Data Model Documentation

Data Model Detail

This document provides an overview of the data model. For simpler and more focused reports, simply copy this initial template and turn off the sections not required.

Class Model

Type: **Package**
Package: Model
Detail: *Created on 19/03/2012. Last modified on 19/03/2012.*
Notes:

Class Model
Created By: juliano on 19/11/2005
Last Modified: 25/06/2012, *Version:*1.0

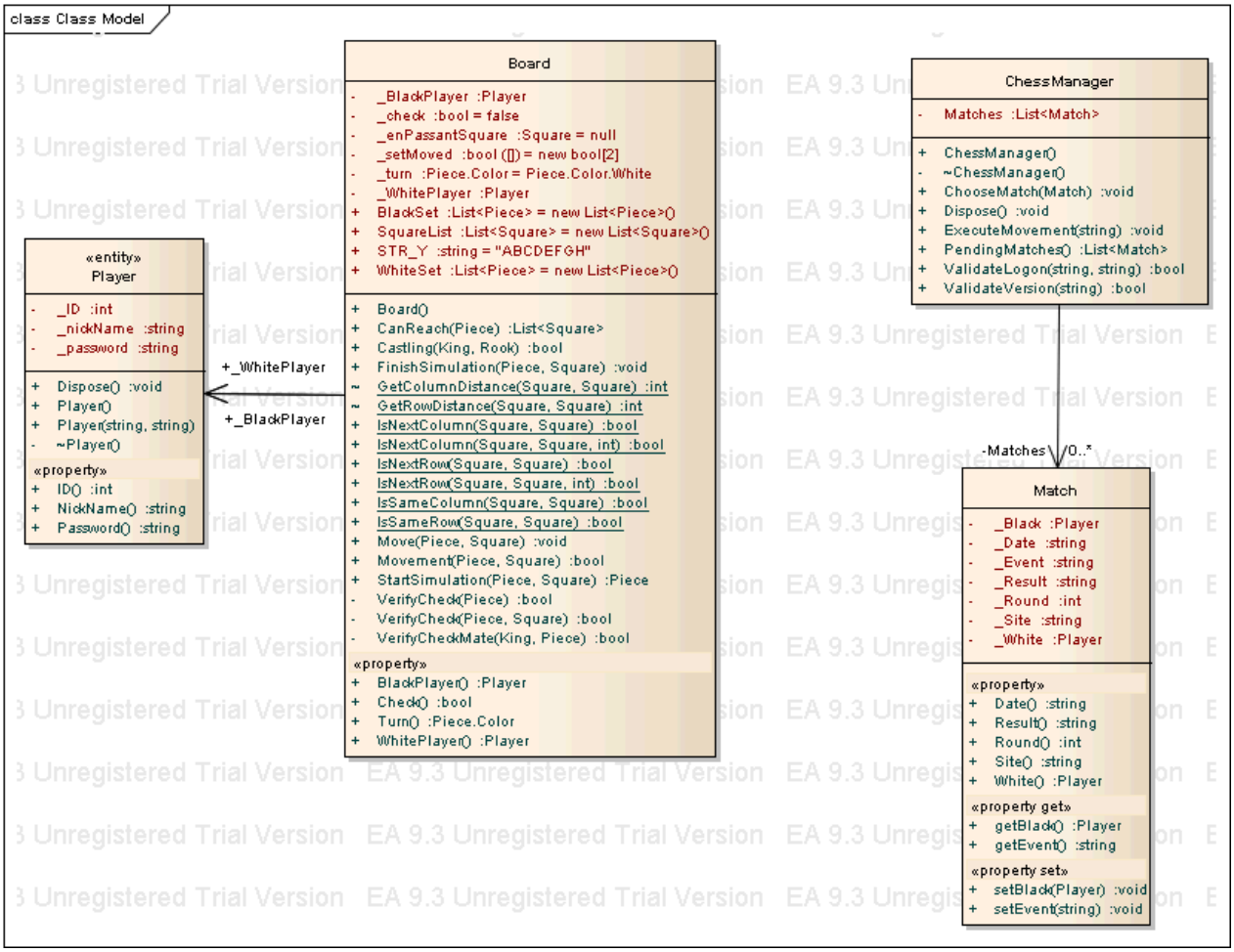

Figure: 1

Engine Class Model

Created By: juliano on 29/03/2012
Last Modified: 25/06/2012, *Version:*1.0

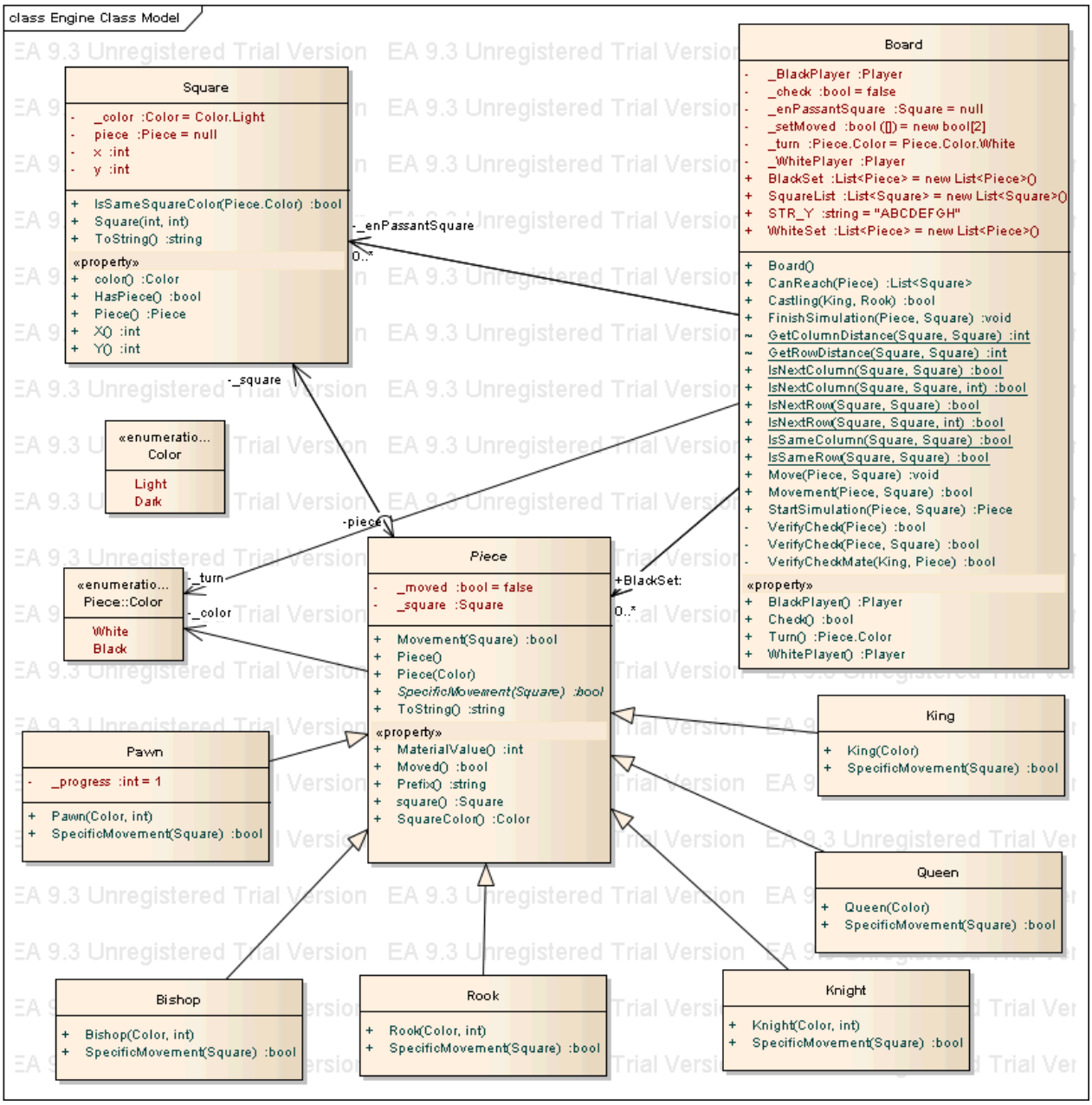

Figure: 2

Bishop

Database: C#, *Stereotype:* , *Package:* Class Model
Detail: *Created on* 29/03/2012. *Last modified on* 24/05/2012.
Notes: This is a kind of piece that can´t sit on a square of different color from its original one.

Constraints

Name	Type	Columns	Initial Code	Notes
Bishop	Public	color number		
SpecificMovement	Public	targetSquare		

Relationships

Columns	Association	Notes
	Bishop. Piece.	

Board

Database: C#, *Stereotype:* , *Package:* Class Model
Detail: *Created on* 29/03/2012. *Last modified on* 22/06/2012.
Notes: This class is a composition of an eight-by-eight squares table. This class is responsible to create the instances of the squares and two sets of pieces. Besides, it is also responsible to place all those pieces created on its starting point position. When the game starts, it also manages the following items: The movement of the pieces, by calling the Movement method of the specific piece instance and then watching for interaction among the pieces.

Columns

PK	Name	Type	Not Null	Unique	Len	Prec	Scale	Init	Notes
	_BlackPlayer	Player							
	_check	bool			0			false	
	_enPassantSquare	Square			0			null	
	_setMoved	bool			0			new bool[2]	
	_turn	Piece.Color			0			Piece.Color.White	
	_WhitePlayer	Player							
	BlackSet	List<Piece>			0			new List<Piece>()	
	SquareList	List<Square>			0			new List<Square>()	Initializes an empty list of squares
	STR_Y	string			0			"ABCDEFGH"	
	WhiteSet	List<Piece>			0			new List<Piece>()	Initilizes two sets of pieces

Constraints

Name	Type	Columns	Initial Code	Notes
BlackPlayer	Public			
Board	Public			Default constructor that creates an eight-by-eight square table and places two sets of

Name	Type	Columns	Initial Code	Notes
				pieces.
CanReach	Public	piece		Method that returns a list of squares that can be reach by a specific piece. @returns
Castling	Public	king rook		Performs the castling movement, if possible @returns True if the castling was performed
Check	Public			
FinishSimulation	Public	piece square		
GetColumnDistance	Internal	square1 square2		
GetRowDistance	Internal	square1 square2		
IsNextColumn	Public	square1 square2		
IsNextColumn	Public	square1 square2 distance		
IsNextRow	Public	square1 square2		
IsNextRow	Public	square1 square2 distance		
IsSameColumn	Public	square1 square2		
IsSameRow	Public	square1 square2		
Move	Public	piece square		Performs the movement of a piece to the square.
Movement	Public	piece square		Validates a movement of a piece to the square.
StartSimulation	Public	piece square		Method responsible for making an actual move for the specified piece, keeping a bookmark

Name	Type	Columns	Initial Code	Notes
				for the former position. Every call for this method must be followed by a corresponding call for FinishSimulation method. Besides, it must be guaranteed by a transaction context, assuring that the movement can be rolled back, in order to avoid damage to the game integrity. It is recommended that all treatment between the calls should be inside an exception context. @returns
Turn	Public			
VerifyCheck	Private	piece		This method can be called after the commit of the movement @returns
VerifyCheck	Private	piece square		 @returns
VerifyCheckMate	Private	king piece		This method can be called after any method of check verification has returned true @returns
WhitePlayer	Public			

Relationships

Columns	Association	Notes
	Board. Square._enPassantSquare	
	Board. Color._turn	
	Board. **0..*** Piece.BlackSet	
	Board. **0..*** Square.SquareList	
	Board. **0..*** Piece.WhiteSet	
	Board.	

Columns	Association	Notes
	Player._BlackPlayer	
	Board. Player._WhitePlayer	
Movement(Piece, Square)	Game Screen. Board.	
SpecificMovement(Square)	Board. Piece.	
SpecificMovement(Square)	Board. <anonymous>.	
Move(Piece, Square)	Game Screen. Board.	
Move(Piece, Square)	Game Screen. Board.	

ChessManager

Database: C#, *Stereotype:* , *Package:* Class Model
Detail: *Created on* 20/06/2012. *Last modified on* 25/06/2012.
Notes:

Columns

PK	Name	Type	Not Null	Unique	Len	Prec	Scale	Init	Notes
	Matches	List<Match>			0				

Constraints

Name	Type	Columns	Initial Code	Notes
ChessManager	Public			
~ChessManager	Private			
ChooseMatch	Public	match		@param ="Match"
Dispose	Public			
ExecuteMovement	Public	movement		
PendingMatches	Public			
ValidateLogon	Public	user password		
ValidateVersion	Public	version		

Relationships

Columns	Association	Notes
	ChessManager. **0..*** Match.Matches	

Color

Database: C#, *Stereotype:* «enumeration», *Package:* Class Model
Detail: *Created on* 29/03/2012. *Last modified on* 29/03/2012.
Notes:

Columns

PK	Name	Type	Not Null	Unique	Len	Prec	Scale	Init	Notes
	Light				0				
	Dark				0				

King

Database: C#, *Stereotype:* , *Package:* Class Model
Detail: *Created on* 29/03/2012. *Last modified on* 24/05/2012.
Notes: This is the most important piece in the game. This kind of piece participates in a special movement called castling

Constraints

Name	Type	Columns	Initial Code	Notes
King	Public	color		
SpecificMovement	Public	targetSquare		

Relationships

Columns	Association	Notes
	King. Piece.	

Knight

Database: C#, *Stereotype:* , *Package:* Class Model
Detail: *Created on* 29/03/2012. *Last modified on* 24/05/2012.
Notes: This is the only piece in the game that is able to jump over the others

Constraints

Name	Type	Columns	Initial Code	Notes
Knight	Public	color number		
SpecificMovement	Public	targetSquare		

Relationships

Columns	Association	Notes
	Knight. Piece.	

Match

Database: C#, *Stereotype:* , *Package:* Class Model
Detail: *Created on* 22/06/2012. *Last modified on* 26/06/2012.
Notes:

Columns

PK	Name	Type	Not Null	Unique	Len	Prec	Scale	Init	Notes
	_Black	Player							
	_Date	string							
	_Event	string							
	_Result	string							
	_Round	int							
	_Site	string							

	_White	Player							

Constraints

Name	Type	Columns	Initial Code	Notes
Date	Public			
getBlack	Public			
getEvent	Public			
Result	Public			
Round	Public			
setBlack	Public	newVal		
setEvent	Public	newVal		
Site	Public			
White	Public			

Relationships

Columns	Association	Notes
	ChessManager. **0..*** Match.Matches	

Pawn

Database: C#, *Stereotype:* , *Package:* Class Model
Detail: *Created on* 29/03/2012. *Last modified on* 24/05/2012.
Notes: This is the most ordinary piece in the game.

Columns

PK	Name	Type	Not Null	Unique	Len	Prec	Scale	Init	Notes
	_progress	int			0			1	

Constraints

Name	Type	Columns	Initial Code	Notes
Pawn	Public	color number		
SpecificMovement	Public	targetSquare		

Relationships

Columns	Association	Notes
	Pawn. Piece.	

Piece

Database: C#, *Stereotype:* , *Package:* Class Model
Detail: *Created on* 29/03/2012. *Last modified on* 24/05/2012.
Notes: This is an abstract class that represents a generic piece. It can be inherited to implement specific behaviors of that kind of piece. It cannot be created. It refers to a square where it sits. When it does not refers to a square, that means it´s out of the board.

Columns

PK	Name	Type	Not Null	Unique	Len	Prec	Scale	Init	Notes
	_moved	bool			0			false	private Color _color;

	_square	Square			0				

Constraints

Name	Type	Columns	Initial Code	Notes
MaterialValue	Public			
Moved	Public			
Movement	Public	targetSquare		Movement constraints are checked out in the pieces, since there are specific things to consider, for example, how many squares this piece is supposed to move. Every class related to a piece of the chess must implement this method and should call this one. @returns
Piece	Public			int _value;
Piece	Public	color		Constructor able to receive the color
Prefix	Public			
SpecificMovement	Public	targetSquare		
square	Public			
SquareColor	Public			
ToString	Public			

Relationships

Columns	Association	Notes
	Piece. Color._color	
	Rook. Piece.	
	Knight. Piece.	
	Queen. Piece.	
	King. Piece.	
	Pawn. Piece.	
	Bishop. Piece.	
	Board. **0..*** Piece.BlackSet	
	Board. **0..*** Piece.WhiteSet	
	Square. Piece.piece	
	Piece.	

Columns	Association	Notes
	Square._square	
SpecificMovement(Square)	Board. Piece.	

Color

Database: C#, *Stereotype:* «enumeration», *Package:* Class Model
Detail: *Created on* 29/03/2012. *Last modified on* 24/05/2012.
Notes:

Columns

PK	Name	Type	Not Null	Unique	Len	Prec	Scale	Init	Notes
	White				0				
	Black				0				

Relationships

Columns	Association	Notes
	Piece. Color._color	
	Board. Color._turn	

Player

Database: C#, *Stereotype:* «entity», *Package:* Class Model
Detail: *Created on* 29/03/2012. *Last modified on* 22/06/2012.
Notes:

Columns

PK	Name	Type	Not Null	Unique	Len	Prec	Scale	Init	Notes
	_ID	int							
	_nickName	string							
	_password	string							

Constraints

Name	Type	Columns	Initial Code	Notes
Dispose	Public			
ID	Public			
NickName	Public			
Password	Public			
Player	Public			
Player	Public	nickName password		
~Player	Private			

Relationships

Columns	Association	Notes
	Board. Player._BlackPlayer	
	Board. Player._WhitePlayer	

Queen

Database: C#, *Stereotype:* , *Package:* Class Model

Detail: *Created on* 29/03/2012. *Last modified on* 24/05/2012.
Notes: This is the second most important piece in the game.

Constraints

Name	Type	Columns	Initial Code	Notes
Queen	Public	color		
SpecificMovement	Public	targetSquare		

Relationships

Columns	Association	Notes
	Queen. Piece.	

Rook

Database: C#, *Stereotype:* , *Package:* Class Model
Detail: *Created on* 29/03/2012. *Last modified on* 25/06/2012.
Notes: This is the third most important piece in the game. This kind of piece participates in a special movement called castling

Constraints

Name	Type	Columns	Initial Code	Notes
Rook	Public	color number		
SpecificMovement	Public	targetSquare		

Relationships

Columns	Association	Notes
	Rook. Piece.	

Square

Database: C#, *Stereotype:* , *Package:* Class Model
Detail: *Created on* 29/03/2012. *Last modified on* 24/05/2012.
Notes: This is a class that compose a board. Each one has its fixed, immutable position, and refers or not to a single volatile piece in the game. When it does not refers to a piece, it means it´s empty.

Columns

PK	Name	Type	Not Null	Unique	Len	Prec	Scale	Init	Notes
	_color	Color			0			Color.Light	This is not a relevant information, it can only be useful for the Bishop
	piece	Piece			0			null	
	x	int			0				represents the horizontal position of the square in the board

	y	int			0				represents the vertical position of the square in the board

Constraints

Name	Type	Columns	Initial Code	Notes
color	Public			This is not a relevant information, it can only be useful for the Bishop
HasPiece	Public			
IsSameSquareColor	Public	squareColor		
Piece	Public			A piece placed in this instance of Square
Square	Public	x y		
ToString	Public			Overrides the method in order to return a chess notation known internationally for the square @returns
X	Public			Represents a reference to the alphabetic sequence in the chess board
Y	Public			Represents a reference to the numeric sequence in the chess board

Relationships

Columns	Association	Notes
	Square. Color._color	
	Board. Square._enPassantSquare	
	Board. **0..*** Square.SquareList	
	Square. Piece.piece	
	Piece. Square._square	

Color

Database: C#, *Stereotype:* «enumeration», *Package:* Class Model
Detail: *Created on* 29/03/2012. *Last modified on* 24/05/2012.
Notes:

Columns

PK	Name	Type	Not Null	Unique	Len	Prec	Scale	Init	Notes
	Light				0				
	Dark				0				

Relationships

Columns	Association	Notes
	Square. Color._color	