Leopard and Goat

• Problem Solved

We have taken the traditional Indian game "Cow and Leopard" as our project. There were many problems we solved in the course of making it. Some of them are as follows:

- <u>Problem with drag</u>: -
  Firstly drag seemed to be quite simple (as we did that in our CS101 project). Because of that we had to make many classes and their objects.
- <u>Complexity of the structure of board: -</u>
  The board unlike the checker or any other simple one has very complex structure, which was even complicated by the moves that the player can make.
- <u>Unsymmetrical players:-</u>
  The two players, namely goat and leopard, are unsymmetrical which created problem in our min-max tree and also in the no. of ways that a player can move; hence we had to make different function for leopard and goat to calculate their next move.

Actual game

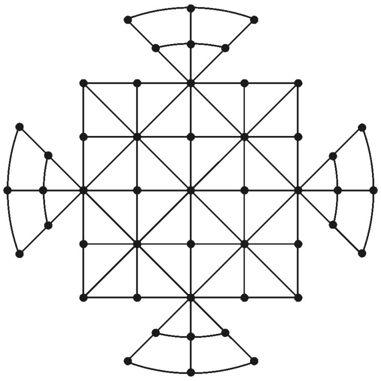
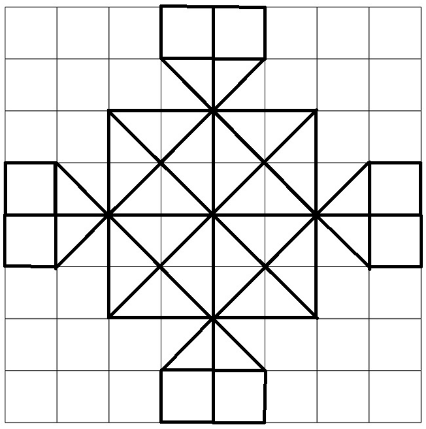
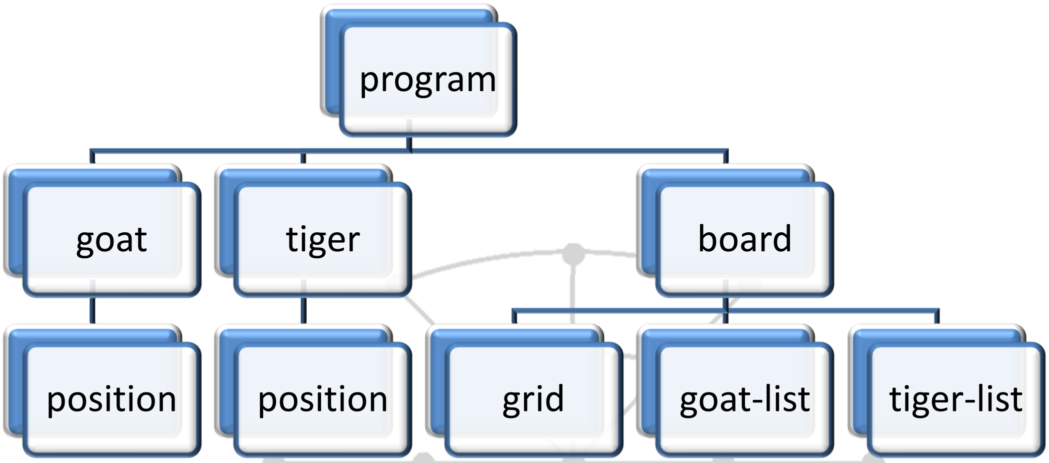


Fig. 1

What 2d vector look like in grid



- ## Idea of design



The above hierarchy chart shows the overview of the whole classes and their fields. In details:-

- Goat:-
  This class essentially represents goat, with all the functions related to it. Function like list-of-next-move and list-of-next-pos give list of next possible legal move and position resp.
- Tiger: -
  This class essentially represents leopard, with all the functions related to it. Function like list-of-next-move and list-of-next-pos give list of next possible legal move and position resp.
- Board:-
  It is the real hero of the game. It is the object "my-board", which is an object of class board which handles whole game, keep track of the positions of every single piece, may it be goat or leopard. It also handles a 9X9 vector, named as grid which in real sense is the actual grid of game. Every change taking place on the board is depicted on the grid.

Samples of input and output

The sample of input is simply playing the game…as we play game it unfolds its complexity.

Limitations and bugs

We tried our best not to come up with any bug in our program. Whatever we made should have been good. But being a program there are some of its limitations. Sometime it will play as stupid as a small kid, while sometime like a super-"stud". This all depends on the move it chooses in alpha beta pruning. Also some times the moves are non-intuitive, as it prefers very weird moves over simple one.

Point of interest

It could appear simple, but the feature of drag that we gave was very tiresome. It created a lot of problem in implementation of the program, but we adhered to this feature even when none of our friends did it. And also the complexity of board…our code for the side parts of the board is at least 10X greater than the main simple board part.

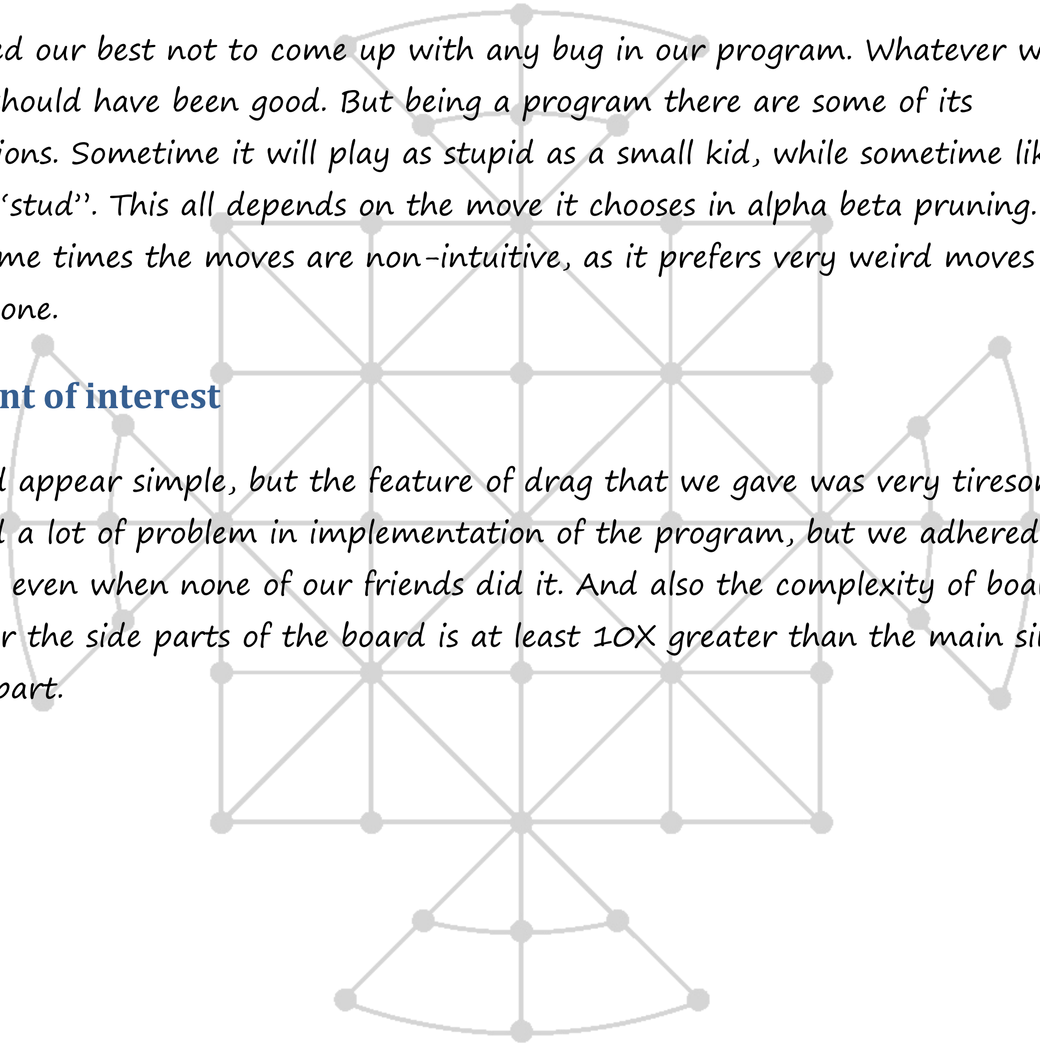


Fig. 1