FACULTY OF ENGINEERING, AIN-SHAMS UNIVERSITY

Connect-Four game

Artificial Intelligence Term Project

Aliaa Mahmoud Ahmed, Class 2
Mostafa Tawfiq Mohammed, Class 3
Mostafa Atef Abd-Allah, Class 3
Mostafa Fahmy Al-Noqrashy, Class 3

5/16/2019

This document presents a simulation of the game, Connect-Four, showing the main function used by the AI side to predict the right moves to guarantee either a winning state or a defensive state. The project is submitted as an artificial intelligence term project, spring 2019, at the Faculty of Engineering, Ain-Shams University.

1 Introduction

Connect-Four is a widely played game. The winner is the one who can space four discs of the same color adjacently either in a vertical, horizontal, or even a diagonal row.

This game software is developed by the programming language python with the use of **OpenCV library** for the **GUI**. It allows a player to play against the computer with three levels of difficulty.

The full source code is available on GitHub and it is open for any further improvements.

github.com/MustafaFahmy93/Connect-4-implementation-in-python

2 Utility Function

The utility function, ***SCORE***, used in the AI module is a simple function that evaluates a score for each and every possible move that can be played from the computer side.

Once the AI turn starts, it starts computing a weight for all possible moves according to the following criteria:

- If there is an empty space with other three adjacent vertical, horizontal, or diagonal empty spaces; ***the score is increased by one.***
- If there is one disc with three adjacent vertical, horizontal, or diagonal empty spaces; ***the score is increased by two.***
- If there are two discs with two adjacent vertical, horizontal, or diagonal empty spaces; ***the score is increased by three.***
- If there are three discs with one adjacent vertical, horizontal, or diagonal empty space; It is a winning move so ***the score is increased by three.***

So far, this algorithm can only detect the best way to win, yet it cannot prevent the player from winning, thus, an improvement to the score function has been developed as follows:

- If there is one player-disc with three adjacent vertical, horizontal, or diagonal empty spaces; ***the score is decreased by two.***
- If there are two player-disc s with two adjacent vertical, horizontal, or diagonal empty spaces; ***the score is decreased by three.***
- If there are three player-discs with one adjacent vertical, horizontal, or diagonal empty space; It is a winning move so ***the score is decreased by three.***

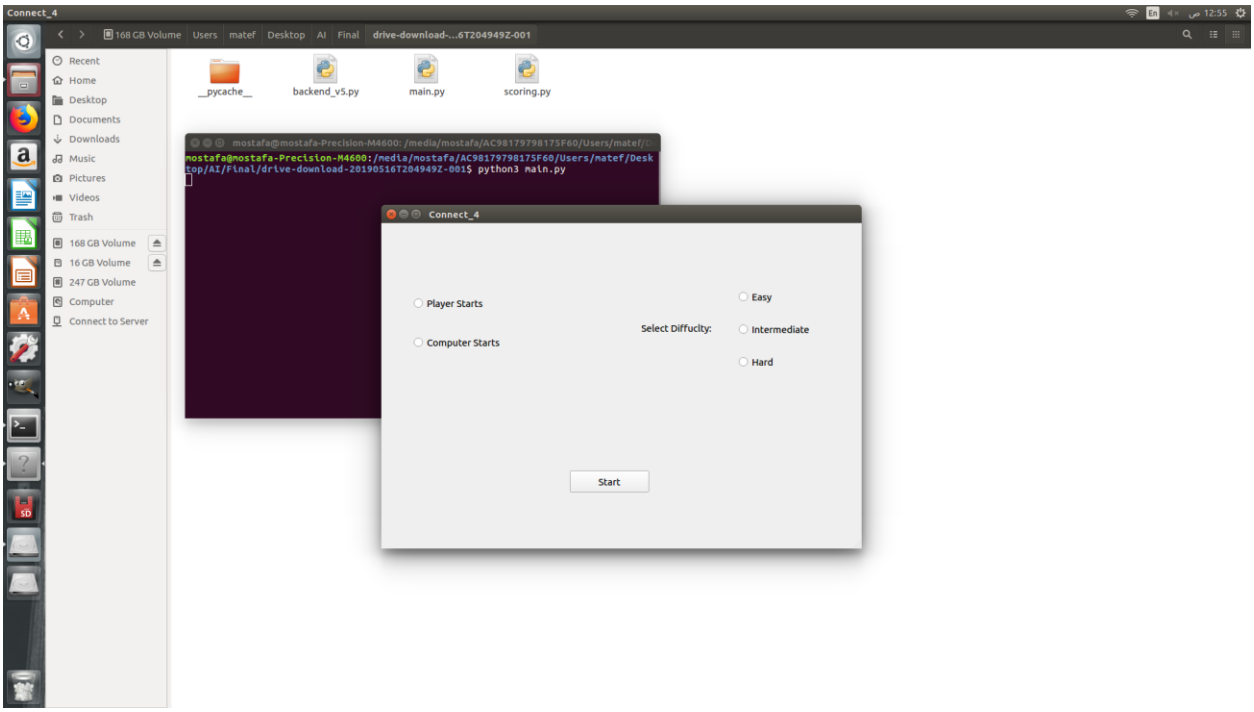
3 Computations Flow

The ***AlphaBeta*** function takes a root node represents a state of the board, it starts finding a sequence of moves according to the **depth** given to it at the function call, then it starts to evaluate the score as explained in the previous section, and finally it starts recursion to propagate to another state, if the value of **Alpha** is **equal** to or **greater than** the value of **Beta**, a **pruning** of all other siblings of the current node occurs.

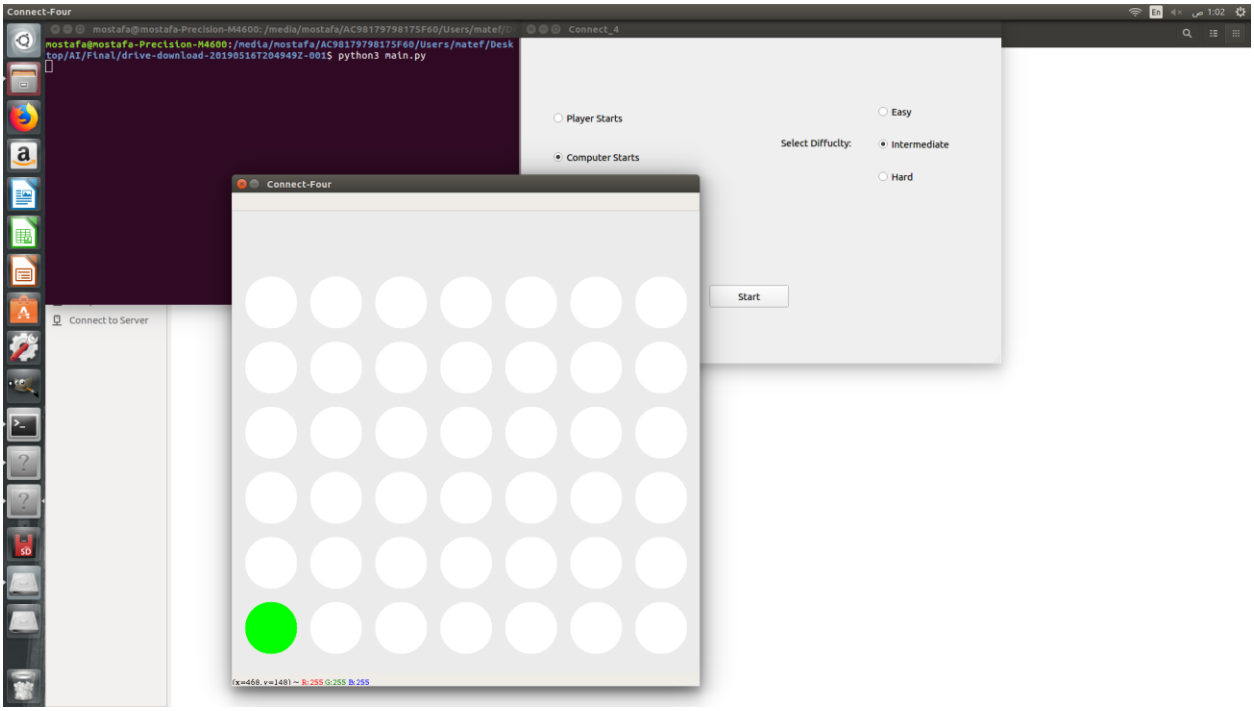
4 User Guide

In this section you will find a guide to play the game without any technical details

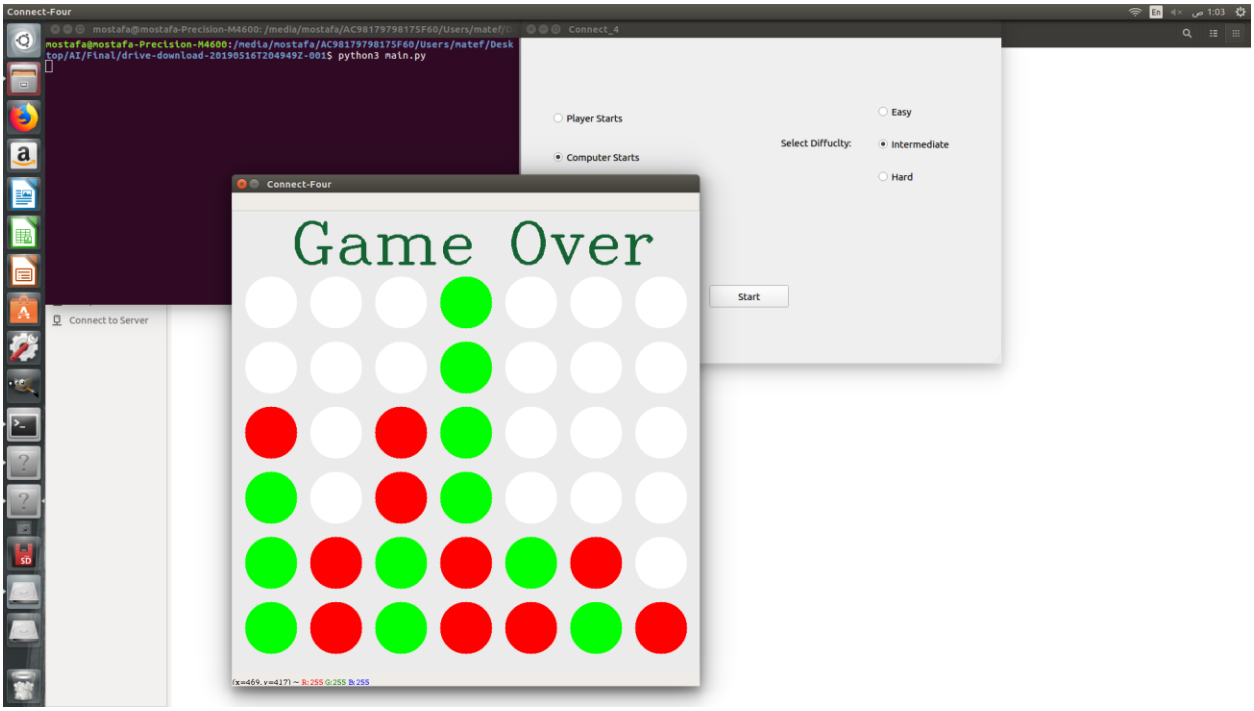
Once you double click the executable file, a window will pop up asking you to set the game conditions.



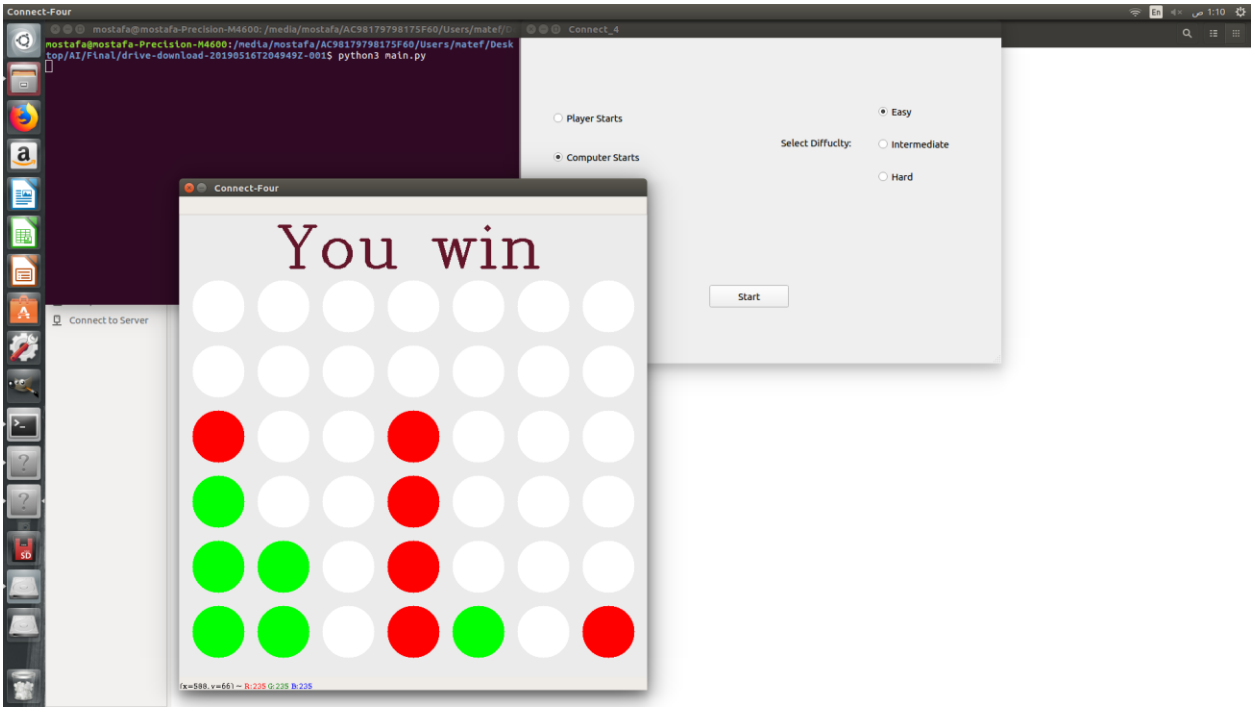
After choosing the needed conditions another window will pop with the game itself.



Single left mouse button click to choose which column exactly you shall play in, play until you lose,



or until you win.



5 Work Split

GUI: Aliaa Mahmoud, Mostafa Tawfiq, and Mostafa Atef.

AlphaBeta function: Mostafa Fahmy, Mostafa Tawfiq.

Score function: Aliaa Mahmoud, and Mostafa Fahmy.

Other backend functions: Mostafa Atef, Mostafa Fahmy, and Aliaa Mahmoud.

Project document: Mostafa Atef.

Video: Aliaa Mahmoud.