



André Gomes 97541
Pedro Monteiro 97484
José Trigo 98597



universidade
de aveiro

Table of contents

01

The Algorithm

02

Heuristics

03

Look Ahead

04

Results

05

Conclusion

06


References



The Algorithm

To develop the Tetris playing artificial intelligence, the first step was to find a good approach to how to solve the problem.

After checking what was provided to us, we decided to calculate, for each tetris piece, the best possible position (using **Heuristics**, described in the next slide), and updating the game with that position, so that the next iteration would take in consideration the previous one.



We also used a **cache** to avoid recalculating every possible combination of starting positions for already known pieces.



The Heuristics

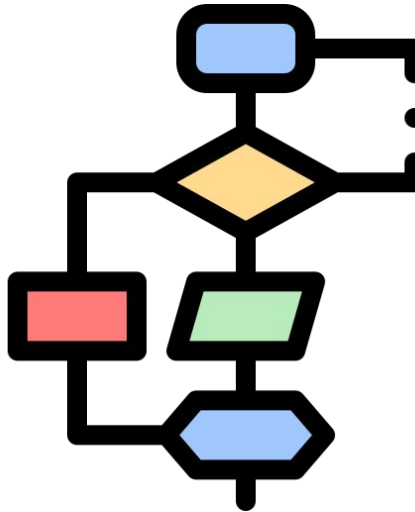
In order to evaluate how a piece should be played, we did a little research and found out there were **4 heuristics** in particular that we should pay more attention to:

- **Bumpiness**: given by the sum of the height difference of all the columns;
- **Heights**: given by the sum of the height of all the columns;
- **Completed Lines**: given by the number of lines completed in the game;
- **Holes**: given by the number of holes found in the game at the moment;

After addressing the right weight to each one of the heuristics (check the 1st website in the references), we added a condition to change how the game was played if we reached a certain height, by giving more weight to the **Completed Lines** heuristic.

Look Ahead

In order to implement this functionality we see the best 5 actual candidate moves, put them in the board and then calculate the cost/evaluation for the next piece, returning 5 candidate moves for each situation.



Results

Without implementing the Look-Ahead and adding the heuristic condition, we managed to reach the high-score of **659 points**, with an average per 10 trials of about **151 points**.

After making the game be played with the knowledge of what 1 piece will come up next and with the heuristic change, we reached a high-score of **687 points**, with an average per 10 trials of about **300 points**.

As we can see, these small changes were able to pump up both the high-score and the average points.



Conclusion

To sum up everything that has been said to this point, with the difference in approaches to the problem from the 1st delivery date to the 2nd, we were able to notice the difference in how the AI behaved.

References



- [Tetris AI – The \(Near\) Perfect Bot | Code My Road](#)
 - [How to write a Tetris Bot in Python](#)
- 