BlutForce, simple yet strong.

a Tablut player.

Architecture and language

1. We decided to use Java for its strong typing and for the possibility to reuse and extend the classes already present in the project.

2. The two players were made using the aima library. Aima implements different search algorithms. Furthermore, it provides the interfaces to be implemented in order to correctly use search strategies.

3. Several heuristics has been created for the evaluation of the states. These ones were made using our knowledge of the game.

4. MinMax with alpha-beta pruning and iterative deepening has been used in order to explore as much node of the tree as possible within the time allocated for each move of the player,

Heuristics

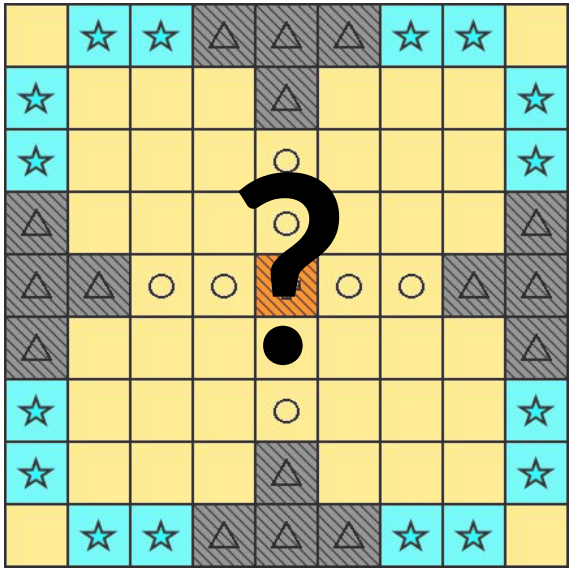
Heuristics are used for the state evaluation and results to be crucial for the game. Good heuristics brings to victory.

They are divided into white and black heuristics.

Some heuristics are equal for both players except for the weight assigned to them.

For example, in a state of the board where the king can escape in two moves a positive weight is assigned to the white player and a negative for the black.

Another example is the value associated with the loss of a pawn.
The pawn loss for the white player is more penalizing than the black counterpart, due to the smaller number of pieces.



White heuristics

In particular, for the white player, we implemented the following heuristics (some of them are explained in the next slide):

- King Protected;
- King In Castle;
- Best Position;
- King Near Castle;
- White Loosed
- Black Eaten
- Escape Free
- Escape Path Free
- King Almost Captured
- Pawns Under Attack
- Plus some bonuses for specific situations during the game.

White heuristics

- **King Protected:** King have at least three white pawns around;
- **Best Positions:** Some position on the board are considered more important for the white player to maintain control of the center;
- **Escape Free:** There are no pawns between the king and one of the escape cells. So the King can escape in a single move;
- **Escape Path Free:** Similar to the previous one, but we are looking for a two-moves escape in this case;
- **King Almost Captured:** Used when there is only one missing black pawn in order to capture the King;
- **Pawns Under Attack:** Used to compute the number of white pawns that have one black pawn near them.

Black heuristics

In particular, for the black player, we implemented the following heuristics:

- Black Loosed
- White Eaten
- King Almost Captured
- King Under Attack
- Escape Free
- Escape Path Free

Black heuristics

Black heuristics are much simpler than white ones.

The black player focuses on exploring as many nodes as possible maximizing the number of white pawns eaten while keeping exit points under control.

Performance evaluation

Here are some statistics on the performance of our players.
We can notice how the larger is the timeout for each move, the higher will be the number of expanded nodes, but not in linear way.
Note: performance are evaluated on the virtual machine.

	Depth		Nodes expanded	
Avg in 10 seconds	3	4	130K	75K
Avg in a minute	4	4	800K	500K
	For black	For white	For black	For white

Thanks for your attention.

Developed and presented by

- Andrea Zecca
- Antonio Lopez
- Stefano Colamonaco