

ONLINE BAGHCHAL EXCHANGE (OBX) NOTATION

Manual

By Bhupal Sapkota

<https://baghchal.net>

v1.0

Jan 2019

Table of Content

BAGHCHAL GAME

BAGHCHAL RULES

ONLINE BAGHCHAL (BAGHCHAL.NET)

ONLINE BAGHCHAL EXCHANGE (OBX) NOTATION

SAMPLE CODE

REFERENCES

1. BAGHCHAL GAME

BAGHCHAL is an ancient Nepali game. It is so popular in Nepal that we rarely find a single Nepali who has never played this game at least once in his life. We can see people playing this game on Baghchal Boards drawn on stone slabs at every CHAUTARI and DEURALI, places made to have a short or long rest.

BAGHCHAL is an indoor game of planning and strategic thinking. This game is played on a simple board with 20 goats and 4 tigers. There are all together 25 cross points (rooms) on the square board out of which the rooms at the four corner contain the four tigers at the beginning of the game. The games proceeds placing the goats in empty rooms. Tigers try to capture the goats and goats try to lock the tigers. If all the tigers are locked goats win the game. Similarly, If more than five goats are captured it is impossible to lock the tigers and hence, the tigers win the game.

Deployment of tactics, combination of strict logic and unlimited scope for the display of initiative and imagination have made the game very interesting and resembles more or less like the famous game Chess.

Baghchal (Meaning - Tiger Moving game or Tiger & Goats game) is a two player strategy board game originating in Nepal. Its finite (regarding the number of available strategies or choices), dynamic (players do not simultaneously decide their strategies), sequential (players alternate turns), non-cooperative (players compete and cannot win or lose together), zero sum, complete and perfect information game.

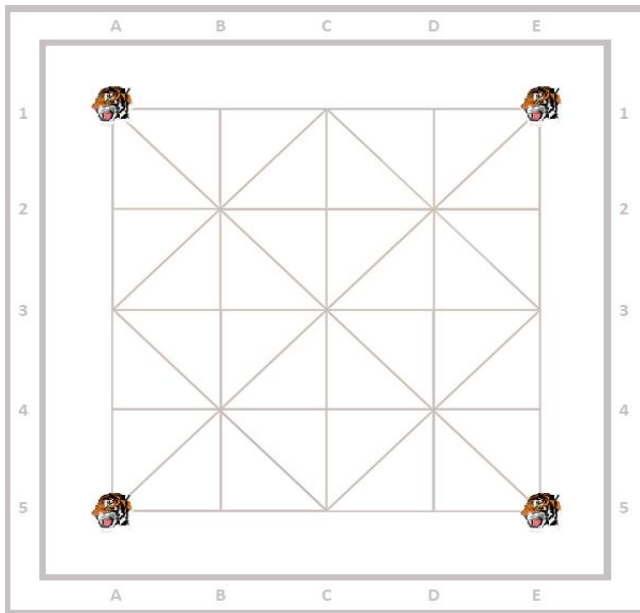


Figure - Initial Board position

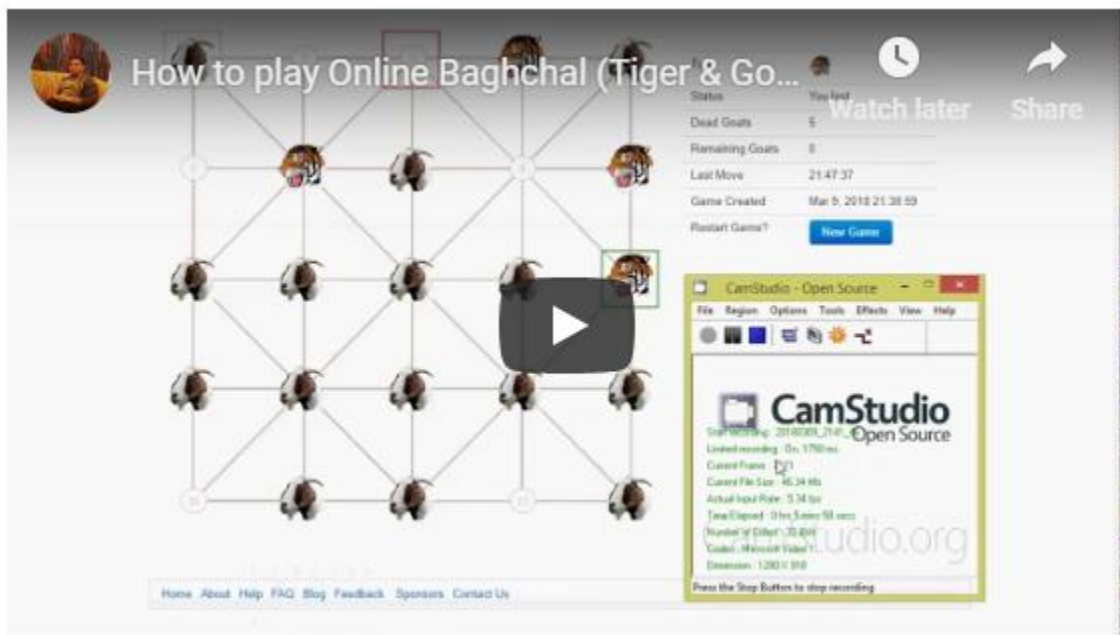
The game is asymmetric in that one player controls four tigers and the other player controls up to twenty goats. The tigers 'hunt' the goats while the goats attempt to block the tigers' legal moves.

A Little History

There is no any authentic documentation or other proofs about the date and place of the origin of BAGHCHAL. It is said BAGHCHAL is originated 2000 to 2500 years before by the farmers who lived in the hilly regions of our country with their herd of sheep, goats and other domesticated animals. As they brought their animals for grazing, the wild animals (specially the tiger) attacked their animals and kill them. The farmers had to manage a special security to prevent their animals from the attack of the wild animals. They trained their mastiff dogs to fight with the tigers. The dogs used to be armed with special chains (Janjir) round their neck to prevent them from the attack of tigers. Specially designed rooms were made somewhere in the jungle near the village to trap the tigers which are called "BAGH KHOR" meaning "The trap for tiger". We still find one or more deserted BAGHKHORs in almost each village of our country. Goats or

sheep were kept at the safe corners of these specially designed rooms to tempt the tigers. Tigers were lured and entered the special rooms where they were trapped. In this way, the farmers (shepherds) spent most of their time thinking about the security of their animals from the attack of the tigers. During this process of strategic thinking they formulated the game BAGHCHAL. Since then this game is played from generations to generations not only in NEPAL but some places of INDIA and TIBET as well.

2. BAGHCHAL RULES



Watch How to Play - <https://www.youtube.com/watch?v=fCrjk6j-GoU>

General Rules:

- The game board is a grid of intersecting orthogonal and diagonal lines upon which pieces are placed.
- There are 25 intersections in the board where game pieces can be placed
- The pieces are made up of 4 tigers and 20 goats.
- The pieces move along the lines to adjacent intersections. One intersection at a time.
- Tiger can move two intersections when jumping a goat over to a empty intersection.
- One intersection is left for 2nd phase of game play where goats have to move
- All 4 Tigers start at corner positions.
- Goats play in two phases.
- In 1st phase: All 20 goats must be placed turn by turn into the board.
- In 2nd phase: Goats move after all 20 have been placed on the board.
- Players alternate their moves.

Rules for Tigers:

- The goal of the tigers is to capture five goats by jumping over them.
- They can start capturing goats any moment after the match has started.
- They can capture only one goat at a time.
- A tiger cannot jump over another tiger.
- A tiger can only jump over one goat.
- Tiger can jump over a goat in any direction, but it must be to an adjacent intersection following the lines drawn on the board.
- A tiger cannot jump over a corner position goat piece.

Rules for Goats:

- The goal of the goats is to immobilize the tigers by surrounding them in such a way that no tiger can move.
- They must leave the board when captured.
- They cannot jump over the tigers or other goats.

Winning Rules:

- Tigers win when they capture 5 goats
- Goats win when they block all possible legal moves of the tigers.
- Tigers win when they block all possible legal moves of the goats (called Locked Goats)
- **Locked Goats:** When all goats are locked by tigers and goats cannot make any move in their turn then it is considered to be a grand win of tigers, equivalent to winning twelve games..
- **Dandee:** Dandee means a straight and strong stick. When all four tigers are locked diagonally in a straight line then it is called Dandee and considered as a grand win of goats, equivalent to winning twelve games. It possibly means to lock the all four tigers in desired rooms (in a straight line) by sticking them.

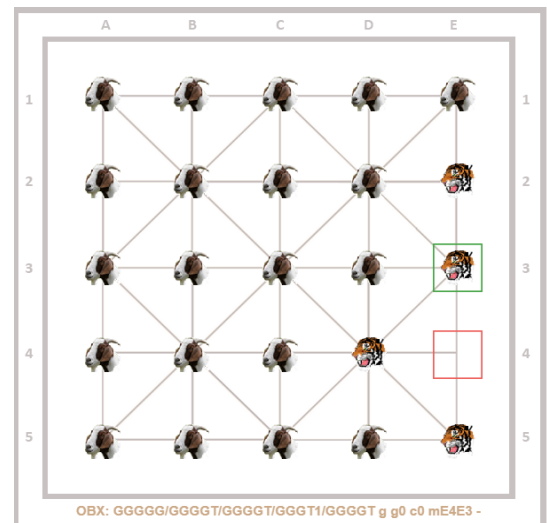


Figure- Locked Goats

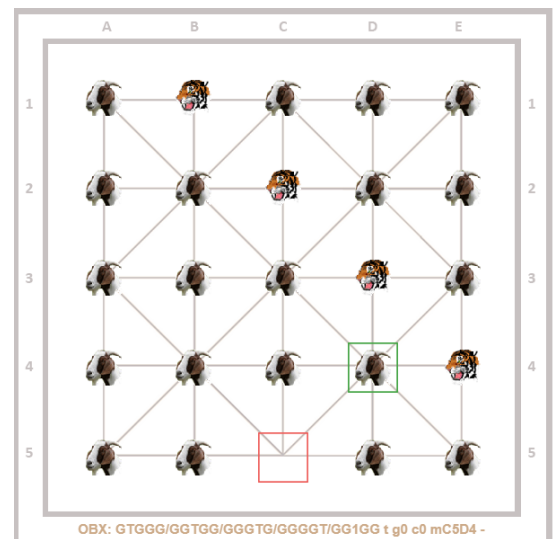


Figure - Dandee position (tigers locked diagonally at B1, C2, D3, E4)

3. ONLINE BAGHCHAL (BAGHCHAL.NET)

Online Baghchal was an online real-time version of Baghchal board game. Playing Baghchal has been my hobby since childhood so I developed Online Baghchal website and mobile apps. However, due to lack of time and resources I discontinued the site. I still locally run the platform code time to time out of nostalgia.

Similar to Lichess.org I was aiming to make Online Baghchal a go to place for Baghchal players and travelers in Nepal to play and socialize online. Baghchal is in a way connected to tourists and travelers in Nepal since every trek or hike they go on they will find people playing Baghchal on village tree shades and Baghchal is one of the most popular souvenir from Nepal.

So instead I moved focus from Game play aspect of baghchal.net towards Online Baghchal Exchange (OBX) notation. Currently baghchal.net hosts OBX Viewer and Bot testing software.

4. ONLINE BAGHCHAL EXCHANGE (OBX) NOTATION

While working on Online Baghchal, I thought it'd be cool to have third party bots into game play platform. If we search in Github even today there are 50+ good quality Baghchal apps out there built by individual and scholars. But each of them worked independently of others and there was no easy way we could reuse the code to evaluate our board positions. So out of necessity, similar to FEN in Chess, Online Baghchal Exchange (OBX) is a notation was developed. The purpose of OBX is to provide all the necessary information to evaluate or to restart a game from a particular position. OBX can also be used to record offline games to be studied and analyzed later.

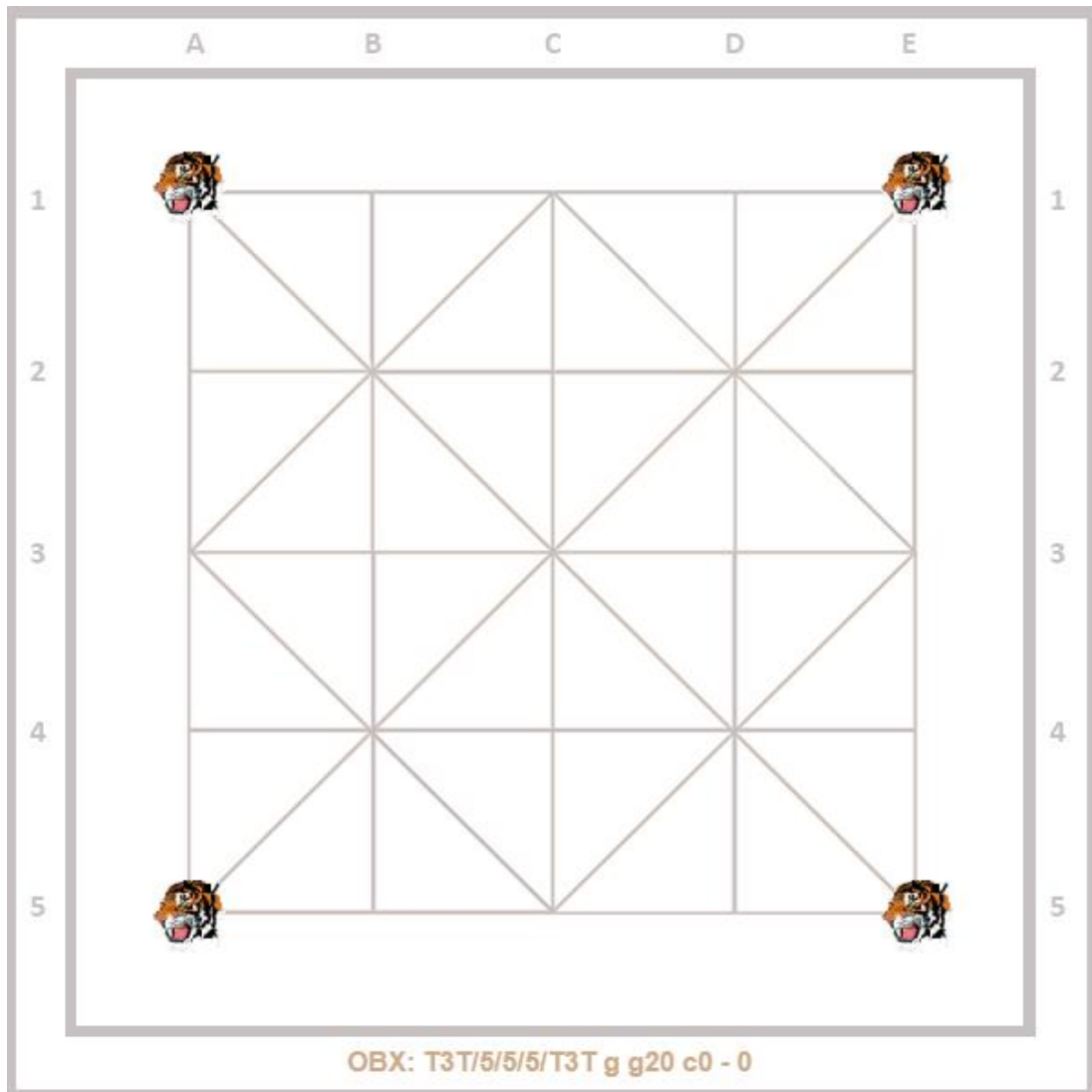


Figure – Initial board position OBX: T3T/5/5/5/T3T g g20 c0 - 0

Board coordinates and notation in Baghchal is a simple algebraic notation. The columns from left to right are Labeled as A, B, C, D, E and the rows from up to down are Labeled as 1, 2, 3, 4, 5 for simplicity of use while hand coding board positions or while representing in computer data structures. The rooms are noted first with Column label and then with Row label. For example the room at the top-left corner is noted as A1 and the room at the bottom-right corner is noted as E5. Similarly the rooms at the top rows are noted as A1, B1, C1, D1 and E1.

OBX Definition:

An OBX record defines a particular game position, all in one text line and using only the ASCII character set. Characters are case-insensitive. An OBX record contains six fields, the separator between fields is a space. The fields are:

1. Piece placement (Starting from A1 to E5). Each row is described, starting with row 1 and ending with row 5, the contents of each rooms are described for columns A through E representing T for tiger, G for goat, digits 1-5 for the numbers of empty rooms, and "/" separates each row.
2. Turn. "g" means Goat moves next. "t" means Tiger.
3. Remaining goats. Represents Goats to be placed in the board. In the format "gX", where X is the number of remaining goats.
4. Captured goats. Represents Goats captured during the gameplay. The number of remaining goats will reflect the change accordingly. In the format "cY", where Y is the number of captuted goats.
5. Last move (optional). Representing the last move which resulted in the current board position. In the format "mPN", where P is the previous room and N is the next room. For goat's placement phase use format "mN", where N is the placed room. For no value use "-".
6. Move number (optional). Initial position being 0 each successive turns will get numbered in increasing order. For no value use "-".

NOTES: Pieces (separated by / for each row) turn (t or g) goatsCount(eg. g20 initial position, g0 all goats place in board) capturedGoatCount (eg. c5) lastMove(eg. mC2E2, Top left corner: A1, Bottom right corner: E5) moveNumber (eg. 20, initial board position being 0)

i.e <pieces> <turn> <goatsCount> <capturedGoatCount> <lastMove> <moveNumber>

OBX Example:

OBX for Starting position:

T3T/5/5/5/T3T g g20 c0 - 0

Here is the OBX after Goat's first move of placement phase at C1.

T1G1T/5/5/5/T3T t g19 c0 mC1 1

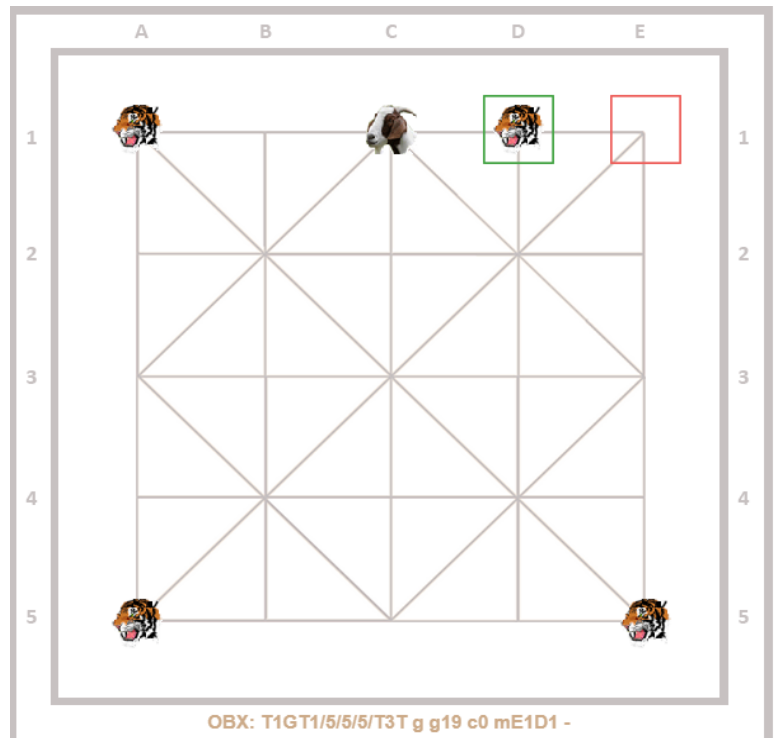
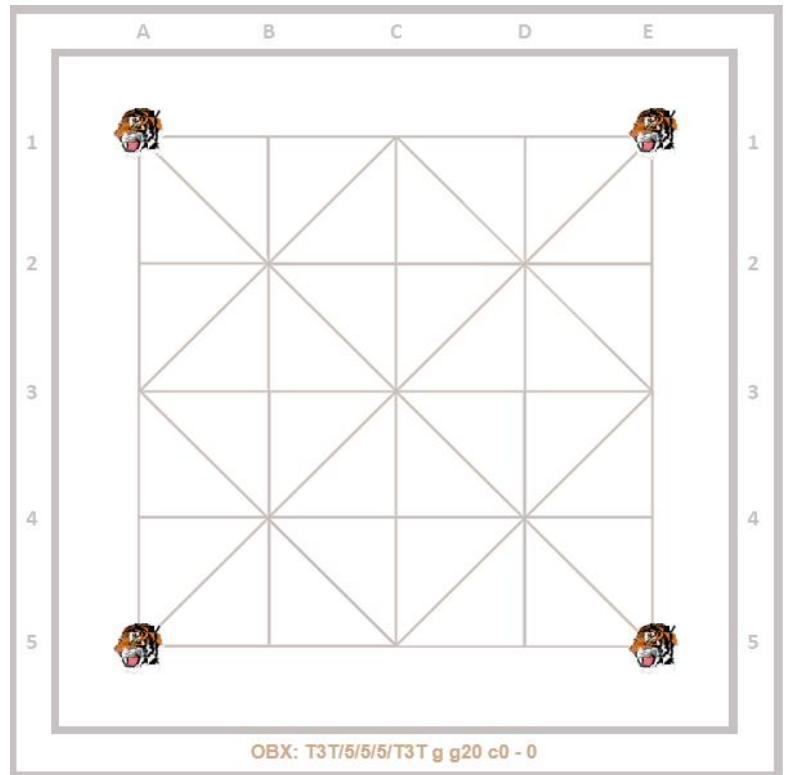
And then, tiger moves to D1 from E1 to threaten the Goat at C1

T1GT1/5/5/5/T3T g g19 c0 mE1D1 2

And then, Goat places next goat at B2 to save previous got from the capture.

TGGT1/5/5/5/T3T t g18 c0 mB2 3

And the game continues...



5. SAMPLE CODE

Initial Game State Array

```
$initialGameState = array (  
  'board' => array (2,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,2,0,0,0,2), // 0 to 24  
  'turn' => 'g', // t or g  
  'remainingGoats' => 0, //max 20  
  'capturedGoats' => 0, //max 5  
  'lastMove' => "-", //fromCoordinate|toCoordinate  
  'moveNumber' => 0 //initial position is 0th move  
);
```

Convert room indices 0-24 to coordinates (A1=0, E5=24) etc

```
function getMoveNotation($boardIndex)  
{  
  $xCoords = ["A", "B", "C", "D", "E"];  
  
  $x = $xCoords[$boardIndex % 5];  
  $y = intval($boardIndex/5) + 1;  
  
  return $x . $y;  
}
```

Get room index from move coordinates:-

```
function getBoardIndexFromMoveNotation($coordinate)  
{  
  
  $xCoordsLetters = ["A", "B", "C", "D", "E"];  
  
  $xCoord = $coordinate[0];  
  $xIndex = array_search($xCoord, $xCoordsLetters);  
  
  $yCoord = $coordinate[1];  
  $yIndex = ($yCoord % 10) - 1;  
  
  return $xIndex ? ($xIndex + $yIndex * 5) : false;  
}
```

Get OBX from Game State

Assumes array values 2 for tigers and 0 for empty space.
To represent moves 5 is added/subtracted to/from the values.

```
function getObxFromGameState($gameState){

    $obx = "";
    $last_move = $gameState['lastMove'];
    $lastMoveFrom= "";
    $lastMoveTo= "";

    $turn = $gameState['turn']; //default turn to goat if not provided.
    $goatCount= "";
    $moveNumber = $gameState['moveNumber'];
    $capturedGoats = $gameState['capturedGoats'] ? $gameState['capturedGoats'] : 0;
    $emptyCount = 0;

    for($i=0;$i<25;$i++)
    {
        $temp = $gameState['board'][$i];

        if ($temp == 5) //moved from here
        {
            $lastMoveFrom = getMoveNotation($i);
        }

        if ($temp > 5) //moved to here
        {
            $lastMoveTo = getMoveNotation($i);

            if ($temp == 6) //goat moved
            {
                $turn = "t";
            }
            else if ($temp == 7) //tiger moved
            {
                $turn = "g";
            }
        }
    }

    if($temp >= 5) $temp -= 5;

    if($temp == 1)
    {
        if ($emptyCount !=0)
        {
            $obx .= $emptyCount;
        }
    }
}
```

```

        $emptyCount =0;
    }

    $goatCount++;
    $obx .="G";
}

else if($temp == 2)
{
    if ($emptyCount !=0)
    {
        $obx .= $emptyCount;
        $emptyCount =0;
    }

    $obx .= "T";
}
else if ($temp==0) {
    $emptyCount++;
}

if ($i%5 == 4)
{
    if ($emptyCount !=0)
    {
        $obx .= $emptyCount;
        $emptyCount =0;
    }

    if ($i != 24)

    {
        $obx .="/";
    }

}

}

//turn
$obx .= " " . $turn;

//goatsCount
$obx .= " g" . (20-$goatCount-$capturedGoats);

//captureCount
$obx .= " c" . $capturedGoats;

$last_move = $lastMoveFrom . $lastMoveTo;

```

```

//lastmove
if ($last_move != "")
{
    $obx .= " m" . $last_move;
}
else
{
    $obx .= " -";
}

if($goatCount == 0 && $capturedGoats == 0) $obx .= " 0";
else $obx .= " " . $moveNumber;

return $obx;
}

```

Get Game State from OBX:

```

function getGameStateFromObx($obx)
{
    $gameState = array (
        'board' => array (0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0), // 0 to 24
        'turn' => 'g', // t or g
        'remainingGoats' => 0, //max 20
        'capturedGoats' => 0, //max 5
        'lastMove' => "-", //fromCoordinate|toCoordinate
        'moveNumber' => 0 //initial position is 0th move
    );

    $obx = trim($obx);
    $obx .= " "; //extra space to make parsing simpler.
    $obxSegments = explode(" ", $obx);

    /*
    Array
    (
        [0] => GTGG1/TGGGG/GGGGG/T1T1G/GGGGG
        [1] => g
        [2] => g1
        [3] => c1
        [4] => mD4C4
        [5] => -
        [6] =>
    )*/

    //print_r($obxSegments);
    $obxBoardValues = $obxSegments[0];
    $obxTurn = $obxSegments[1];

```

```

$obxRemainingGoats = $obxSegments[2];
$obxCapturedGoats = $obxSegments[3];
$obxLastMove = $obxSegments[4];
$obxMoveNumber = $obxSegments[5];

$obxBoardValuesRows = explode("/", $obxBoardValues);

//print_r($obxBoardValuesRows);

$boardIndex = 0;
$emptyCount = array(1,2,3,4,5);

foreach ($obxBoardValuesRows as $value)
{

    $rowValuesLength = strlen($value);
    $i = 0;

    while ($i < $rowValuesLength && $boardIndex < 25)
    {

        $obxCharRaw = $value[$i];
        if (in_array(intval($obxCharRaw), $emptyCount))
        {
            for($j=0; $j<intval($obxCharRaw); $j++)
            {

                $gameState['board'][$boardIndex] = 0;
                $boardIndex++;

            }

        }

        if($obxCharRaw == "t" || $obxCharRaw=="T")
        {

            $gameState['board'][$boardIndex] = 2;
            $boardIndex++;

        }

        if($obxCharRaw=="g" || $obxCharRaw=="G")
        {

            $gameState['board'][$boardIndex] = 1;
            $boardIndex++;

        }

        $i++;

    }

}

```

```

if($objTurn=="g" || $objTurn=="G") {
    $gameState['turn'] = 'g';
}

if($objTurn=="t" || $objTurn=="T") {
    $gameState['turn'] = 't';
}

$gameState['remainingGoats'] = intval(substr($objRemainingGoats, 1));
$gameState['capturedGoats'] = intval(substr($objCapturedGoats, 1));
$moveCoordinates = strtoupper(substr($objLastMove, strlen($objLastMove) > 1 ? 1 : 0));

if($moveCoordinates!='-') {

    $gameState['board'] = makeMove($gameState['board'], $moveCoordinates);
    $gameState['lastMove'] = $moveCoordinates;

}

if ($objMoveNumber != '-') {
    $gameState['moveNumber'] = intval ($objMoveNumber);
}

return $gameState;
}

```

OBX Moves Matrix:

This matrix tabulates the number of valid moves for goat or tiger pieces. Can be used to do positional evaluation of the board while finding best moves.

```

$objMovesMatrix = [
    [ 3.0, 3.0, 5.0, 3.0, 3.0],
    [ 3.0, 8.0, 4.0, 8.0, 3.0],
    [ 5.0, 4.0, 8.0, 4.0, 5.0],
    [ 3.0, 8.0, 4.0, 8.0, 3.0],
    [ 3.0, 3.0, 5.0, 3.0, 3.0],
];

```

OBX Tiger Capture Moves Matrix:

```

$objTigerCaptureMovesMatrix = [
    [ 3.0, 2.0, 5.0, 2.0, 3.0],
    [ 2.0, 3.0, 3.0, 3.0, 2.0],
    [ 5.0, 3.0, 8.0, 3.0, 5.0],
    [ 2.0, 3.0, 3.0, 3.0, 2.0],
    [ 3.0, 2.0, 5.0, 2.0, 3.0],
];

```


OBX Tiger Moves Matrix:

Sum of above two matrices. Used for evaluating tiger moves.

```
$obxTigerMovesMatrix = [  
  [ 6.0, 5.0, 5.0, 5.0, 6.0],  
  [ 5.0, 11.0, 7.0, 11.0, 5.0],  
  [ 10.0, 7.0, 16.0, 7.0, 10.0],  
  [ 5.0, 11.0, 7.0, 11.0, 5.0],  
  [ 6.0, 5.0, 10.0, 5.0, 6.0],  
];
```

6. REFERENCES

1. "System Analysis for BAGHCHAL", Raghunath Dhungel
2. "Play in Scientific and Mathematical Non-Formal Education. Bagh Chal, a Tigers-and-Goats Game", Carlos Sergio Gutiérrez-Pereraa, Alicia Fernández-Oliverasb, María Luisa Oliveras
3. Baghchal Game Wikipedia
4. FEN Notation Wikipedia
5. BoardGameGeek Baghchal
6. Baghchal.net