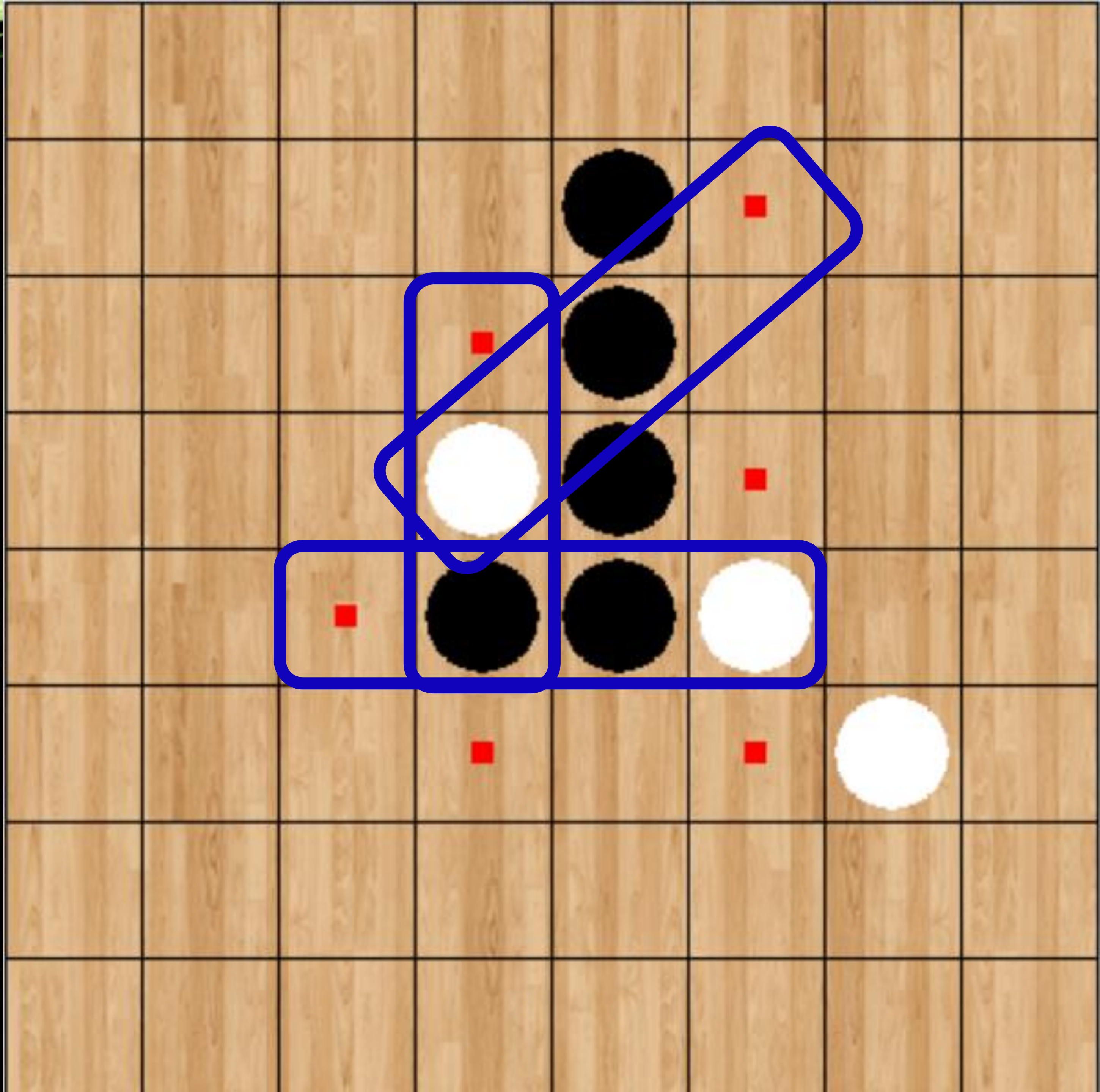
REVERSI

USING PYTHON PROGRAMMING

Group 3	
林益帆	B06401038
江雨柔	B06303077
鄺思靜	T07703119

HOW TO PLAY

- Number of players required: 2
- Board: 64 squares (8 rows x 8 columns)
- Rules:
  1. Players ***take turn to make one move*** by placing a piece on the board.
  2. Placed pieces ***can never be moved*** to another square later in the game.
  3. The incorporated piece must ***outflank*** one or more of the opponent placed pieces (one straight row in vertical, horizontal or diagonal).



- Scoring:

  - Each piece is counted as one point.

  - The ***game is over*** when all the squares of the board are taken or none of the players can move.

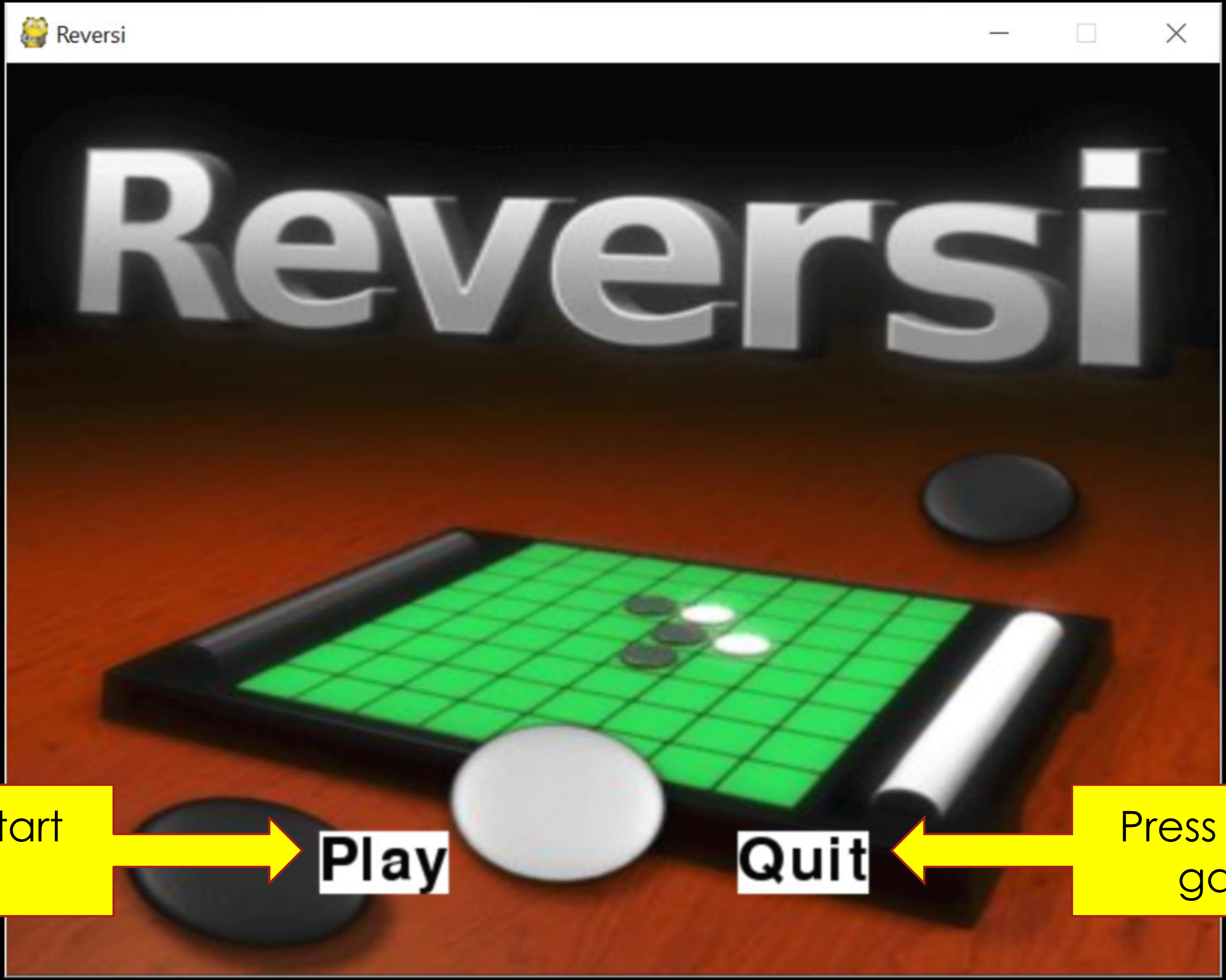
    Winner → player who has more pieces/higher score on the board

    Game tie → both players have the same number of pieces/same score on the board.

WHAT HAVE WE DONE?

- Adding features such as:
  - Choosing “play” or “quit” game
  - Choosing player mode
  - Designing “Two players” mode
  - Choosing difficulty
  - Designing “hard” mode by using minimax with alpha beta pruning
  - Choosing tile color
  - Display of tile color, scoring, current turn, new games and hint on the interface of playing game
  - Display of results and option to quit or start a new game on the interface of ending the game.

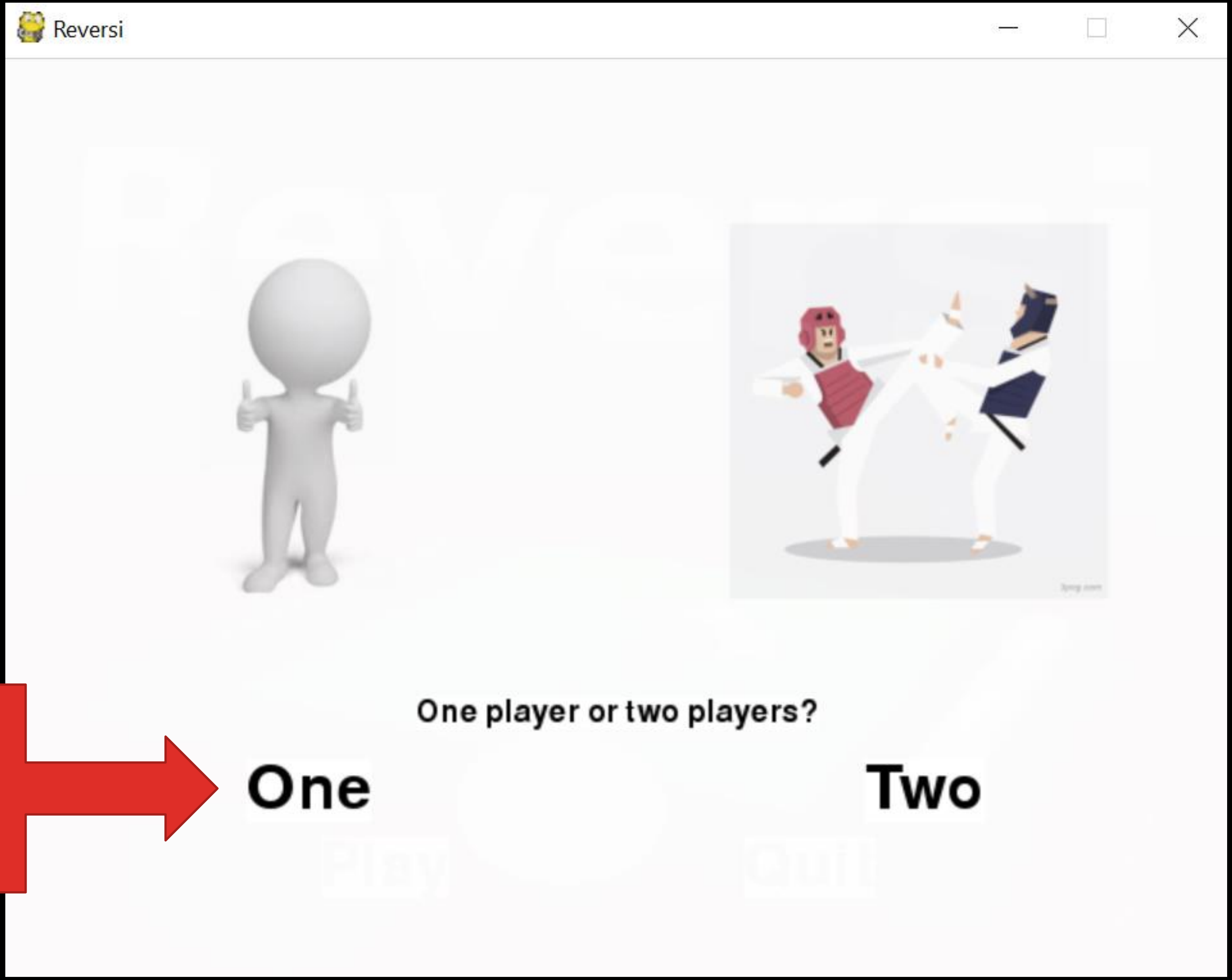
1. STARTING THE GAME



Press to start game

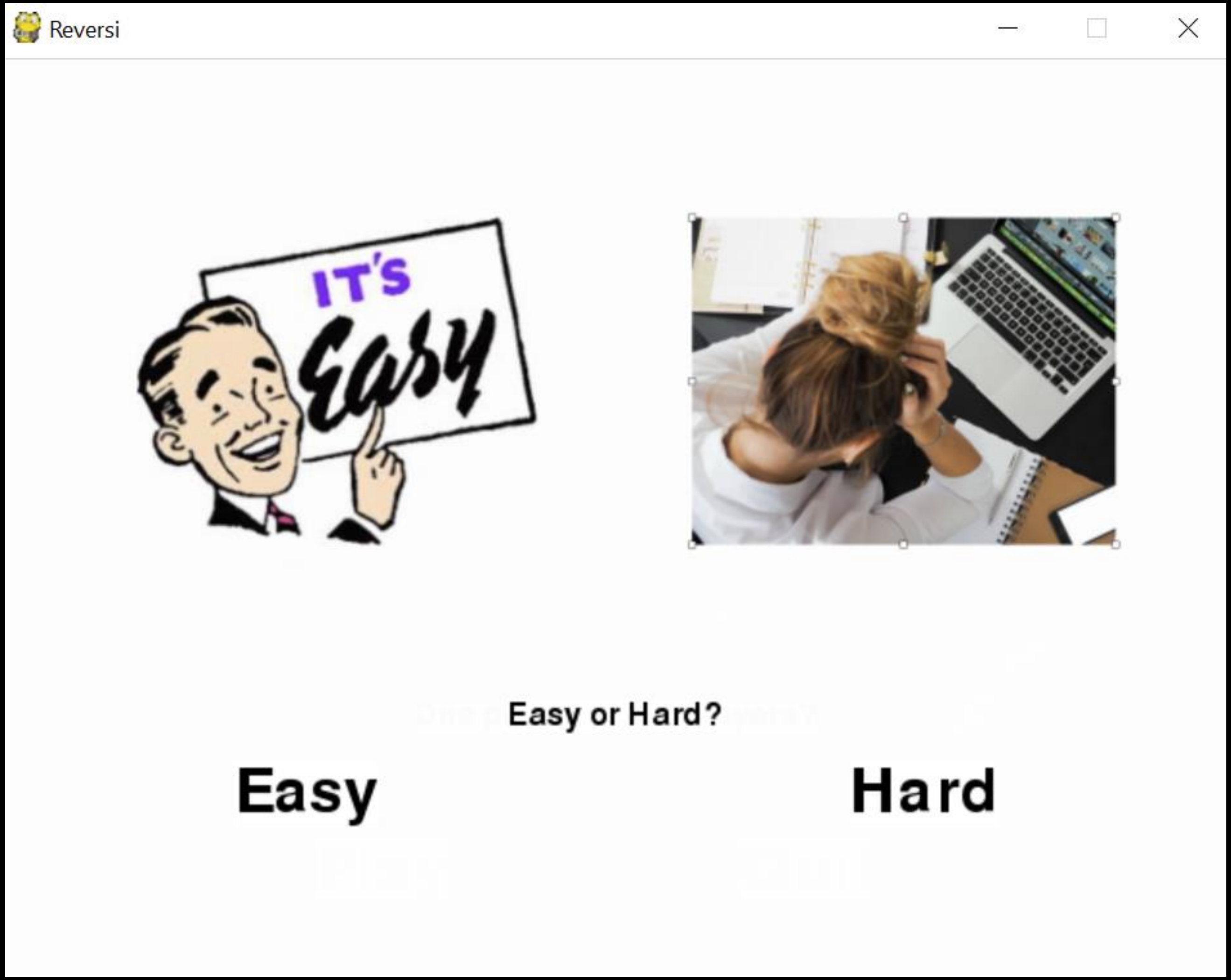
Press to quit game

2. CHOOSING PLAYER MODE



Play with computer

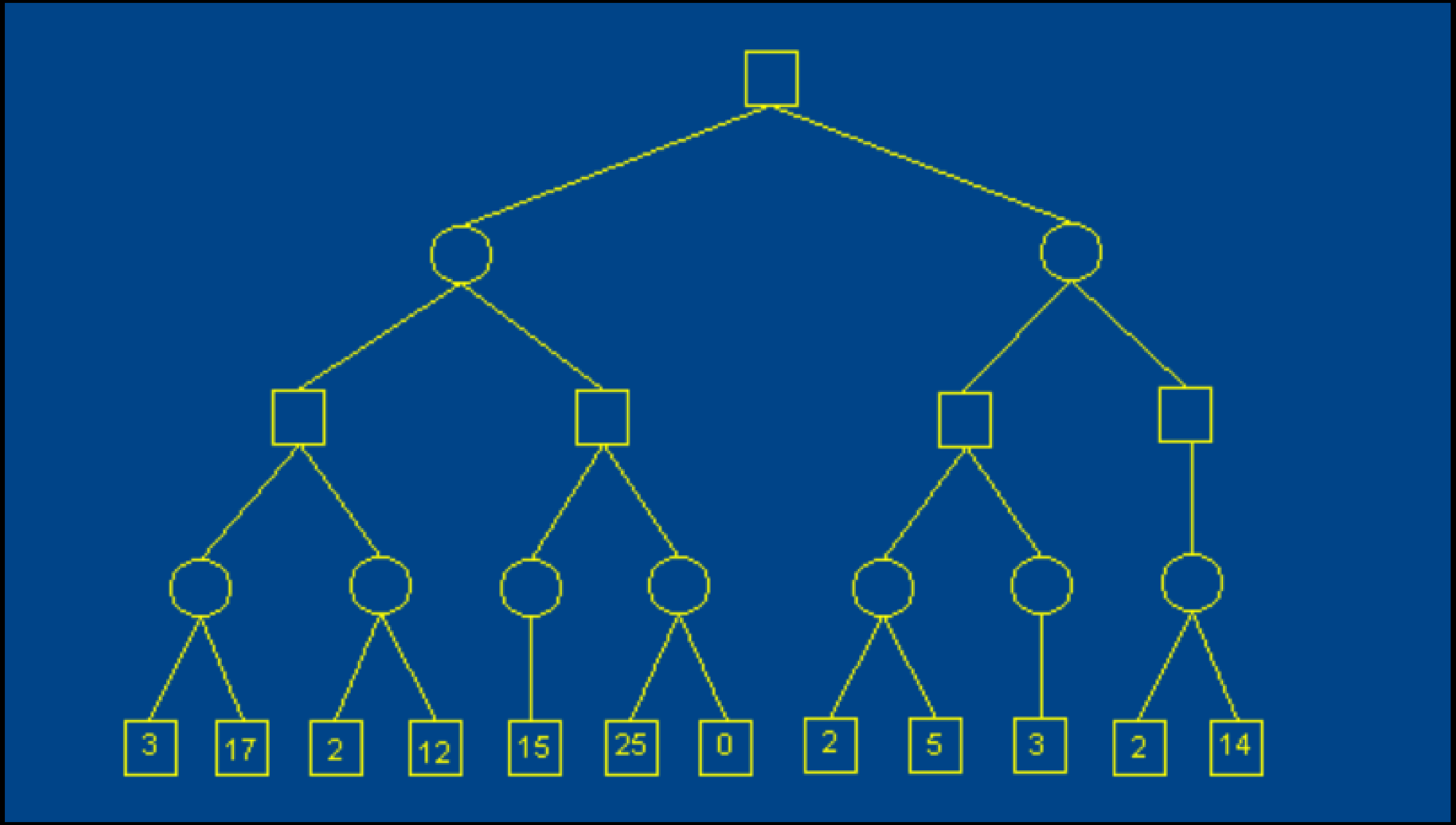
3. CHOOSING DIFFICULTY



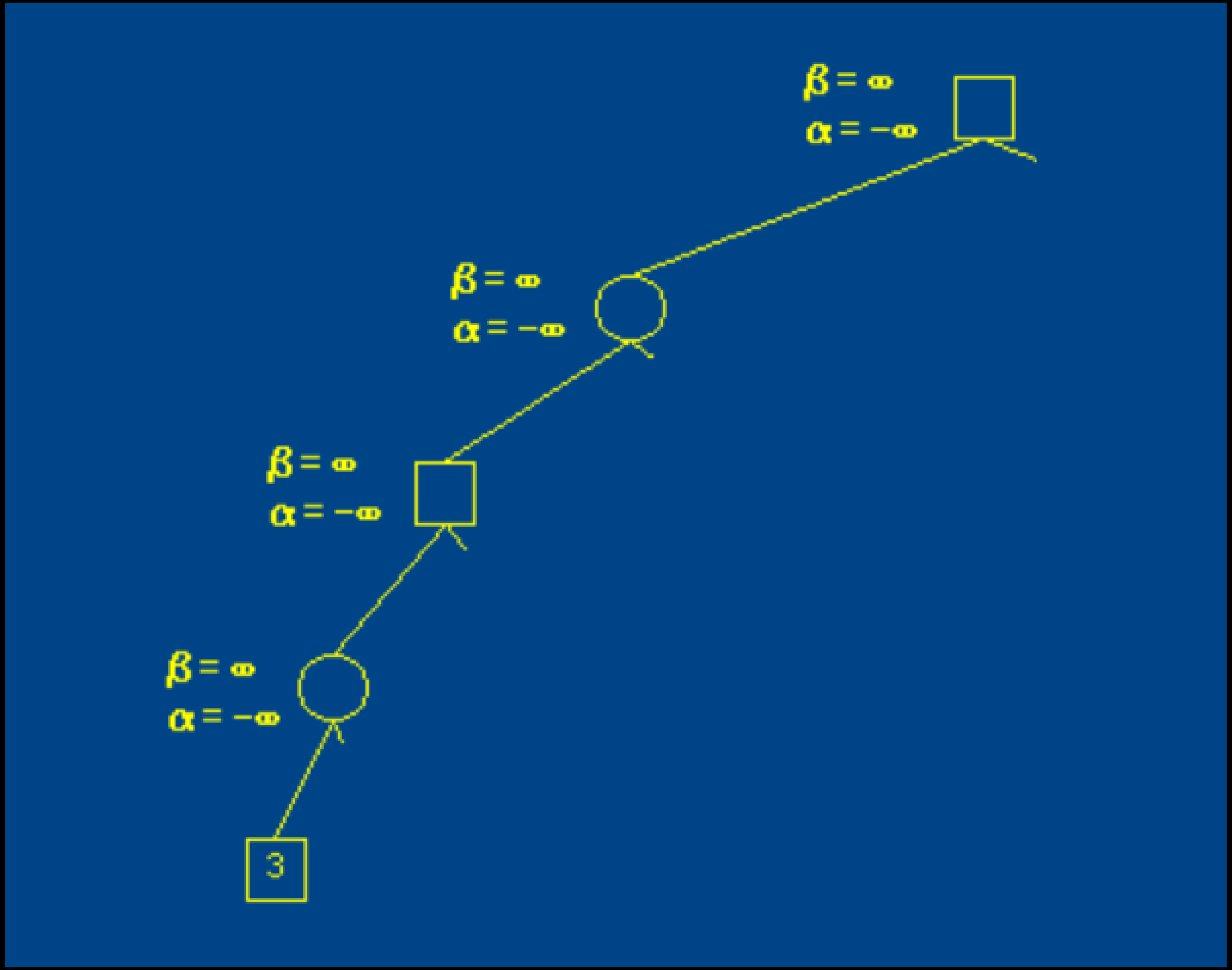
HARD: USING MINIMAX WITH ALPHA BETA PRUNING

- The minimax algorithm is a way of finding an optimal move in a two player game. *Alpha-beta pruning* is a way of finding the optimal minimax solution while avoiding searching subtrees of moves which won't be selected.
- Minimax algorithm consists of:
  - Min node → Your opponent
  - Max node → Yourself
  - Evaluation function → Applied at terminal nodes

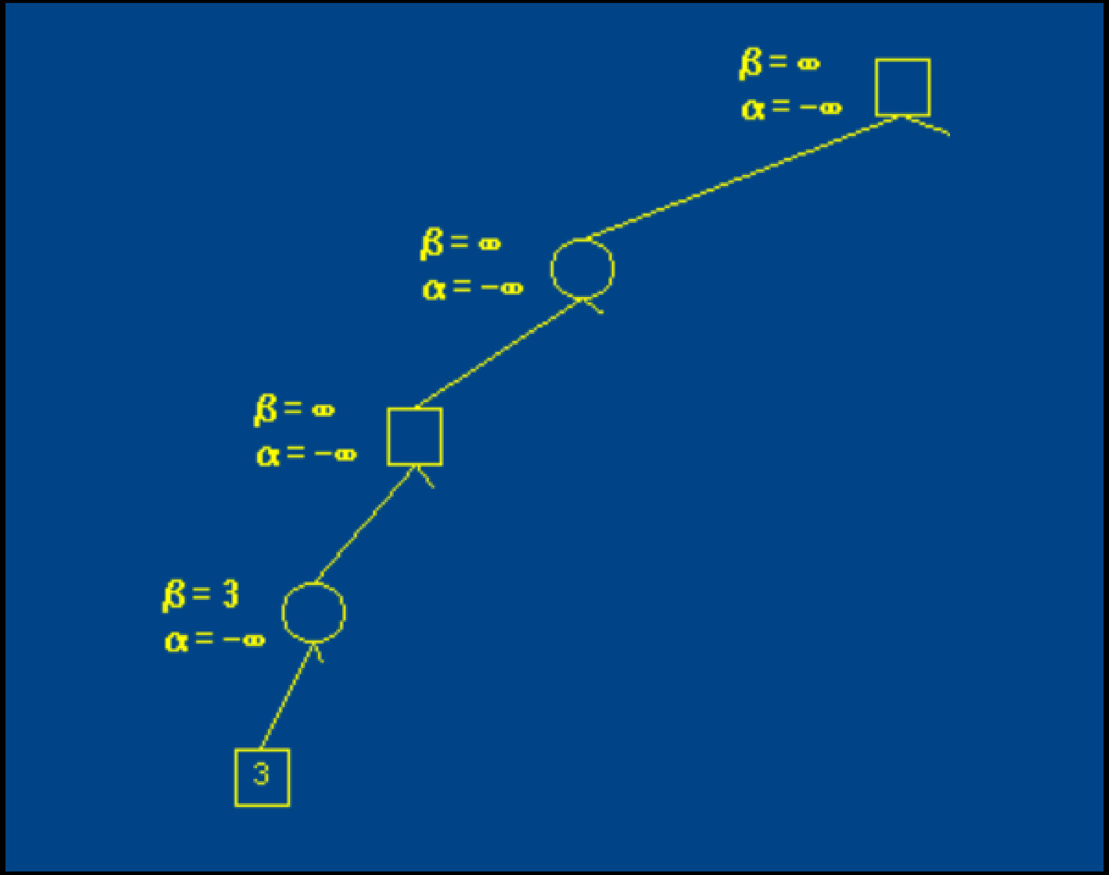
- Here is an example: Nodes representing your moves are generally drawn as squares; nodes representing your opponent's moves are generally drawn as circles. This is a given tree →



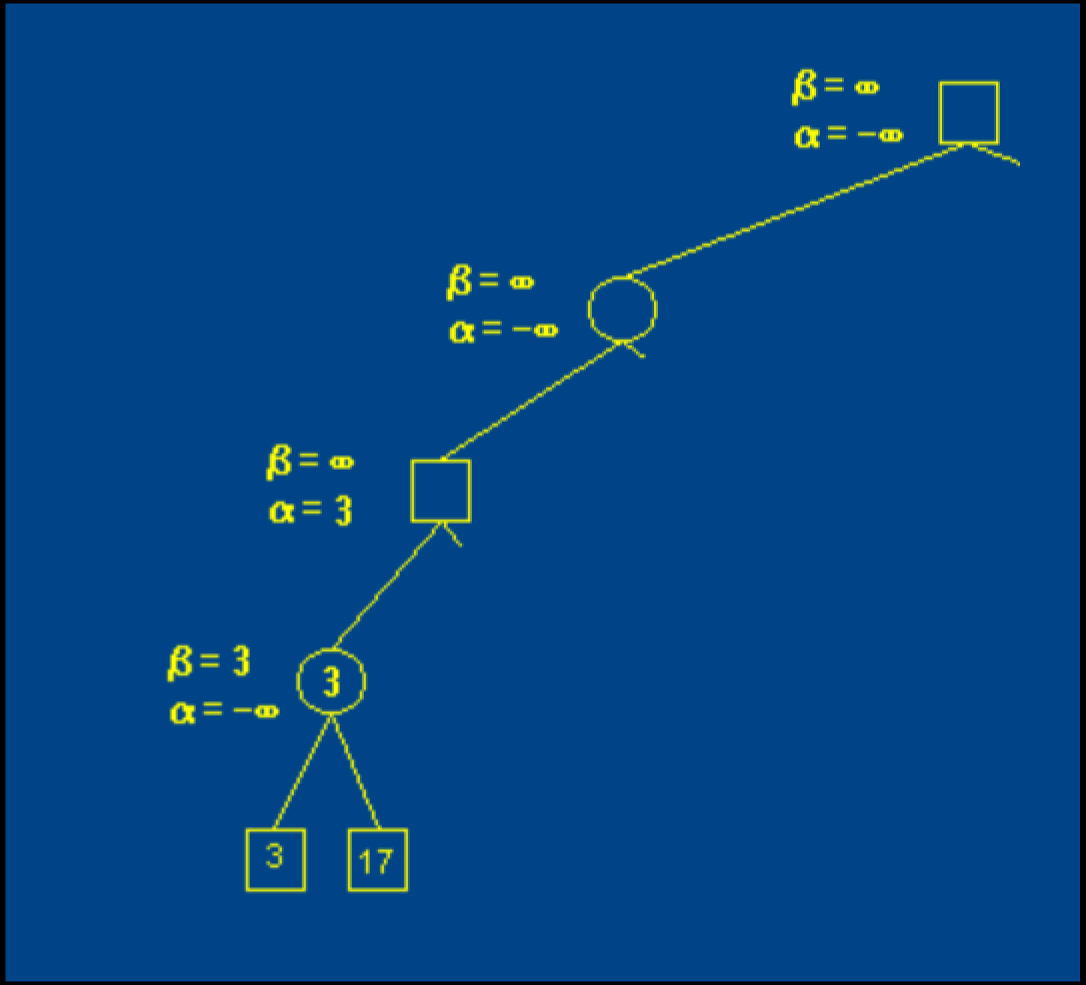
- Now, we start to implement the concept of minimax algorithm with alpha beta pruning.
- We get to the first node at depth 4, which is a terminal node. We apply the evaluation function here, and get the value 3. Thus we have this →



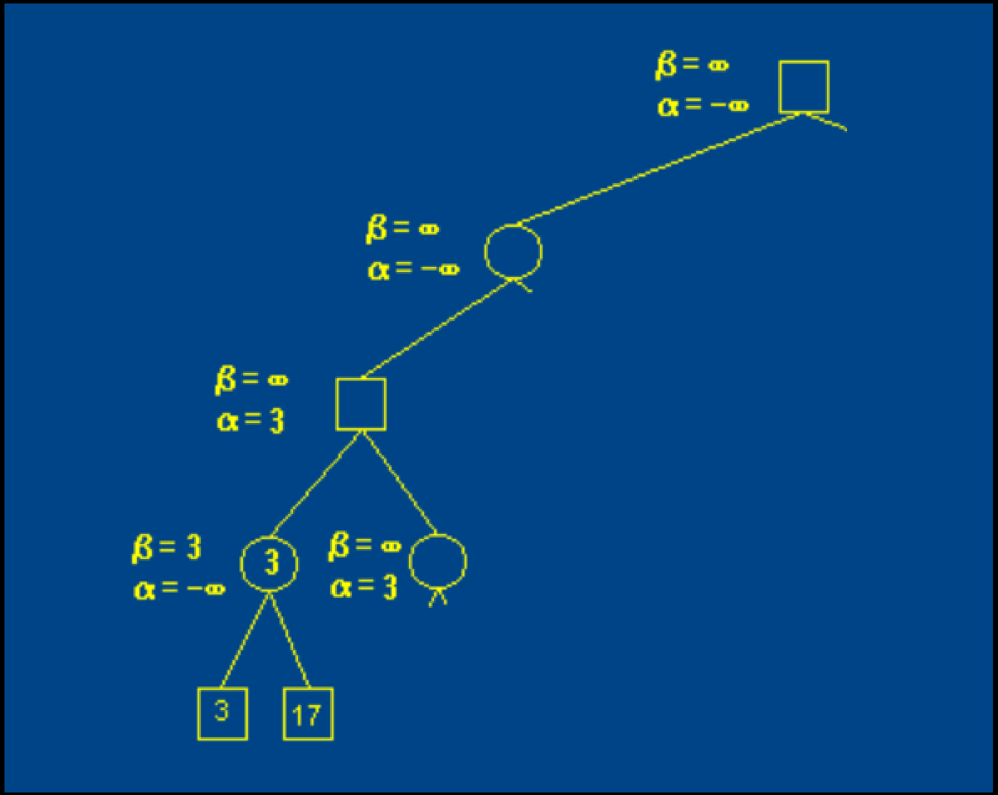
- We pass this node back to the min node above.
- Since this is a min node, we now know that the minimax value of this node must be less than or equal to 3. In other words, we change beta to 3.



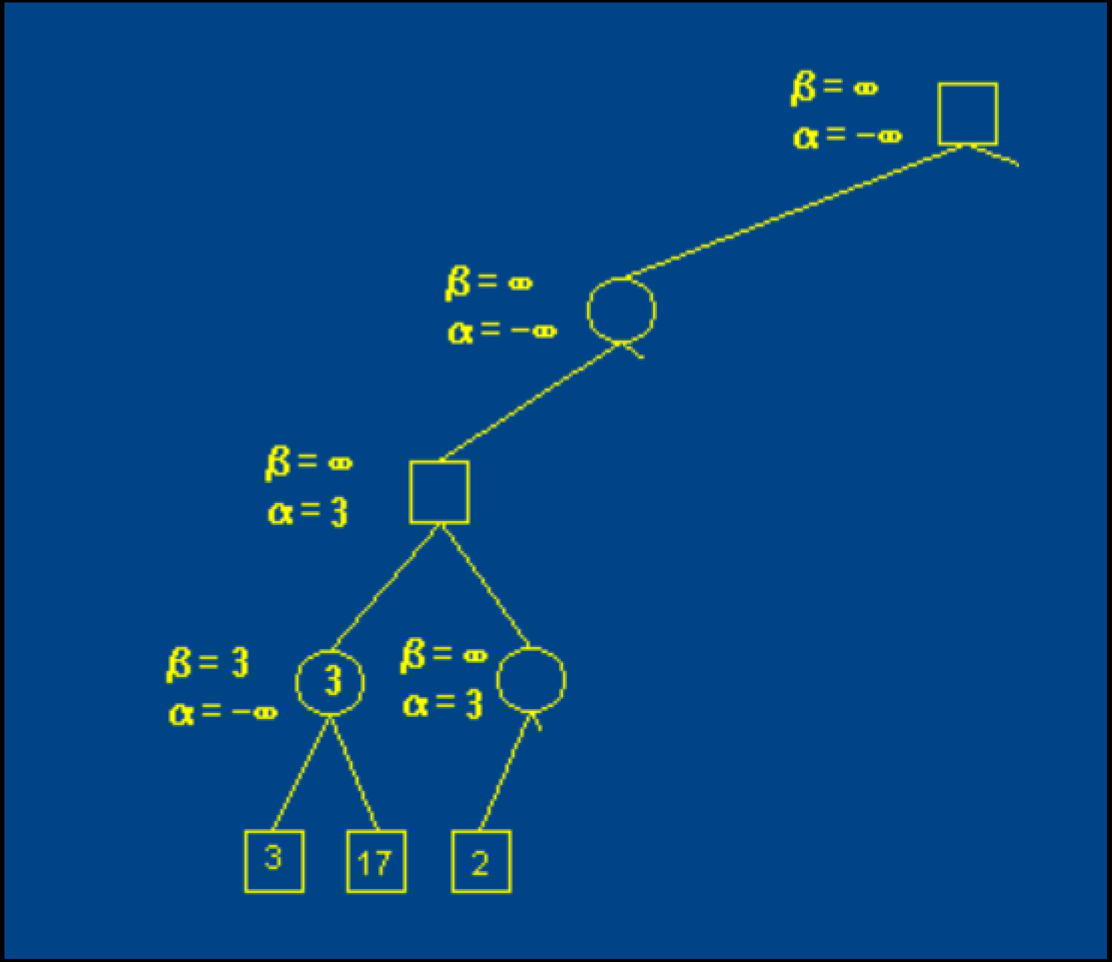
- Note that the alpha and beta values at higher levels in the tree didn't change. When processing actually returns to those nodes, their values will be updated.
- Next we generate the next child at depth 4, implement our evaluation function, and return a value of 17 to the min node above:
- Since this is a min node and 17 is greater than 3, this child is ignored. Now we've seen all of the children of this min node, so we return the beta value to the max node above. Since it is a max node, we now know that it's value will be greater than or equal to 3, so we change alpha to 3.



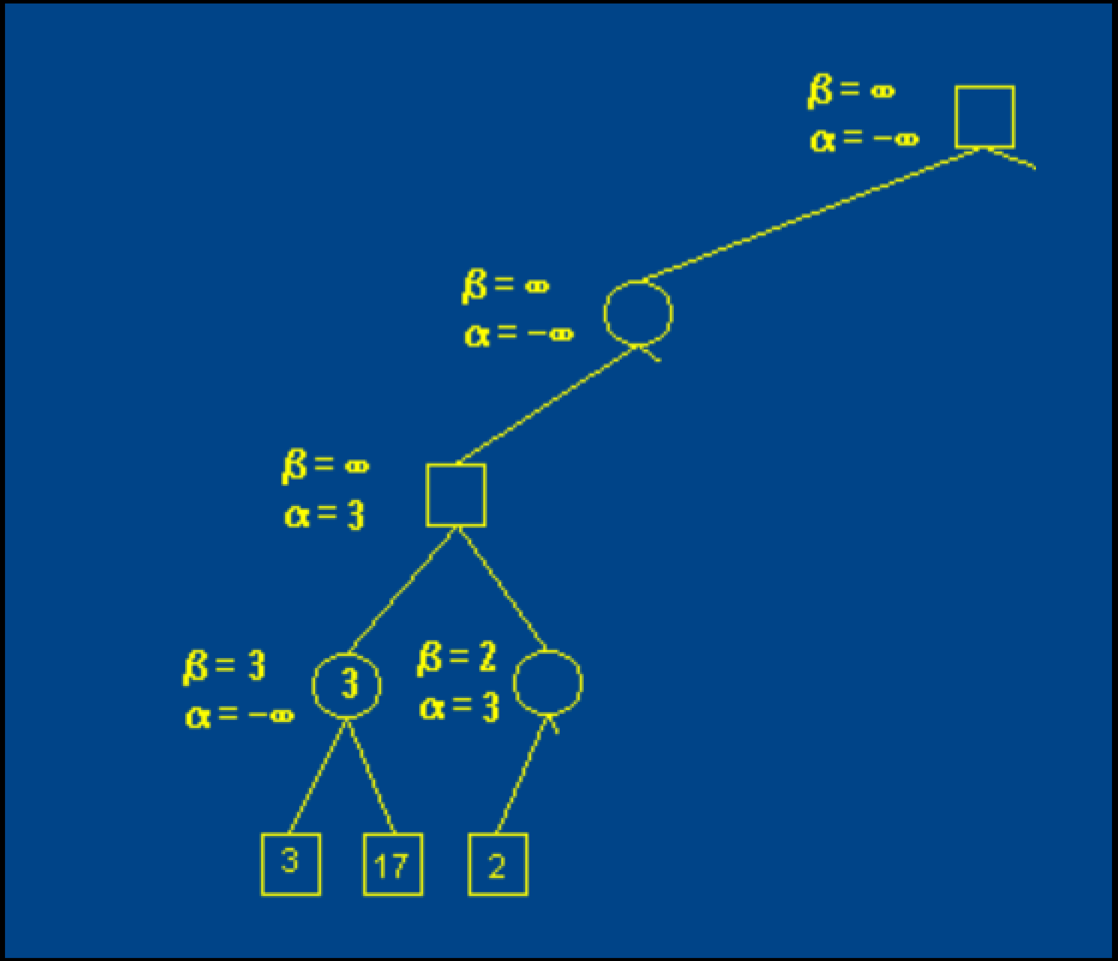
- We generate the next child of the current max node and pass the bounds along:



- Since the current min node is not at the target depth, we generate its first child, implement the evaluation function on that node, and return it's value →

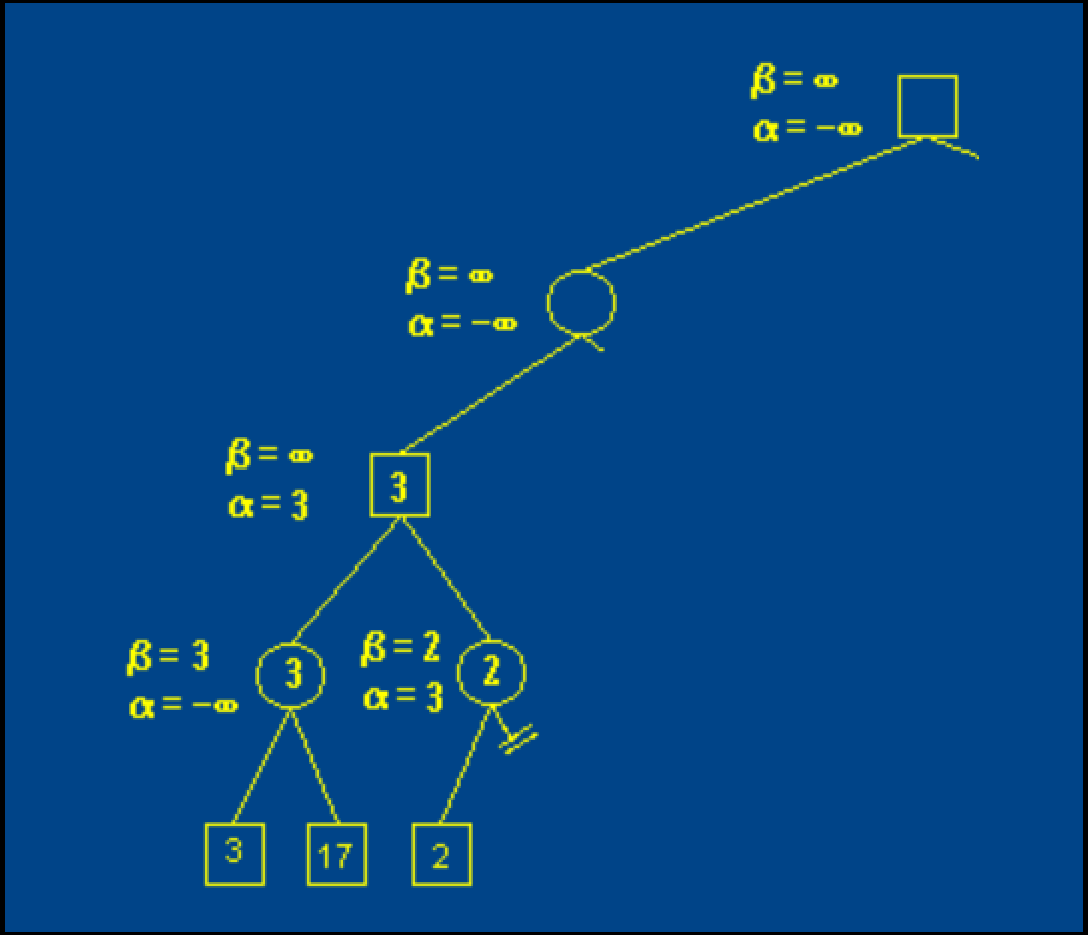


- Since this is a min node, we now know that the value of this node will be less than or equal to 2, so we change beta to 2 →

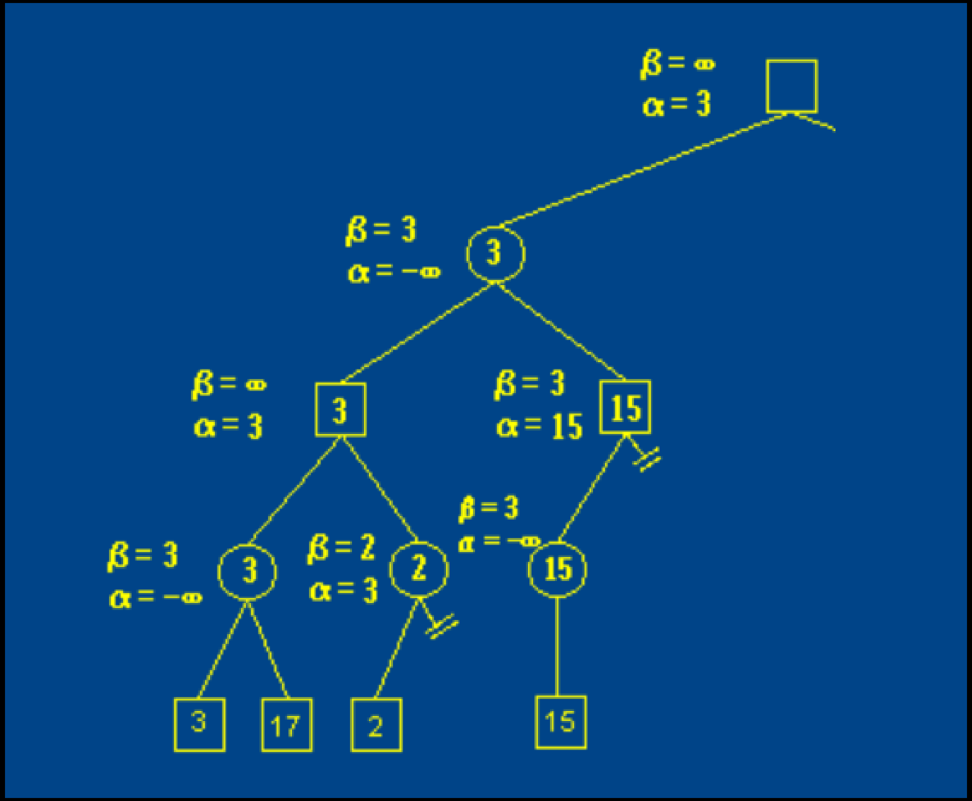


- The only way we could find a solution path at this min node is if we found a child node with a value that was both greater than 3 and less than 2. Since that is impossible, we can stop evaluating the children of this node, and return the beta value (2) as the value of the node.
- Admittedly, we don't know the actual value of the node. There could be a 1 or 0 or -100 somewhere in the other children of this node. But even if there was such a value, searching for it won't help us find the optimal solution in the search tree. The 2 alone is enough to make this subtree fruitless, so we can prune any other children and return it.

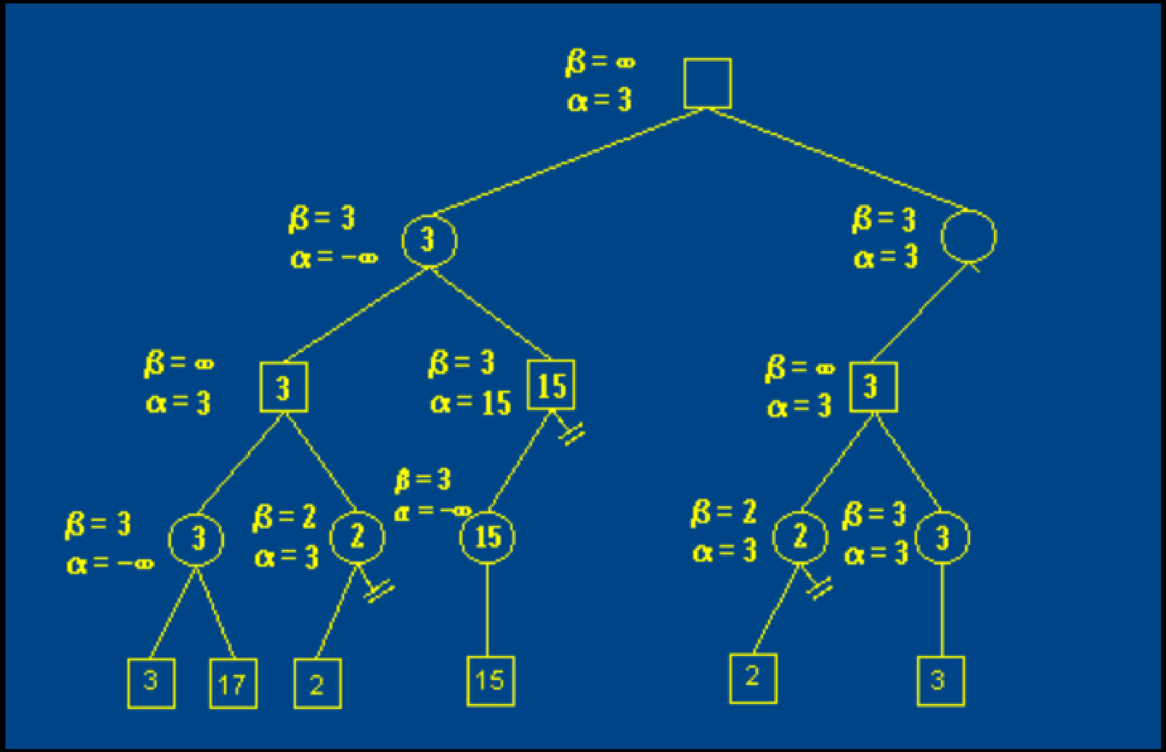
- Back at the parent max node, our alpha value is already 3, which is more restrictive than 2, so we don't change it. At this point we've seen all the children of this max node, so we can set its value to the final alpha value →



- The same concept applies to the parent min node and the results are:



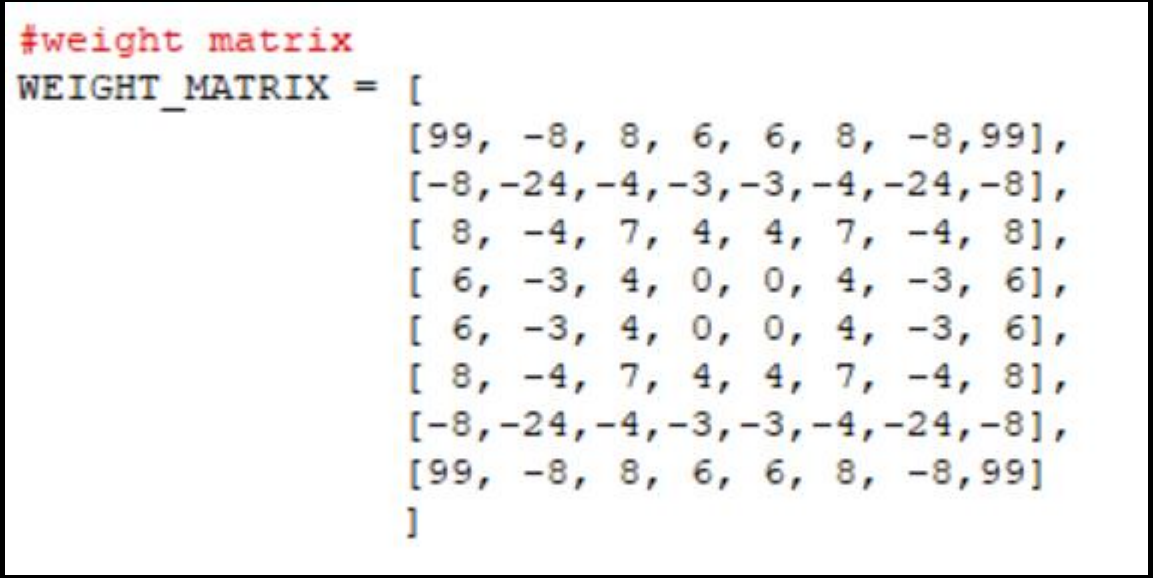
- In turn, the same concept applies to the parent max node and the results are →
- By using this method, it is possible to find the optimal strategy without searching through all the subtrees.



Reference:
CS 161 Recitation Notes - Minimax with Alpha Beta Pruning.
Retrieved from http://web.cs.ucla.edu/~rosen/161/notes/alphabeta.html

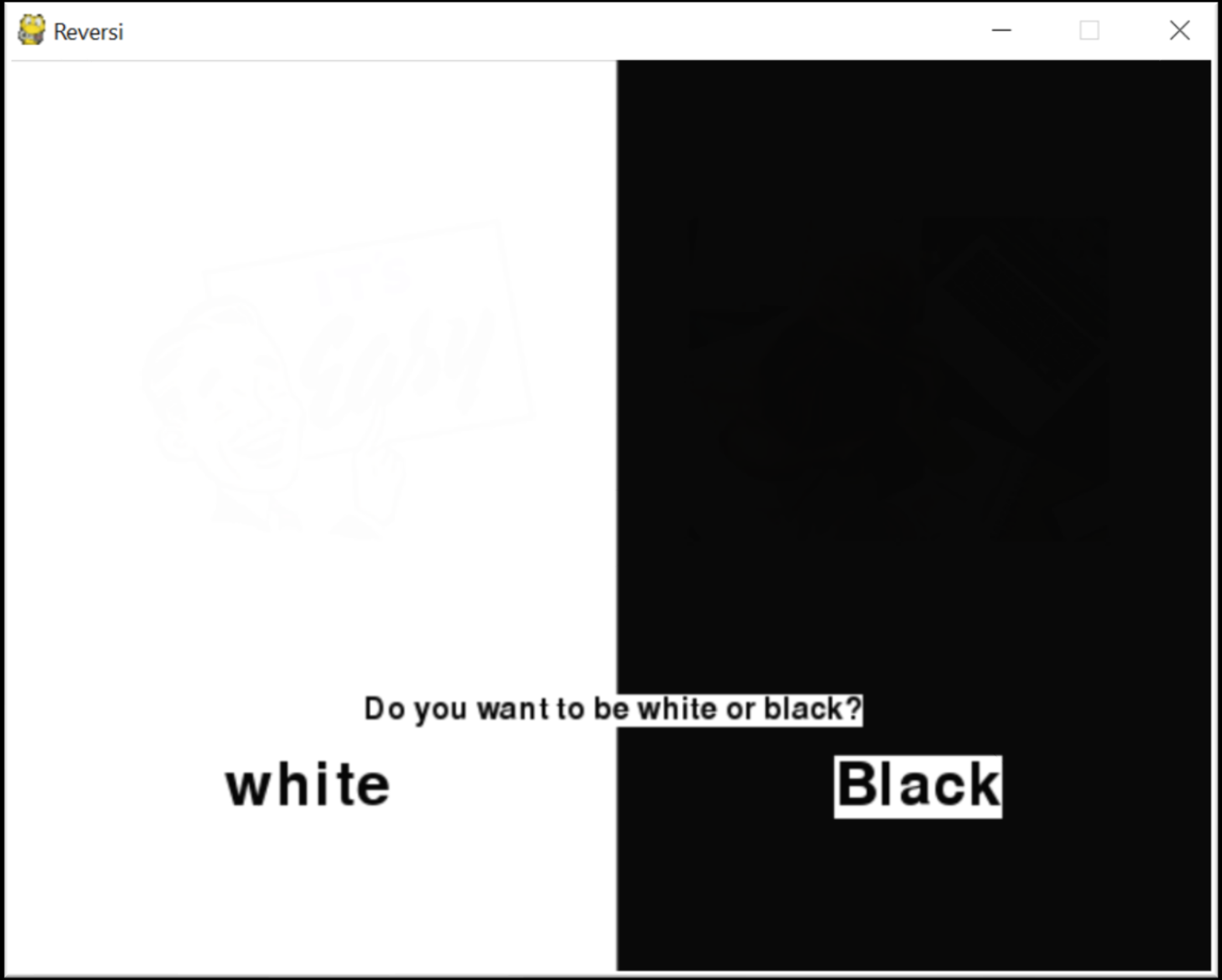
APPLICATION OF ALPHA BETA PRUNING IN OUR PROJECT: SET UP A CLASS - "ALPHABETASEARCH"

- Evaluation function is implemented by traversing the board at terminal nodes ("getUtility").
- Alpha beta pruning for max node ("getMaxValue").
- Alpha beta pruning for min node ("getMinValue").
- Return the board state after executing the optimal strategy ("executeSearch").



```
#weight matrix
WEIGHT_MATRIX = [
                [99, -8, 8, 6, 6, 8, -8,99],
                [-8,-24,-4,-3,-3,-4,-24,-8],
                [ 8, -4, 7, 4, 4, 7, -4, 8],
                [ 6, -3, 4, 0, 0, 4, -3, 6],
                [ 6, -3, 4, 0, 0, 4, -3, 6],
                [ 8, -4, 7, 4, 4, 7, -4, 8],
                [-8,-24,-4,-3,-3,-4,-24,-8],
                [99, -8, 8, 6, 6, 8, -8,99]
                ]
```

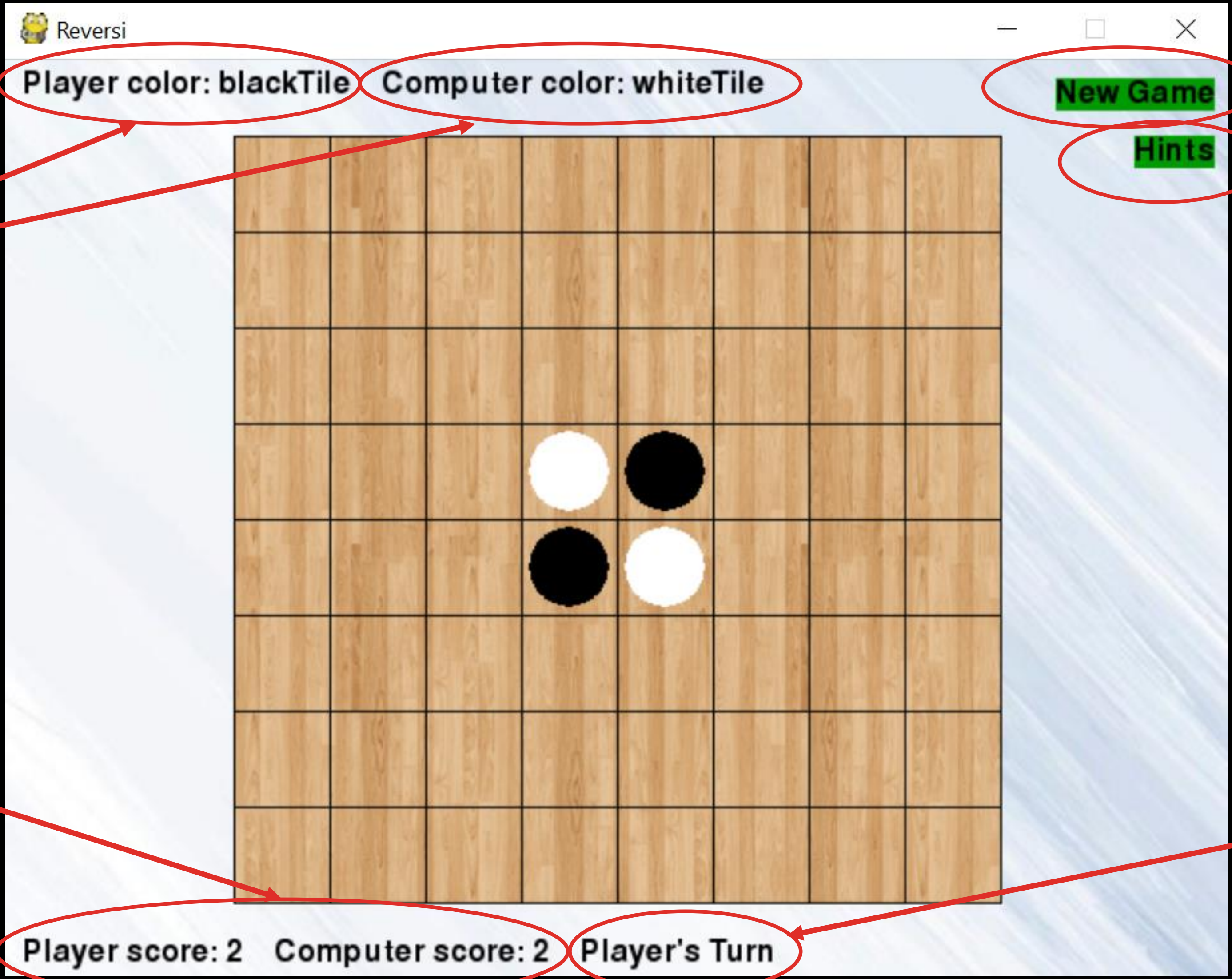
4. CHOOSING TILE COLOR



Reversi

Do you want to be white or black?

5. ENTERING THE GAME


Reversi

Player color: blackTile Computer color: whiteTile

Player score: 2 Computer score: 2 Player's Turn

Start a new game again

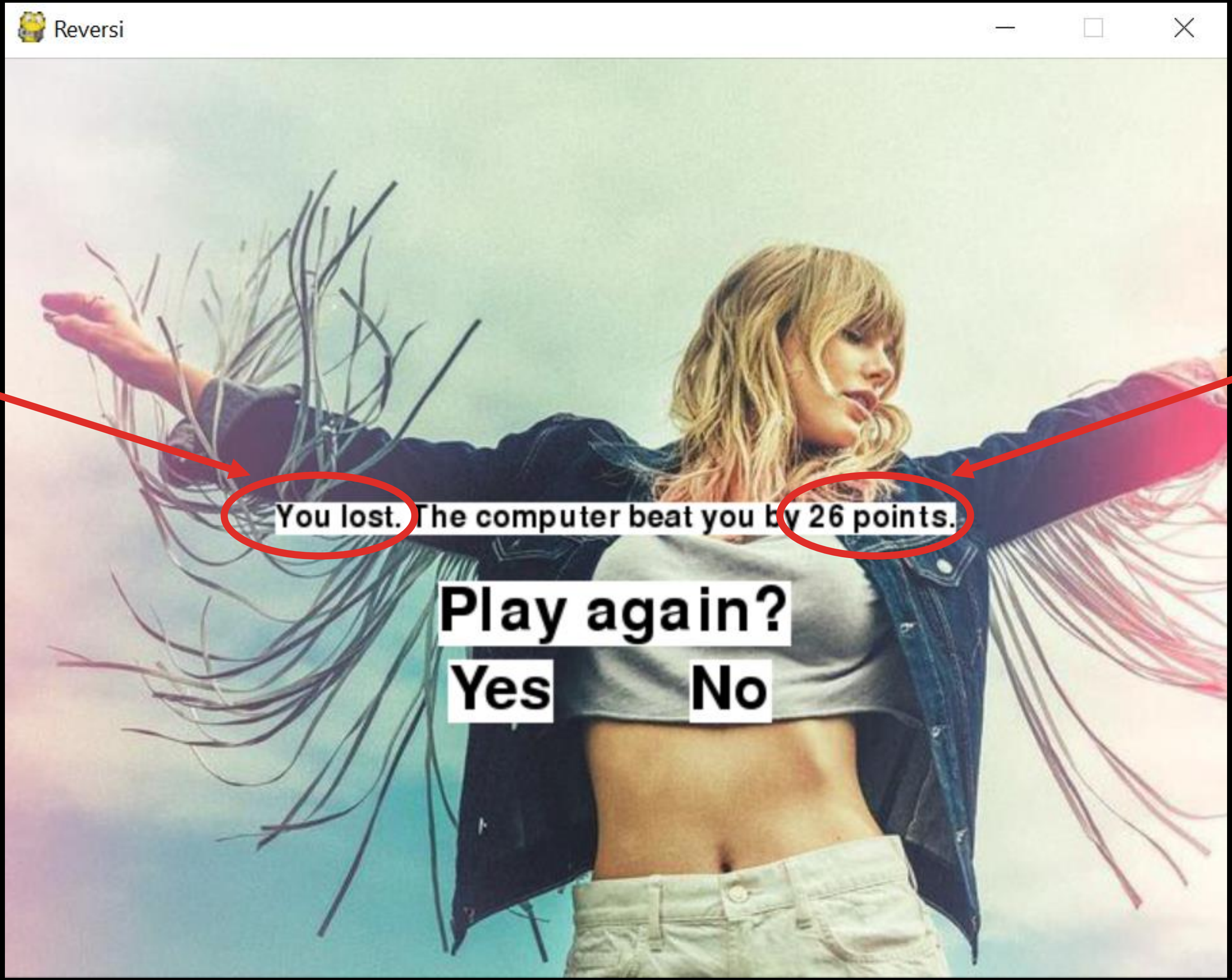
Click to show hints

Tile color

Scoring (1 piece = 1 point)

Showing the current turn

6. ENDING THE GAME



You lost. The computer beat you by 26 points.

Results of the game

Points difference

DEMONSTRATION VIDEO

Youtube video link:

https://youtu.be/gx-g1rC6p_Q

THANK YOU FOR LISTENING!

- Q&A Session -