application.

Class: MyApp class

Root widget of the Quoridor application. Sets up the app theme and navigation starting with StartScreen .

Constructors

```
MyApp ({ Key ? key })
```

const

Properties

```
hashCode → int
```

The hash code for this object. no setter inherited

```
key → Key ?
```

Controls how one widget replaces another widget in the tree. final inherited

```
runtimeType → Type
```

A representation of the runtime type of the object. no setter inherited

Methods

```
build ( BuildContext context ) → Widget
```

Describes the part of the user interface represented by this widget. override

```
createElement ( ) → StatelessElement
```

Creates a StatelessElement to manage this widget's location in the tree. inherited

```
debugDescribeChildren ( ) → List < DiagnosticsNode >
```

Returns a list of DiagnosticsNode objects describing this node's children. inherited

```
debugFillProperties ( DiagnosticPropertiesBuilder properties ) → void
```

Add additional properties associated with the node. inherited

```
noSuchMethod ( Invocation invocation ) → dynamic
```

Invoked when a nonexistent method or property is accessed. inherited

```
toDiagnosticsNode ( { String ? name , DiagnosticsTreeStyle ? style }) → DiagnosticsNode
```

Returns a debug representation of the object that is used by debugging tools and by DiagnosticsNode.toStringDeep . inherited

```
toString ( { DiagnosticLevel minLevel = DiagnosticLevel.info }) → String
```

A string representation of this object. inherited

```
toStringDeep ( { String prefixLineOne = '' , String ? prefixOtherLines , DiagnosticLevel
minLevel = DiagnosticLevel.debug , int wrapWidth = 65 }) → String
```

Returns a string representation of this node and its descendants. inherited

```
toStringShallow ( { String joiner = ', ' , DiagnosticLevel minLevel =
DiagnosticLevel.debug }) → String
```

Returns a one-line detailed description of the object. inherited

```
toStringShort ( ) → String
```

A short, textual description of this widget. inherited

Views > Pages > Board

Classes

`Board`

Main board widget hosting game state, turn handling, and rendering.

Constants

`boardCol`

`boardRow`

`initCol`

`myInitRow`

`myWallColor`

`oppInitRow`

`oppWallColor`

Class: Board class

Main board widget hosting game state, turn handling, and rendering. Manages the complete game loop including player turns, move validation, win detection, and UI updates. Creates and coordinates player instances.

Constructors

`Board ({ Key ? key , required bool isOppHuman })`

const

Properties

`hashCode → int`

The hash code for this object. no setter inherited

`isOppHuman → bool`

Whether the opponent is human (true) or AI (false). final

```
key → Key ?
```

Controls how one widget replaces another widget in the tree. final inherited

```
runtimeType → Type
```

A representation of the runtime type of the object. no setter inherited

Methods

```
createElement ( ) → StatefulElement
```

Creates a StatefulElement to manage this widget's location in the tree. inherited

```
createState ( ) → State < Board >
```

Creates the mutable state for this widget at a given location in the tree. override

```
debugDescribeChildren ( ) → List < DiagnosticsNode >
```

Returns a list of DiagnosticsNode objects describing this node's children. inherited

```
debugFillProperties ( DiagnosticPropertiesBuilder properties ) → void
```

Add additional properties associated with the node. inherited

```
noSuchMethod ( Invocation invocation ) → dynamic
```

Invoked when a nonexistent method or property is accessed. inherited

```
toDiagnosticsNode ( { String ? name , DiagnosticsTreeStyle ? style }) → DiagnosticsNode
```

Returns a debug representation of the object that is used by debugging tools and by DiagnosticsNode.toStringDeep . inherited

```
toString ( { DiagnosticLevel minLevel = DiagnosticLevel.info }) → String
```

A string representation of this object. inherited

```
toStringDeep ( { String prefixLineOne = '' , String ? prefixOtherLines , DiagnosticLevel
minLevel = DiagnosticLevel.debug , int wrapWidth = 65 }) → String
```

Returns a string representation of this node and its descendants. inherited

```
toStringShallow ( { String joiner = ', ' , DiagnosticLevel minLevel =
DiagnosticLevel.debug }) → String
```

Returns a one-line detailed description of the object. inherited

```
toStringShort ( ) → String
```

A short, textual description of this widget. inherited

Views > Pages > Pawn

Classes

```
Pawn
```

Represents a player's pawn on the board.

Class: Pawn class

Represents a player's pawn on the board. Stores position and visual properties for rendering.

Constructors

```
Pawn ({ required bool isWhite , required String imagePath , required int currRow , required int currCol })
```

Creates a new pawn with the given properties.

Properties

```
currCol ↔ int
```

Current column position (0-16, even numbers only). getter/setter pair

```
currRow ↔ int
```

Current row position (0-16, even numbers only). getter/setter pair

```
hashCode → int
```

The hash code for this object. no setter inherited

```
imagePath → String
```

Asset path to the pawn image. final

```
isWhite → bool
```

Whether this pawn belongs to the white (bottom) player. final

```
runtimeType → Type
```

A representation of the runtime type of the object. no setter inherited

Methods

```
noSuchMethod ( Invocation invocation ) → dynamic
```

Invoked when a nonexistent method or property is accessed. inherited

```
toString ( ) → String
```

A string representation of this object. inherited

Views > Pages > Scoreboard > Walls

Classes

```
ScoreboardWalls
```

Displays remaining walls for each side and whose turn it is.

Class: ScoreboardWalls class

Displays remaining walls for each side and whose turn it is. Shows a colored indicator for the active player and wall counts for both human and opponent.

Constructors

```
ScoreboardWalls ({ Key ? key , required int myNumOfWalls , required int oppNumOfWalls , required bool isMyTurn })
```

Creates a scoreboard widget with the given game state. const

Properties

```
hashCode → int
```

The hash code for this object. no setter inherited

```
isMyTurn → bool
```

Whether it's currently the human player's turn. final

```
key → Key ?
```

Controls how one widget replaces another widget in the tree. final inherited

```
myNumOfWalls → int
```

Number of walls remaining for the human player. final

```
oppNumOfWalls → int
```

Number of walls remaining for the opponent. final

```
runtimeType → Type
```

A representation of the runtime type of the object. no setter inherited

Methods

```
build ( BuildContext context ) → Widget
```

Describes the part of the user interface represented by this widget. override

```
createElement ( ) → StatelessElement
```

Creates a StatelessElement to manage this widget's location in the tree. inherited

```
debugDescribeChildren ( ) → List < DiagnosticsNode >
```

Returns a list of DiagnosticsNode objects describing this node's children. inherited

```
debugFillProperties ( DiagnosticPropertiesBuilder properties ) → void
```

Add additional properties associated with the node. inherited

```
noSuchMethod ( Invocation invocation ) → dynamic
```

Invoked when a nonexistent method or property is accessed. inherited

```
toDiagnosticsNode ( { String ? name , DiagnosticsTreeStyle ? style }) → DiagnosticsNode
```

Returns a debug representation of the object that is used by debugging tools and by DiagnosticsNode.toStringDeep . inherited

```
toString ( { DiagnosticLevel minLevel = DiagnosticLevel.info }) → String
```

A string representation of this object. inherited

```
toStringDeep ( { String prefixLineOne = '' , String ? prefixOtherLines , DiagnosticLevel
minLevel = DiagnosticLevel.debug , int wrapWidth = 65 }) → String
```

Returns a string representation of this node and its descendants. inherited

```
toStringShallow ( { String joiner = ', ' , DiagnosticLevel minLevel =
DiagnosticLevel.debug }) → String
```

Returns a one-line detailed description of the object. inherited

```
toStringShort ( ) → String
```

A short, textual description of this widget. inherited

Views > Pages > Square

Classes

```
Square
```

Renders a playable square on the board and highlights selection/valid moves.

Class: Square class

Renders a playable square on the board and highlights selection/valid moves. Displays a pawn image if occupied, and changes color to indicate selection or valid move destinations.

Constructors

```
Square ({ Key ? key , required Pawn ? piece , required bool isValidMove , required bool isSelected , required void onTapFunc () })
```

Creates a square widget with the given properties. const

Properties

```
hashCode → int
```

The hash code for this object. no setter inherited

```
isSelected → bool
```

Whether this square is currently selected by the player. final

```
isValidMove → bool
```

Whether this square is a valid move destination for the selected pawn. final

```
key → Key ?
```

Controls how one widget replaces another widget in the tree. final inherited

```
onTapFunc → void Function ()
```

Callback invoked when the square is tapped. final

```
piece → Pawn ?
```

The pawn occupying this square (null if empty). final

```
runtimeType → Type
```

A representation of the runtime type of the object. no setter inherited

Methods

```
build ( BuildContext context ) → Widget
```

Describes the part of the user interface represented by this widget. override

```
createElement ( ) → StatelessElement
```

Creates a StatelessElement to manage this widget's location in the tree. inherited

```
debugDescribeChildren ( ) → List < DiagnosticsNode >
```

Returns a list of DiagnosticsNode objects describing this node's children. inherited

```
debugFillProperties ( DiagnosticPropertiesBuilder properties ) → void
```

Add additional properties associated with the node. inherited

```
noSuchMethod ( Invocation invocation ) → dynamic
```

Invoked when a nonexistent method or property is accessed. inherited

```
toDiagnosticsNode ( { String ? name , DiagnosticsTreeStyle ? style }) → DiagnosticsNode
```

Returns a debug representation of the object that is used by debugging tools and by DiagnosticsNode.toStringDeep . inherited

```
toString ( { DiagnosticLevel minLevel = DiagnosticLevel.info }) → String
```

A string representation of this object. inherited

```
toStringDeep ( { String prefixLineOne = '' , String ? prefixOtherLines , DiagnosticLevel
minLevel = DiagnosticLevel.debug , int wrapWidth = 65 }) → String
```

Returns a string representation of this node and its descendants. inherited

```
toStringShallow ( { String joiner = ', ' , DiagnosticLevel minLevel =
DiagnosticLevel.debug }) → String
```

Returns a one-line detailed description of the object. inherited

```
toStringShort ( ) → String
```

A short, textual description of this widget. inherited

Views > Pages > Wall

Classes

```
Wall
```

Renders a wall segment cell and shows ownership color.

Class: Wall class

Renders a wall segment cell and shows ownership color. Wall segments are displayed in red (human) or blue (opponent) when placed, otherwise in the default background color.

Constructors

```
Wall ({ Key ? key , required bool isWallSelected , required String ? wallColortxt , required void onTapFunc () })
```

Creates a wall widget with the given properties. const

Properties

```
hashCode → int
```

The hash code for this object. no setter inherited

```
isWallSelected → bool
```

Whether a wall has been placed at this position. final

```
key → Key ?
```

Controls how one widget replaces another widget in the tree. final inherited

```
onTapFunc → void Function ()
```

Callback invoked when this wall segment is tapped. final

```
runtimeType → Type
```

A representation of the runtime type of the object. no setter inherited

```
wallColortxt → String ?
```

Color identifier ('r' for red, 'b' for blue) if wall is placed. final

Methods

```
build ( BuildContext context ) → Widget
```

Describes the part of the user interface represented by this widget. override

```
createElement ( ) → StatelessElement
```

Creates a StatelessElement to manage this widget's location in the tree. inherited

```
debugDescribeChildren ( ) → List < DiagnosticsNode >
```

Returns a list of DiagnosticsNode objects describing this node's children. inherited

```
debugFillProperties ( DiagnosticPropertiesBuilder properties ) → void
```

Add additional properties associated with the node. inherited

```
noSuchMethod ( Invocation invocation ) → dynamic
```

Invoked when a nonexistent method or property is accessed. inherited

```
toDiagnosticsNode ( { String ? name , DiagnosticsTreeStyle ? style }) → DiagnosticsNode
```

Returns a debug representation of the object that is used by debugging tools and by DiagnosticsNode.toStringDeep . inherited

```
toString ( { DiagnosticLevel minLevel = DiagnosticLevel.info }) → String
```

A string representation of this object. inherited

```
toStringDeep ( { String prefixLineOne = '' , String ? prefixOtherLines , DiagnosticLevel
minLevel = DiagnosticLevel.debug , int wrapWidth = 65 }) → String
```

Returns a string representation of this node and its descendants. inherited

```
toStringShallow ( { String joiner = ', ' , DiagnosticLevel minLevel =
DiagnosticLevel.debug }) → String
```

Returns a one-line detailed description of the object. inherited

```
toStringShort ( ) → String
```

A short, textual description of this widget. inherited

Views > Screens > Game > Screen

Classes

```
GameScreen
```

Main game screen that hosts the board for either human vs human or vs computer.

Class: GameScreen class

Main game screen that hosts the board for either human vs human or vs computer. Wraps the Board widget in a Scaffold with an app bar showing the game title.

Constructors

```
GameScreen ({ Key ? key , required bool isOppHuman })
```

const

Properties

```
hashCode → int
```

The hash code for this object. no setter inherited

```
isOppHuman → bool
```

Whether the opponent is human (true) or AI (false). final

```
key → Key ?
```

Controls how one widget replaces another widget in the tree. final inherited

```
runtimeType → Type
```

A representation of the runtime type of the object. no setter inherited

Methods

```
createElement ( ) → StatefulElement
```

Creates a StatefulElement to manage this widget's location in the tree. inherited

```
createState ( ) → State < GameScreen >
```

Creates the mutable state for this widget at a given location in the tree. override

```
debugDescribeChildren ( ) → List < DiagnosticsNode >
```

Returns a list of DiagnosticsNode objects describing this node's children. inherited

```
debugFillProperties ( DiagnosticPropertiesBuilder properties ) → void
```

Add additional properties associated with the node. inherited

```
noSuchMethod ( Invocation invocation ) → dynamic
```

Invoked when a nonexistent method or property is accessed. inherited

```
toDiagnosticsNode ( { String ? name , DiagnosticsTreeStyle ? style }) → DiagnosticsNode
```

Returns a debug representation of the object that is used by debugging tools and by DiagnosticsNode.toStringDeep . inherited

```
toString ( { DiagnosticLevel minLevel = DiagnosticLevel.info }) → String
```

A string representation of this object. inherited

```
toStringDeep ( { String prefixLineOne = '' , String ? prefixOtherLines , DiagnosticLevel
minLevel = DiagnosticLevel.debug , int wrapWidth = 65 }) → String
```

Returns a string representation of this node and its descendants. inherited

```
toStringShallow ( { String joiner = ', ' , DiagnosticLevel minLevel =
DiagnosticLevel.debug }) → String
```

Returns a one-line detailed description of the object. inherited

```
toStringShort ( ) → String
```

A short, textual description of this widget. inherited

Views > Screens > Start > Screen

Classes

```
StartScreen
```

Entry screen to select opponent type and start a new game.

Constants

```
computer
```

```
human
```

Class: StartScreen class

Entry screen to select opponent type and start a new game. Displays a menu allowing the user to choose between playing against another human or against the AI.

Constructors

```
StartScreen ({ Key ? key })
```

const

Properties

```
hashCode → int
```

The hash code for this object. no setter inherited

```
key → Key ?
```

Controls how one widget replaces another widget in the tree. final inherited

```
runtimeType → Type
```

A representation of the runtime type of the object. no setter inherited

Methods

```
createElement ( ) → StatefulElement
```

Creates a StatefulElement to manage this widget's location in the tree. inherited

```
createState ( ) → State < StartScreen >
```

Creates the mutable state for this widget at a given location in the tree. override

```
debugDescribeChildren ( ) → List < DiagnosticsNode >
```

Returns a list of DiagnosticsNode objects describing this node's children. inherited

```
debugFillProperties ( DiagnosticPropertiesBuilder properties ) → void
```

Add additional properties associated with the node. inherited

```
noSuchMethod ( Invocation invocation ) → dynamic
```

Invoked when a nonexistent method or property is accessed. inherited

```
toDiagnosticsNode ( { String ? name , DiagnosticsTreeStyle ? style }) → DiagnosticsNode
```

Returns a debug representation of the object that is used by debugging tools and by DiagnosticsNode.toStringDeep . inherited

```
toString ( { DiagnosticLevel minLevel = DiagnosticLevel.info }) → String
```

A string representation of this object. inherited

```
toStringDeep ( { String prefixLineOne = '' , String ? prefixOtherLines , DiagnosticLevel
minLevel = DiagnosticLevel.debug , int wrapWidth = 65 }) → String
```

Returns a string representation of this node and its descendants. inherited

```
toStringShallow ( { String joiner = ', ' , DiagnosticLevel minLevel =
DiagnosticLevel.debug }) → String
```

Returns a one-line detailed description of the object. inherited

```
toStringShort ( ) → String
```

A short, textual description of this widget. inherited