

Nama : Nigel Sahl  
NIM : 13521043

## 1010! Bot

**Secara umum**, algoritma kecerdasan buatan yang digunakan dalam bot untuk gim 1010 adalah algoritma greedy dengan pendekatan heuristik. Algoritma ini digunakan untuk mencari langkah terbaik yang dapat diambil oleh bot untuk menempatkan bentuk-bentuk (figures) di papan permainan sehingga dapat mencetak skor tertinggi berdasarkan beberapa kriteria evaluasi.

Cara kerja algoritma intelegensi buatan untuk gim 1010!:  
Algoritma ini diimplementasikan dalam fungsi utama `best_move`

### **best\_move:**

Fungsi ini digunakan untuk memilih langkah terbaik yang dapat diambil oleh bot berdasarkan pendekatan heuristik. Algoritma berjalan sebagai berikut:

- Menggunakan `predict_next_types` untuk memperkirakan beberapa tipe bentuk berikutnya. (tidak jadi digunakan karena memerlukan waktu yang sangat lama dalam fungsi selanjutnya yaitu `score_all_pos` untuk menghitung score semua kemungkinan dari figures yang telah diprediksi)
- Mencari semua kemungkinan posisi di papan permainan untuk setiap tipe bentuk berikutnya.
- Menghitung skor untuk setiap posisi berdasarkan fungsi `score_field`.
- Memilih langkah dengan skor tertinggi sebagai langkah terbaik yang dapat diambil oleh bot.
- Pendekatan greedy digunakan di sini karena bot hanya mencoba beberapa langkah ke depan dan memilih langkah yang memberikan skor tertinggi pada saat itu.

Penyebab tidak menggunakan `predict_next_types` dan `score_all_pos` dalam pengambilan keputusan:

`predict_next_types`: Meskipun algoritma ini berguna untuk memprediksi tipe bentuk berikutnya, prediksi bentuk yang hanya berdasarkan probabilitas kemunculan tidak selalu akurat. Sebagai gantinya, algoritma lebih memilih untuk mengevaluasi posisi yang memungkinkan secara langsung berdasarkan skor yang dihitung dengan heuristik pada saat itu.

`score_all_pos`: Fungsi ini menghitung skor untuk semua kemungkinan posisi bentuk berikutnya (yang telah diprediksi) dan semua kemungkinan penempatannya dalam field, namun karena waktu dan memori yang diperlukan, ia memakan waktu yang sangat lama dan tidak praktis untuk digunakan dalam mencari langkah terbaik. Sebagai alternatif, algoritma menggunakan pendekatan greedy dengan `best_move` untuk mencari langkah terbaik yang lebih efisien.

### **score\_field:**

Fungsi ini digunakan untuk menilai skor dari papan permainan berdasarkan tiga kriteria evaluasi heuristik:

- Jumlah garis (lines) yang dapat terbentuk: Semakin banyak garis yang terbentuk setelah menempatkan sebuah bentuk, semakin tinggi skor yang diberikan.

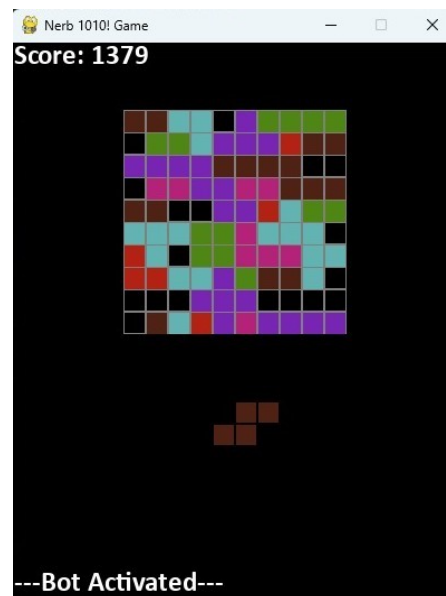
- Jumlah nol (0) dalam satu baris (rows): Semakin sedikit nol dalam satu baris, semakin tinggi skor yang diberikan.
- Jumlah grup nol (0) dalam setiap baris: Semakin sedikit grup nol (0) dalam setiap baris, semakin tinggi skor yang diberikan.

Algoritma berjalan sebagai berikut:

- Melakukan iterasi pada setiap baris dalam papan permainan.
- Menghitung jumlah nol dalam setiap baris dan mencatat jumlah baris yang terisi penuh tanpa nol.
- Menghitung jumlah grup nol dalam setiap baris dengan menghitung perubahan dari angka non-nol ke angka nol dalam setiap baris. (nol merepresentasikan field tersebut masih kosong)
- Menghitung skor berdasarkan jumlah garis yang terbentuk, jumlah nol minimum dalam satu baris, dan jumlah grup nol dalam setiap baris.
- Pada implementasinya penulis hanya menggunakan jumlah garis dan grup nol karena merasa itu cukup dan memberikan hasil yang cukup bagus pada beberapa percobaan yang dilakukan.

```
score_lines = lines ** 2
score_rows = rows - min_zeroes
score_group_zeroes = rows*cols - sum_of_group_zeroes
# score = score_lines * 3 * rows + score_rows
# score = rows*cols * score_lines * 3 + score_rows * rows + score_group_zeroes
score = rows*cols * score_lines * 3 + score_group_zeroes
return score
```

Terdapat beberapa perhitungan score dari 3 pendekatan secara heuristic, pada pendekatan yang hanya mengandalkan score\_lines dan score\_rows memberikan score akhir permainan yang sangat kecil pada rentang puluhan. Kedua, untuk pendekatan yang mengandalkan semua komponen memberikan hasil yang cukup baik di rentang sekitar 600-an. Terakhir adalah pendekatan yang mengandalkan score\_lines dan score\_group\_zeroes yang memberikan nilai tertinggi yaitu 1379 (high score sementara).



Heuristic Score:

Skor heuristik (heuristic score) digunakan untuk mengevaluasi kualitas dari suatu posisi pada papan permainan. Algoritma ini memiliki beberapa kriteria evaluasi yang membantu menentukan langkah terbaik yang dapat diambil oleh bot. Kriteria evaluasi tersebut adalah:

1. Jumlah Garis (Lines) yang Dapat Terbentuk:  
Semakin banyak garis yang terbentuk setelah menempatkan sebuah bentuk, semakin tinggi skor yang diberikan. Ini berarti bahwa bot mencoba untuk menempatkan bentuk di tempat yang dapat membentuk sebanyak mungkin garis secara horizontal maupun vertikal, karena garis yang terbentuk akan memberikan skor yang lebih tinggi.
2. Jumlah Nol (0) dalam Satu Baris (Rows):  
Semakin sedikit nol dalam satu baris, semakin tinggi skor yang diberikan. Bot cenderung menghindari menempatkan bentuk di baris yang memiliki banyak nol, karena baris dengan sedikit nol cenderung lebih mudah diisi dan membentuk garis.
3. Jumlah Grup Nol (0) dalam Setiap Baris:  
Semakin sedikit grup nol dalam setiap baris, semakin tinggi skor yang diberikan. Bot mencoba untuk menempatkan bentuk di baris yang menjadikan jumlah grup nol sedikit, karena ini memberikan peluang untuk lebih mudah mengisi dan membentuk garis.

Pendekatan heuristic score membantu bot dalam memilih langkah terbaik dengan mengevaluasi potensi dari setiap posisi berdasarkan skor yang dihasilkan oleh tiga kriteria evaluasi tersebut. Pendekatan ini memungkinkan bot untuk mencari langkah yang mengoptimalkan peluang untuk membentuk garis dan mencetak skor tertinggi.