CSC 4343 Homework 3 (due May 1st by end of day)

You may work in a group of 2 people for this homework.

If you form a group, only one person needs to submit the code in moodle. At the beginning of the code file, put the names of the group members in a comment line.

Your task in this homework is to implement and train a neural network that can play the Gomoku game.

The Python implementation of the game is in gomoku.py and gamegui.py provides a GUI for displaying the game as well as a player class that takes user inputs to make a move. "gomoku.py" also provides a player that chooses moves randomly.

You should implement a player (class) that uses your neural network to determine and return a move when the current configuration of the game board is given as the input. The player should also have a function that returns the keras neural network model it uses. Follow the template below to implement your player class (the class should have implementation for these functions):

```
class MyPlayer:
    def __init__(self, id):
        self.id = id  #player 0 is the player that makes the first move.
        # add your neural network initialization here


    def get_move(self, board):
    # Given the current board configuration, return the move the player want
    to make. "board" is a (binary) numpy array of the size [2, board_height,
    board_width]. (We fix board height and width to be 11.) Player 0's pieces
    are marked in board[0] and player 1's pieces are in board[1]. You need to
    return x, y (the row and col index) of the position where you want to put
    your piece. Make sure your move is legal, i.e., at location (x, y), there
    is no piece on the current board.
        return x, y


    def get_model(self):
    # Return the keras neural network model you use.
        return m
```

Neural Network Structure and Training:

Different from HW1&2, we don't fix the NN structure you should use. The game board can be viewed as an image. Therefore, you may use some type of CNN of your choice.

For training the neural network, you can follow any of the reinforcement learning approaches we discuss in class. You can also train the NN any other way you prefer. You may even generate your own data to train the neural network. The training process is up to your choice. We only require that the moves of your player should be based on a keras neural network model.

Homework Submission:

Upload a Python file named **Yourlastname.py** (change Yourlastname to your actual last name. note the capitalization of the first letter.) The MyPlayer class implementation should be in the file. Same as HW1&2, we'll not train your neural network model from scratch. You should train the model as part of the homework. In the constructor of your player class, you should load a trained model from a model file named **Yourlastname.h5**. You should not upload the .h5 file of the trained model to moodle. Instead, you should share it in your google drive and put the share link in a comment line at the beginning of the .py file. (If you have a group, the names of the group members should be on the first comment line and the link should be on the second line.) Your code should work when the .py file and the model file are in the same directory.

Player Score

We will test your player against the following players in games:

Test 1.	Win against random players (players that put pieces at random position available)
Test 2.	Each other

Each pair of players will play multiple (>10) games. A player beats the other if it wins a majority over these games. A player passes Test 1 will get 90 points. We will hold a tournament (Test 2) for the players that pass test 1. (A win in a tournament game will give a score 1, a loss, -1 and a tie 0.) The players with the top 5 scores will receive the remaining 10 points. (If less than 5 players have scores strictly better than the rest, only those with strictly better scores will receive the points.)

We will run the players using the code similar to the following:

- Two NN players compete against each other:

```
import Zhang
import Chen

g = Gomoku()
p1 = Zhang.MyPlayer(0)
p2 = Chen.MyPlayer(1)
for i in range(10):
    g.play(p1, p2)
    print(i, g.result)
    g.reset()
```

- An NN player plays against a random player:

```
import Zhang

g = Gomoku()
p1 = Zhang.MyPlayer(0)
p2 = RandomPlayer(1)
g.play(p1, p2)
print(g.result)
```