Project Title:

Implementation of Multiple Tic-Tac-Toe Agents

Spring 2021

Team Members:

1. Praveen Mandadi
2. Raja Muppalla

Description of Project

The game is played in a 3x3 Grid between any two AI Agents at the same time.

The game is to be played between two people.

One of the player chooses 'O' and the other 'X' to mark their respective cells.

When any agent completes such a sequence, or when agents fill up the grid without reaching any pattern, the game ends.

Objective of the Agents

The goal for each agent is to win the game by forming a horizontal, vertical, or diagonal line of all X OR all O in a grid in which each agent plays one after the other.

The second goal is to ensure that your adversary is unable to create an X OR O pattern since this is a zero-sum game.

Statement of Project objectives

We're creating four AI agents who will play the game as guests.

The user can choose which two agents will compete in a Tic-Tac-Toe tournament.

The logic of the game is Agent 1 wins against Agent 2 or Agent 2 wins against Agent 1or Tie between Agent 1 and Agent 2.

APPROACH

Algorithms Used :

Adversarial search Algorithms

1. Minimax algorithm using traditional approach.
2. Minimax algorithm using alpha-beta pruning approach.
3. Expectimax algorithm.

Reinforcement Learning Algorithm

1. Q-learning

Deliverable s

- Documentation report(README.md)
- Developed python programming Algorithms(.py files)
- GitHub repository link
- YouTube video
- PPT slides

Evaluation methodology

01

Agents should be conditioned so that the game's performance is as accurate as if it were played by two human brains.

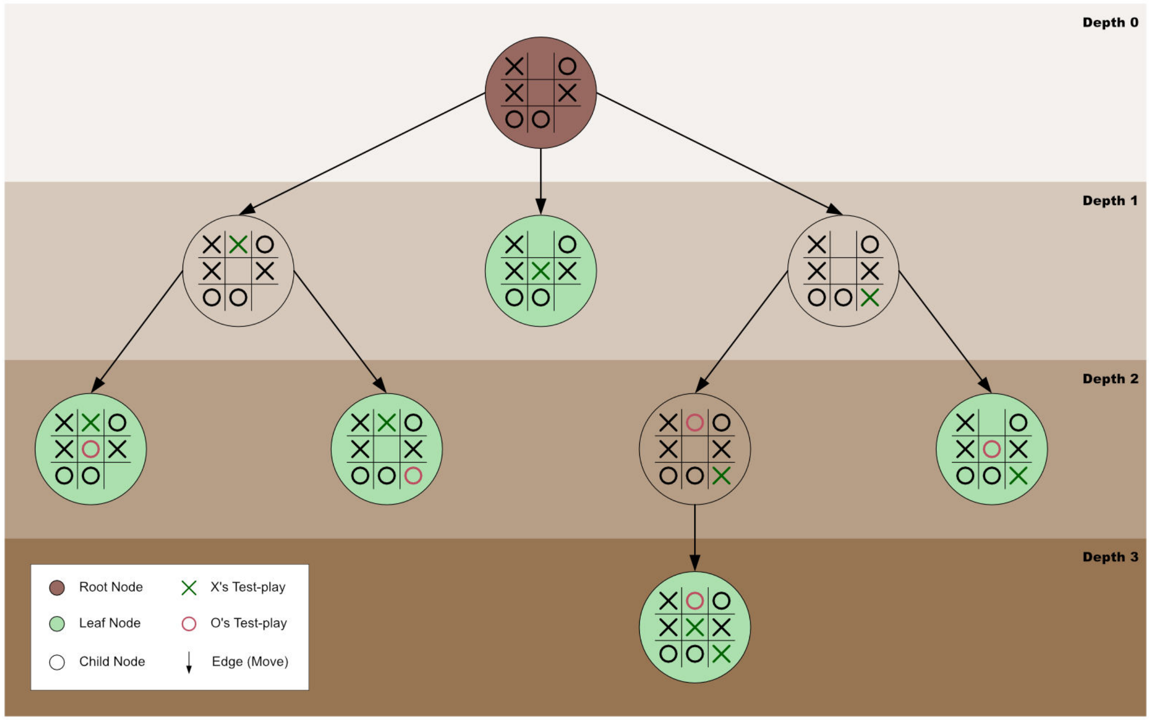
02

The success of the project is determined by the successful implementation of four AI algorithms.

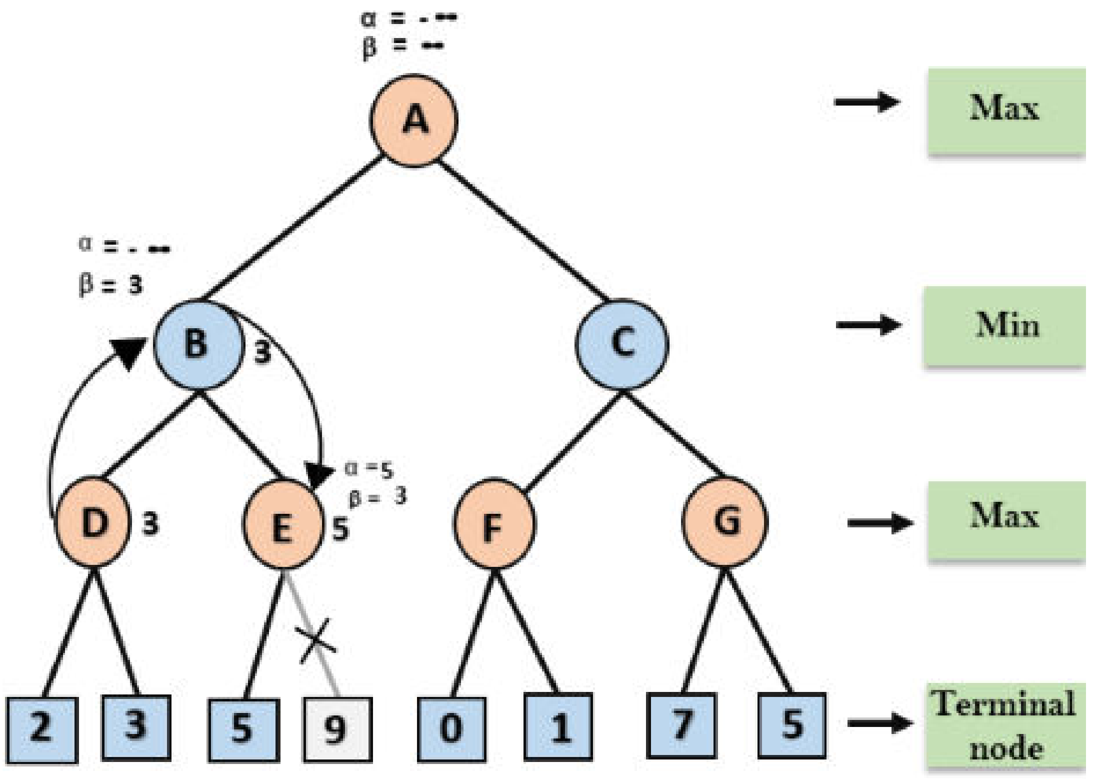
03

While making the next move, none of the agents can freeze or struggle.

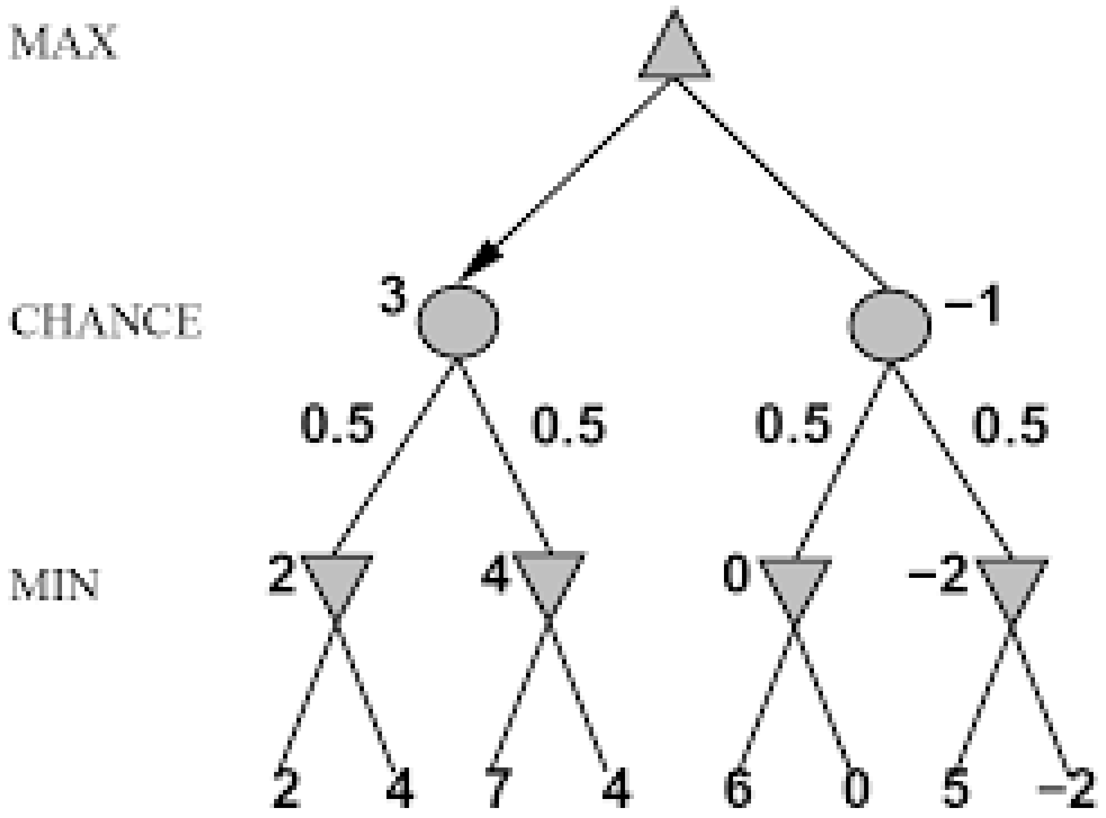
Minimax graph



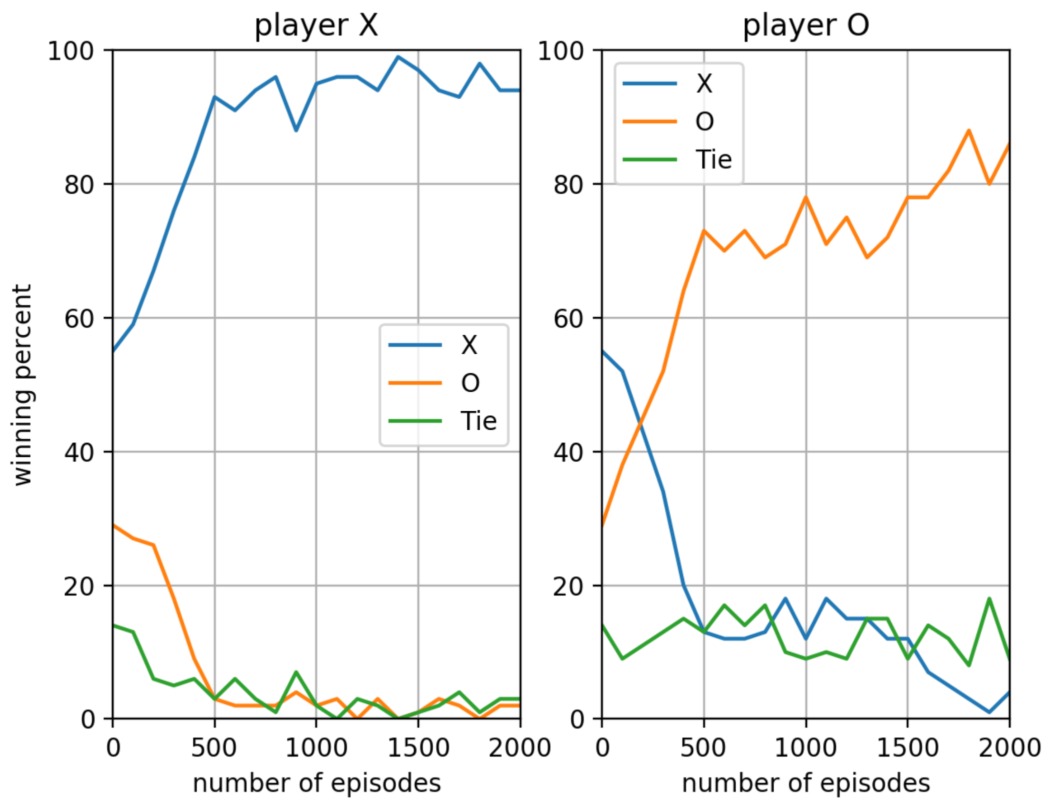
Minimax Alpha-Beta graph



Expectimax graph



Q-Learning graph



Code Explanation

```
irror_mod = modifier_ob.
 mirror object to mirror_
irror_mod.mirror_object =
operation == "MIRROR_X":
irror_mod.use_x = True
irror_mod.use_y = False
irror_mod.use_z = False
_operation == "MIRROR_Y":
irror_mod.use_x = False
irror_mod.use_y = True
irror_mod.use_z = False
_operation == "MIRROR_Z":
irror_mod.use_x = False
irror_mod.use_y = False
irror_mod.use_z = True

election at the end -add
_ob.select= 1
er_ob.select=1
ntext.scene.objects.active
("Selected" + str(modifier_
irror_ob.select = 0
 bpy.context.selected_obj
ata.objects[one.name].sel

int("please select exactl

- OPERATOR CLASSES ----

types.Operator):
 X mirror to the selected
ject.mirror_mirror_x"
ror X"

ontext):
ext.active_object is not
```

Thank You