

Exploring and Expanding the World of Artificial Intelligence

Ridip De
B.Tech, 4th Year,
Computer Science and
Engineering,
SRM University, India

R.P. Mahapatra, PhD
Professor and HOD,
Computer Science and
Engineering,
SRM University, India

Partha Sarathi Chakraborty
Assistant Professor, Member
IEEE,
Computer Science and
Engineering,
SRM University, India

ABSTRACT

This work is an effort to explore the available applications of Artificial Intelligence, in conjugation with other computing applications and put them to interesting and meaningful use through simple modeling. Tic-Tac-Toe model is considered as a suitable candidate for this experiment due to its simple yet versatile facets.

General Terms

Artificial Intelligence, Speech Recognition, Text-to-Speech (TTS), Tic-Tac-Toe Algorithms and Modes.

Keywords

AI, Speech Recognition, Text-to-Speech, Algorithm, Tic-Tac-Toe modes, Smart Tic-Tac-Toe.

1. INTRODUCTION

Artificial intelligence can be well-defined as the capability of a digital computer to execute tasks commonly associated with intelligence of the human beings. Artificially Intelligent systems are endowed with the intellectual process traits of humans. Such systems have the ability to think, reason out their action, learn from past experience, ascertain meaning or generalize. Humans have always been aiming to develop systems with human-level intelligence.

McCarthy, the father of AI, coined the term in 1955. He defined Artificial intelligence as "the science and engineering of making intelligent machines".

Artificial Intelligence can be used in various aspects of day to day life be it stock trading, remote sensing, maintaining law, scientific discovery, medical diagnosis etc. One of the major uses of AI can be easing how humans interact with the computer which not only does make the human life easier but would also help the differently-abled persons to interact with today's high end machines.

To demonstrate the above mentioned major use, Tic-Tac-Toe model can be considered as a suitable candidate for this experiment due to its simple yet versatile facets. There are a couple of ways of developing the algorithm to simulate the algorithm for "Tic-Tac-Toe". Our approach has adopted to aim at the following:

- Explore the available options under Artificial Intelligence which can be put to interesting and meaningful use.
- Designing a user friendly user interface.
- To help differently abled interact with computers.

2. STRATEGIES

Contrary to a popular belief, tic-tac-toe isn't purely a game of chance. There are some strategies that can enhance the chances of victory. The first most move will decide the destiny of the game. The following points describe in brief about the various strategies that one must keep in mind before making the first move:

- The first move should be in one of the four corners. Doing so will reduce the predictive-ness of the game by the opponent.
- Avoid the edges as a first move (edges are the four boxes that are neither the center nor the corner)
- Placing the first move at the center will restrict victory from other positions

3. EXISTING MODELS

Several models and algorithms exist to develop Tic-Tac-Toe. The common one being is MinMax algorithm, which was initially developed for two-player zero-sum game theory.

MinMax algorithm is a decision theory for reducing the probable loss. It is widely used in decision theory, game theory, statistics and philosophy.

The main drawback of this algorithm, while developing a game like Tic-Tac-Toe is that the moves from the computer are quite predictable and monotonous thus making the gaming experience unrealistic.

To maintain spontaneity of the game we decided not to adapt existing algorithm.

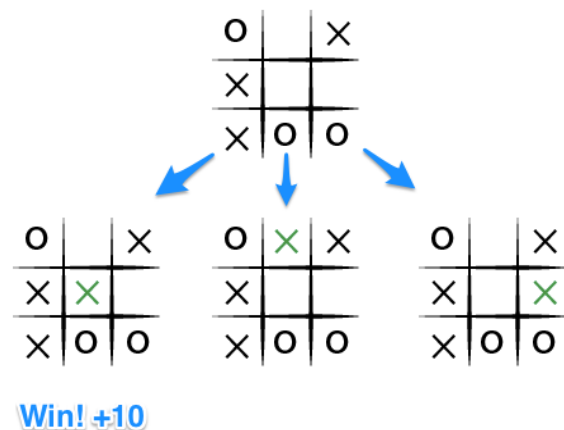


Fig 1: Possible Moves

The game tree starts with the initial position and consecutive nodes contain the possible unique ways the game can be played from each position. Overall a complete game tree for

3x3 Tic-Tac-Toe has 255168 leaf nodes. The partial game tree for Tic-Tac-Toe is shown in fig. 2.

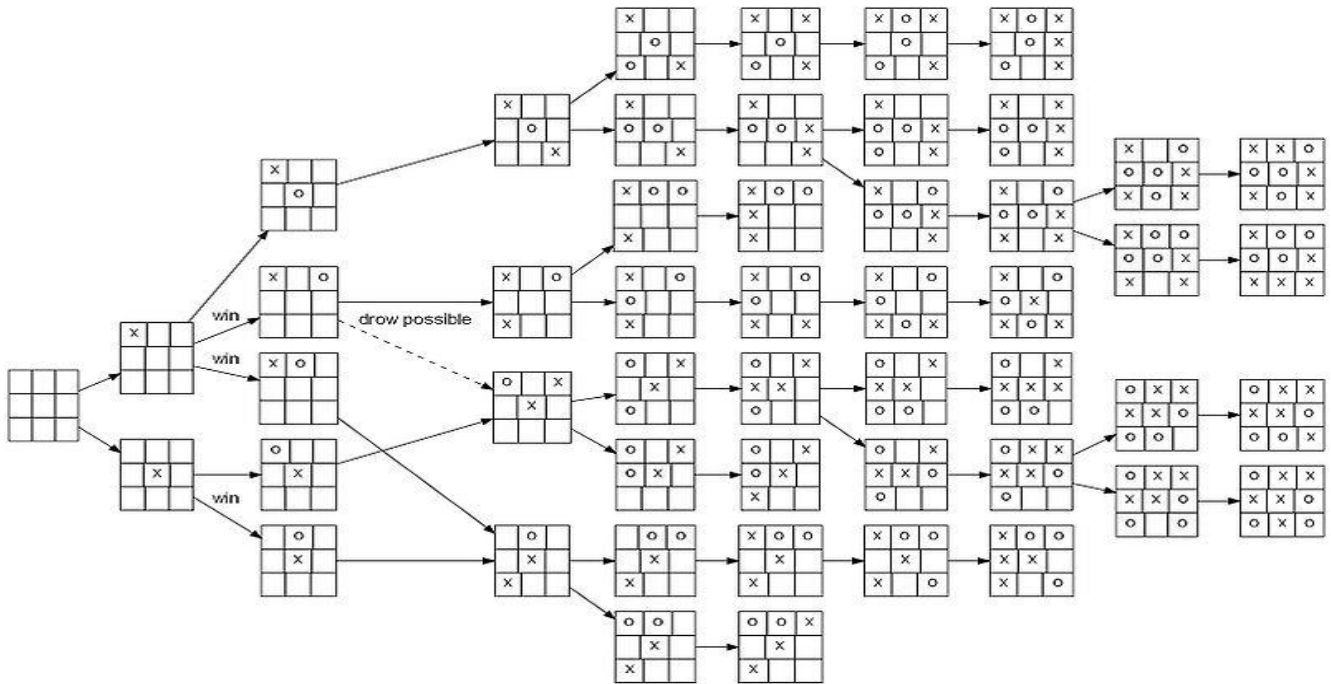


Fig 2: Game Tree for Tic-Tac-Toe

4. UPGRADED MODEL

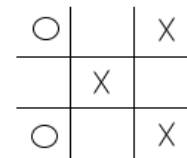
We have developed an algorithm for our Smart Tic-Tac-Toe that makes the game experience more realistic and give a feel of playing against another human. The new algorithm is designed using a combination of recursion and backtracking algorithm techniques. A method CheckResult() was created to concurrently check for the result of the game, after each move. The first move being the crucial one, has to be meticulously made as discussed in section 2. Our algorithm follows the first move strategy.

This new model depicts an environment that makes use of Artificial Intelligence which can be used by the differently abled people to interact with the computers. The model makes use of *Touch interface*, *Speech Recognition System* and *Text To Speech (TTS)* technologies which empowers a system to communicate and interact with the user.

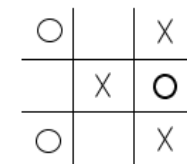
Our algorithm for Tic-Tac-Toe has been architected using two different modes. How the computer reacts to the opponent's move, depends on the mode selected:

- 1) *Defensive Mode*: This is the easy play mode for beginners. Playing in this mode will allow the computer to only defend the opponent's moves. Its goal will not be to win but to prevent the opponent from winning. The algorithm would make its move so that the opponent cannot win even if it had a chance of winning the game.

Example: if there exists such a scenario as in fig 3(a), the computer in DEFENDING mode would respond by playing 'O' at position 6 instead of position 4 to defend the opponent from winning fig 3(b).



(a)

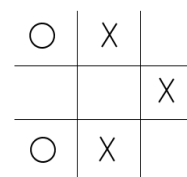


(b)

Fig 3: Defensive Mode Algorithm

- 2) *Attacking Mode*: This is the harder mode. The main intention of the computer shifts to the other side of just defending. Its motive will be to win the game. The algorithm would never miss an opportunity to win a game.

Example: if there exists such a scenario as in figure fig 4a), the computer in ATTACKING mode would respond by playing 'O' at position 4 to win the game fig. 4(b).



(a)

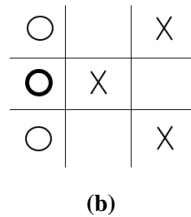


Fig 4: Attacking Mode Algorithm

4.1 Integrating Speech Recognition System

Vocal Speech is the natural means of humans to communicate. Computation will go to the next level if one will be able to communicate with the computers completely with his or her voice.

Fig. 5 depicts the general working principle of the Speech Recognition System. When a user speaks a word, it needs to be converted it into a digital form so that it can be recognized by the computer. The speech recognition engine based on the set of grammar identifies the user's words which can later be used for computation and further processing.

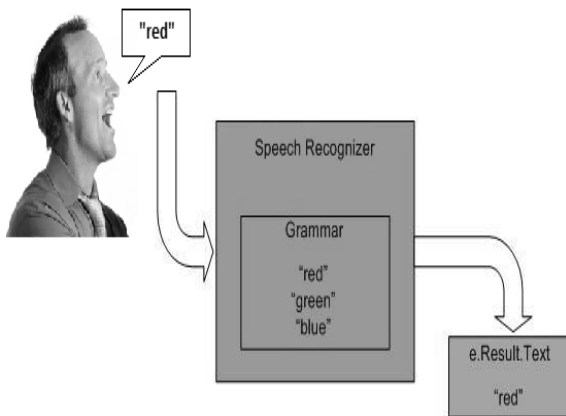


Fig 5: Speech Recognition Working

There are various API's available in the market. One of such is Microsoft's Speech Engine would help me to achieve the same.

First associate Microsoft's Speech Engine with your project. Doing so would enable us to use the Bing.Speech namespace.

Now, instantiate the SpeechRecognizer class as follows:

```
SpeechRecognizer SR = new SpeechRecognizer("en-US",
credentials);
```

The SR is the object of the SpeechRecognizer class and the credentials are the access tokens that I was provided by Microsoft before associating the Bing.Speech package with my project. The SpeechRecognizer property binds the SpeechControl control to an object of the SpeechRecognizer class. This step is required for the SpeechControl control to function.

```
SpeechControl.SpeechRecognizer = SR;
```

The various properties and methods associated with the SpeechRecognizer class are mentioned below:

Dispose() : Disposes the SpeechRecognizer object.

RequestCancelOperation() : intrudes speech recognition. The control now returns to the caller

RecognizeSpeechToTextAsync() : to initiate speech recognition session, which captures and interprets user speech, and then returns the results as a *SpeechRecognitionResult* object.

With the help of an asynchronous function which keeps running in passive mode, *SR.RecognizeSpeechToTextAsync()* method converts the Speech from the user to Text which is then used to respond to the users actions.

4.2 The voice of Computer

The voice of the computer has been achieved by using "Text to speech" technology by using *Windows.Media.SpeechSynthesis*;

After instantiating the *SpeechSynthesizer* class, activate the voice for the computer using:

```
SpeechSynthesizer synthesizer = new SpeechSynthesizer();
```

The *SpeechSynthesisStream* class is instantiated to enable reading and writing audio data generated by the speech synthesis engine (voice) to/from a random access stream:

```
SpeechSynthesisStream synthesisStream = await
synthesizer.SynthesizeTextToStreamAsync(text_to_speak);
```

The method *SynthesizeTextToStreamAsync* asynchronously generates voice output from a string which is passed as a parameter the method *SynthesizeTextToStreamAsync()*. Here the *text_to_speak* is the string that has to be spoken out by the computer.

5. POSSIBLE FURTHER UPGADATION

1. Increase the grid size:

Moving from 3x3 grid size to say 5x5 grid would not only make the gaming entirely different but will also increase the difficulty level of the game and the complexity of developing it would make it more enjoyable. This would also require the beginning and consequent moves to be strategized out wisely.

2. AI learning:

If successfully introduced, defeating the computer would be next to impossible. Victory and future will always belong to the computer.

3. Play over internet:

Taking the gaming online, would allow people to find players and compete against each other. There should also be a dedicated scoring system to store the details of the players.

4. Pattern Recognition:

Introducing Pattern Recognition System would enhance the chances of victory of computer against the user.

6. CONCLUSION

The Smart Tic-Tac-Toe developed by me may appear similar to the familiar game. But the following unique features have been introduced:

- A completely new algorithm, which makes the moves unpredictable and gives a sigh of playing against human.
- Speech recognition and Text to speech technology have been exploited and integrated in this model, to enable the visually impaired people to interact with computers.

The game which was accorded a gold rating from the Microsoft's Windows AppLab team and can be downloaded for free from the Windows App Store under the name "Smart Tic-Tac-Toe".

An enormous scope still exists, how Artificial Intelligence can enhance and improve the Human Machine interaction.

7. ACKNOWLEDGMENTS

The logic, programs and experiments were developed at Department of Computer Science and Engineering, SRM University. The author acknowledges the guidance from co-authors Dr. R. P. Mahapatra (Professor and HOD) and Mr. Partha Sarathi Chakraborty (Asst. Professor).

8. DOWNLOAD LINK

The application developed by us is suitable for laptops and tablets running windows 8.1 or above. It can be downloaded for free from the Windows App Store. The link to which is:

<http://apps.microsoft.com/windows/en-us/app/68556ec4-de47-487a-a55f-e3571fde2175>

or

<http://goo.gl/d3JGq6>

9. REFERENCES

- [1] Lalitha Saraoja Thota, Manal Elsayeed, Suresh Babu, "Implementation of Tic-Tac Toe Game in LabView", International Journal of Computer Trends and Technology Volume 12, June 2 2014
- [2] Preeti Saini, Pameet Kaur, "Automatic Speech Recognition: A Review", International Journal of Engineering Trends and Technology, Volume 4, Issue 4, 2013.
- [3] Santosh k. Gaikwad, Bharti W. Gawali, Pravin Yannavar, "A review on Speech recognition Techniques", International Journal of Computer Applications Volume 10, No-3, Nov, 2010
- [4] Van Cranenburgh, Andreas, Ricus Samid, Maarten van Someran, "Tic-Tac-Toe", 2007
- [5] Allen, Jonathan, M. Sharon Hunnicutt and Denis Kaltt 1987, From text to speech: the MITalk system, Cambridge University Press New York, NY, USA 1987
- [6] MicrosoftTechSupportForum, <https://social.microsoft.com/forums>
- [7] MicrosoftDevelopmentNetwork, <http://msdn.microsoft.com/>
- [8] Herbert Schildt, "C# 3.0 – The complete reference to C#", McGraw Hill Professional