Class Dictionary

Jean-Philippe Miguel-Gagnon, Jérémy Gaouette, Raphaël Rail

Thursday, 31th of march 2022

Presented to : Charles Jacob

Contents

1 Introduction

The goal of this document is to inform the programmer about classes used to create a C# OOP Chess game.

We'll go through this with the MVC model approch to make it clearer for the programmer where to implement his code.

2 Models

2.1 Classe 'Piece'

This is an abstract class for a base piece for the game.

Piece
-_colour : Colour
+Get et Set =>_colour
+CanCollide : bool
+ValidMove(int x1, int y1, int x2, int y2) : bool
+ToString()

2.2 Fields

Field Name	Type	Visibility
_colour	Colour	Private

Description : This field represent the colour a piece, wich can only be black or white as it is for a regular chess game.

2.3 Methods

Method Name	Parameters	Returned Type	Visibility
ValidMove	x1, y1, x2, y2 : int	bool	Public

Parameters : x1 ans y1 represent the coordinates of the position before a possible move and x2 and y2 are the coordinates of the position after the move. These are all of Integer(16) type.
Description : The method calculates if a move is valid and returns true in this case.

Method Name	Parameters	Returned Type	Visibility
CanCollide	None	bool	Public

Parameters : x1 ans y1 represent the coordinates of the position before a possible move and x2 and y2 are the coordinates of the position after the
Description :

Descrptions :

Method Name	Description
GetColor	Getter for the color of a piece
SetColor	Setter for the color of a piece
CanColide	Returns true if the piece can collide
ValidMove	Returns true if the piece is asked to do a valid move
ToString	Methode to be override by children to show all properties of a specific piece