

# Class Dictionary

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

Thursday, 31th of march 2022



Presented to : Charles Jacob

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Models</b>	<b>3</b>
2.1	Classe ‘Piece’ . . . . .	3
2.2	Fields . . . . .	3
2.3	Methods . . . . .	3
2.4	Properties . . . . .	4

# 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 approach 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
+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 and 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 returns true if the piece can move to the 2nd position.

Method Name	Parameters	Returned Type	Visibility
CanCollide	None	bool	Public

**Description :** Returns true if the piece can't go over other pieces, otherwise it returns false.

Method Name	Parameters	Returned Type	Visibility
ToString	None	String	Public

**Description :** Returns true if the piece can't go over other pieces, otherwise it returns false.

## 2.4 Properties

Property Name	Parameters	Returned Type	Visibility
Colour	None	Colour	Public

**Description :** Returns de colour of a piece.