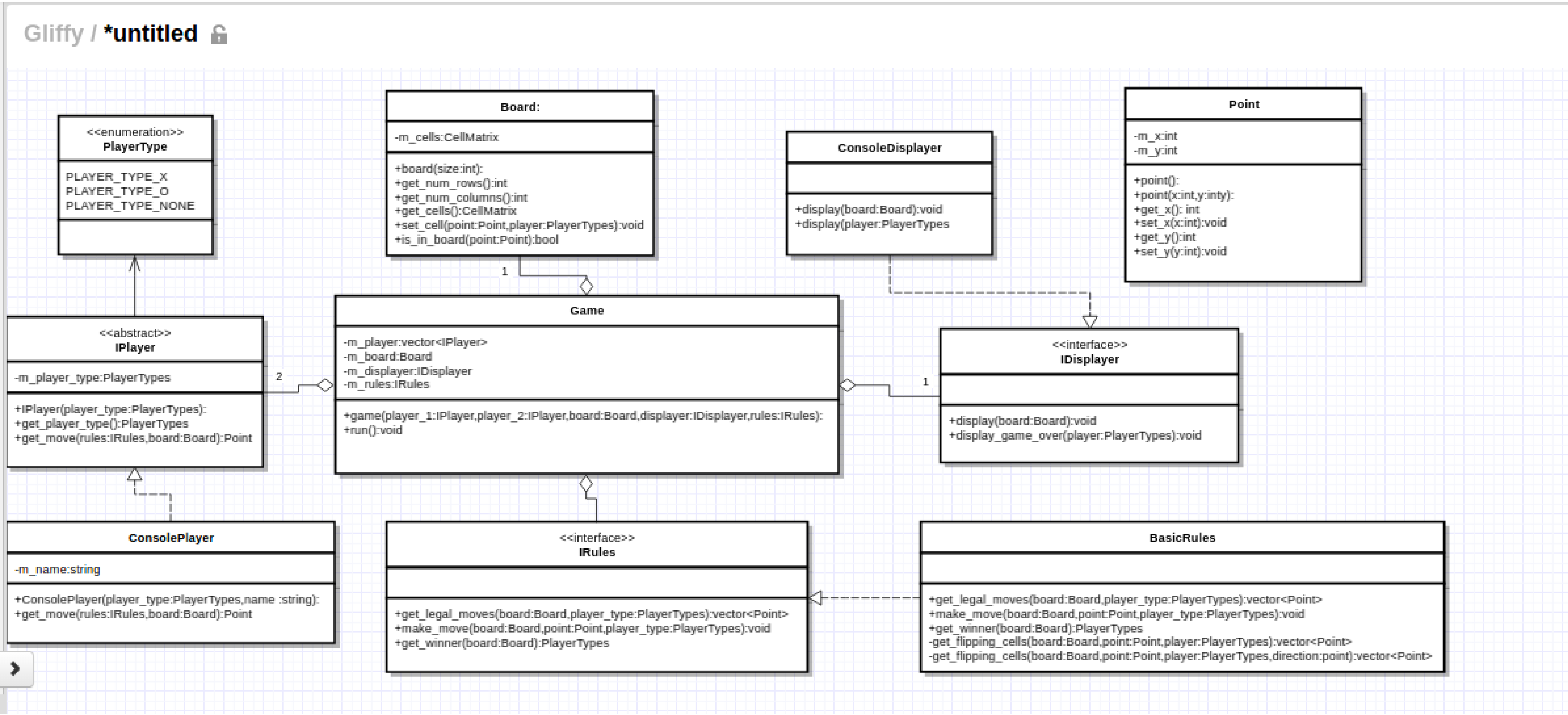

Gliffy / ***untitled**

<<enumeration>>
PlayerType

PLAYER_TYPE_X
PLAYER_TYPE_O
PLAYER_TYPE_NONE

Board:

-m_cells:CellMatrix

+board(size:int):
+get_num_rows():int
+get_num_columns():int
+get_cells():CellMatrix
+set_cell(point:Point,player:PlayerTypes):void
+is_in_board(point:Point):bool

ConsoleDisplayer

+display(board:Board):void
+display(player:PlayerTypes

Point

-m_x:int
-m_y:int

+point():
+point(x:int,y:inty):
+get_x(): int
+set_x(x:int):void
+get_y():int
+set_y(y:int):void

<<abstract>>
IPlayer

-m_player_type:PlayerTypes

+IPlayer(player_type:PlayerTypes):
+get_player_type():PlayerTypes
+get_move(rules:IRules,board:Board):Point

Game

-m_player:vector<IPlayer>
-m_board:Board
-m_displayer:IDisplayer
-m_rules:IRules

+game(player_1:IPlayer,player_2:IPlayer,board:Board,displayer:IDisplayer,rules:IRules):
+run():void

<<interface>>
IDisplayer

+display(board:Board):void
+display_game_over(player:PlayerTypes):void

ConsolePlayer

-m_name:string

+ConsolePlayer(player_type:PlayerTypes,name :string):
+get_move(rules:IRules,board:Board):Point

<<interface>>
IRules

+get_legal_moves(board:Board,player_type:PlayerTypes):vector<Point>
+make_move(board:Board,point:Point,player_type:PlayerTypes):void
+get_winner(board:Board):PlayerTypes

BasicRules

+get_legal_moves(board:Board,player_type:PlayerTypes):vector<Point>
+make_move(board:Board,point:Point,player_type:PlayerTypes):void
+get_winner(board:Board):PlayerTypes
-get_flipping_cells(board:Board,point:Point,player:PlayerTypes):vector<Point>
-get_flipping_cells(board:Board,point:Point,player:PlayerTypes,direction:point):vector<Point>