

```
# TTT_player.py - Handles Tic Tac Toe Players
# Ari Cohen

# quarto_player.py
# Sean Straw & Ari Cohen

import random # used for creating random moves
from TTT_interface import *
from TTT_state import *

class GamePlayer():
    HUMAN = 0
    COMPUTER = 1

    MAXIMIZE = 1
    MINIMIZE = -1

    def __init__(self):
        self.type = GamePlayer.HUMAN
        self.level = 0
        self.time_limit = 10000000

    def get_type(self):
        return self.type

    def set_type(self, new_type):
        self.type = new_type

    def set_level(self, level):
        self.level = level

    def get_move(self, game_state):
        if self.type == GamePlayer.HUMAN:
            return get_human_move(game_state)
        else:
            return get_computer_move(game_state, self.level)

def get_players_info(player0, player1, interface_state):
    players = [player0, player1]
    data = get_players_information(interface_state)
    for index in range(2):
        if data[index*2] == "h":
            players[index].set_type(GamePlayer.HUMAN)
        else:
            players[index].set_type(GamePlayer.COMPUTER)
            players[index].set_level(data[(index*2)+1])
    return

def get_computer_move(game_state, level):
    return get_random_move(game_state)

def get_random_move(game_state):
    move = GameMove()
    good_squares = get_good_squares(game_state)
    chosen_square = random.choice(good_squares)
```

```
move.set_move(chosen_square[0], chosen_square[1], game_state.  
    get_current_piece())  
return [move, GameState.PLAYING]  
  
def get_good_squares(game_state):  
    all_squares = game_state.get_squares()  
    good_squares = []  
    for piece in range(len(all_squares)):  
        if(all_squares[piece] == GameState.EMPTY):  
            good_squares.append([piece/3, piece%3])  
    return good_squares
```