Abstract - It is common for computer games to be used as tools to help introduce basic computer science concepts to the students. Digital games could have a much bigger role in learning than just as a motivational tool.

In this paper, we report the efforts of teaching Artificial Intelligence (AI) concepts to undergraduate students of computer engineering and information technology with games, because game motivate students, which increase retention and helps to become better computer engineers. Here we describe a case study of all sorts of games and/or puzzles inculcate in teaching AI concepts and other searching algorithms to the students. Teaching informed searching techniques like generate-and-test, hill climbing, A-star, AO-star, constraint satisfaction and means-end analysis is easy, if real life problems or puzzles are used. Using water-jug problem, 8-puzzel (sliding tiles), money-banana problem or block world puzzles helps in understanding these informed searching techniques. Students really put more efforts in inventing good heuristics functions for above games.

Minimax algorithm for two player game is complex algorithm which is used in IBM’s Deep-blue, who defeated world chess champion Gary Kasparov in 1997 is also implemented in lab. Our approach in writing and implementing those games or puzzle in understanding concepts better.

Keywords – Artificial intelligence, deep-blue, A-star, digital games

I. INTRODUCTION

Teaching artificial intelligence to undergraduate engineering and management students has become increasingly practical because of the recent explosion in the number and quality of AI resources aimed at applied scientists and business managers [1].

However, the usual drawback in these courses is the diversity and complexity of the different topics included in the course, usually including knowledge representation, problem solving, learning algorithms, fuzzy logic, inferential systems and intelligent agents. Games playing have an appeal towards students, especially CS and IT students, who generally spend a large part of their entertainment time playing games. Games can attract user attention in many different ways [2]. Some people enjoy playing games; some prefer designing and developing them rather than playing games. The number of students enrolled in the course has been continuously increasing during the last years few years.

II. AI AND GAME PLAYING

Game Playing is one of the oldest, most well-studied domains in AI. People like game and are good at playing them. Game playing is often viewed as an indicator of intelligence. In such types of games there is a clear, clean description of the environment. But the state spaces are very large and complicated, sometimes there is imperfect information, which makes the problem challenging [3]. In game playing it is clear when the AI is doing well.

1) Why study Game Playing?
   - It is easy to measure success
   - Easy to represent states of the games
   - There are small number of operators
   - Comparison against humans is possible.
   - Many games can be modeled very easily, although game playing turns out to be very hard.

2) Types of games

Most of the games learned in AI are categorized into perfect and imperfect games. Chess, checkers, go, othello are perfect information. We can see the entire, exact state of the board. Where as scrabble, bridge and most card games are imperfect information games, because cards are hidden, and therefore all information is not known to the opponent.

3) Major problem with Game design
   - Representing the “board” and their positions
   - Generating all legal next boards
   - Evaluating a position
   - Look ahead

III. GAMES FOR ALGORITHM STUDY

According to the undergraduate syllabus we explain many algorithms which require considerable amount of explanation using games. In initial heuristics searching techniques students are learning A*, AO*, means-end analysis, Best-first search algorithm, constraint satisfaction [3].
A. Constraint satisfaction Problems (CSP):

Normally applications of CSP are in scheduling, transportation, frequency allocation, car sequencing, design (car interiors), timetabling, real-time resource allocation etc. But we take cryptoarithmetic puzzles. In these puzzles each letter or symbol represents only one digit throughout the problem [5]. When letters are replaced by their digits, the resultant arithmetical operation must be correct. Other constrains are

i) the numerical base, unless specifically stated, is 10
ii) numbers must not begin with a zero
iii) there must be only one solution to the problem.

For example, [3] the puzzle SEND + MORE = MONEY, after solving, will appear like this:

\[
\begin{array}{cccc}
S & E & N & D \\
9 & 5 & 6 & 7 \\
+ & M & O & R & E \\
1 & 0 & 8 & 5 \\
\hline
\end{array}
\]

\[
\begin{array}{cccc}
M & O & N & E \\
1 & 0 & 6 & 5 & 2
\end{array}
\]

Students take more interest in solving such type of problems.

B. A* Algorithm

Best first search is a simplified A* and it is a combination of depth first and breadth first searches. While explaining this algorithm, we normally take 8-puzzle or sliding tiles problem. The 8-puzzle consists of a 3x3 board with eight numbered tiles and a blank space. A tile adjacent to the blank space can slide into that space, or you can view these actions as moving the blank space in the left, right, up, and down directions. You need to choose a suitable representation for all the states and actions, and compute the \( h(n) \) values.

Here we consider heuristic function for single state is given by number of incorrectly placed tiles. Initial and final states are given in the problem is shown in Fig 1.

\[
\begin{array}{cccc}
1 & 4 & 2 \\
8 & 3 \\
7 & 6 & 5
\end{array}
\]

Start State

\[
\begin{array}{cccc}
1 & 2 & 3 \\
8 & 4 \\
7 & 6 & 5
\end{array}
\]

Goal State

Fig 1 : Initial and Goal state

Students can choose any heuristics function to solve the problem and compare the solution. Using the given heuristics function, initial value of \( h(n) \) is for start and goal state is \( h(n) = 3 \) and \( h(n) = 0 \). Students are expected to solve the problem sing A* algorithm and show their path clearly to the solution. Part of the solution along with heuristics value is given. Partial game tree is shown in the Fig 2.

\[
\begin{array}{cccc}
1 & 4 & 2 \\
8 & 3 \\
7 & 6 & 5
\end{array}
\]

Fig 2: Partial search tree

C. Minimax Algorithm

John von Neumann outlined a search method called Minimax, as it tries to maximise your play, whilst minimising your opponents [3]. The minmax algorithm is applied in two player games, such as tic-tac-toe, checkers, chess, go. It is depth-first, depth-limited search procedure. In order to implement Minimax you need some way of measuring how good (or bad) your positions is. This is often called a utility function. If it is my turn to move, then the root is labeled a "MAX" node; otherwise it is labeled a "MIN" node indicating my opponent's turn. Expanding a complete search tree, until terminal states have been reached and their utilities can be computed. We go back up from the leaves towards the current state of the game. At each min node, back up the worst value among the children. At each max node, back up the best value among the children.

To explain this algorithm, the best puzzle students like to solve is Tic-Tac-Toe. A very rough approximation to the number of possible games for Tic-Tac-Toe is \( 9! = 362,880 \). The partial game tree is shown in Fig 3.

Fig 3: Partial game tree of Tic-Tac-Toe
D. Other games and puzzles

Apart from the 3 example games mentioned many other games and puzzles are introduced in the class so that students make more efforts in learning the concepts of Artificial Intelligence [4]. Following games are mostly liked by students:

i) Missionaries and cannibals (There are 3 missionaries, 3 cannibals, and 1 boat that can carry up to two people on one side of a river.)

ii) Eight queen problem (Place eight queens on a chessboard such that no queen attacks any other)

iii) 3 Black 3 White tiles (3 black and 3 white tiles are given in some initial configuration. Aim is to arrange the tiles such that all the white tiles are to the left of all the black tiles. The cost of a translation is 1 and cost of a jump is 2.)

iv) Water jug problem (Given a full 5-gallon jug and an empty 2-gallon jug, the goal is to fill the 2-gallon jug with exactly one gallon of water.)

v) Block world problem

vi) Remove 5 Sticks (Given the following configuration of sticks, remove exactly 5 sticks in such a way that the remaining configuration forms exactly 3 squares.)

IV. DISCUSSION

Student’s feedback regarding teaching Artificial intelligence with game playing is really interesting and easy to solve the AI problem. Students are more inclined towards writing the code for the above stated puzzles. Our efforts in this regard needs to be supported with many case studies and applications.

V. CONCLUSION

We reported on the efforts to teach computer science and artificial intelligence with games for engineering and management students. We and students are having fun in the process of learning new concepts. The future will show whether we manage to achieve our objectives, namely to increase retention and educate better computer and IT Engineers. Of course, there have been problems and difficulties related to the practical application of the games design in the practical classes, because of lengthy programming code.

REFERENCES

[1] George S. Young* and Sue Ellen Haupt, Teaching Artificial Intelligence To Meteorology Undergraduates, Fourth Conference on Artificial Intelligence Applications to Environmental Science, The Pennsylvania State University, P1.1


