

KELOMPOK BINGUNG

FINAL PROJECT KKA

BY:

1. Nadya Saraswati Putri (5025221246)
2. Helsa Sriprameswari Putri (5025221154)
3. Yasmin Putri Sujono (5025221273)

KELOMPOK BINGUNG

DESKRIPSI PERMASALAHAN

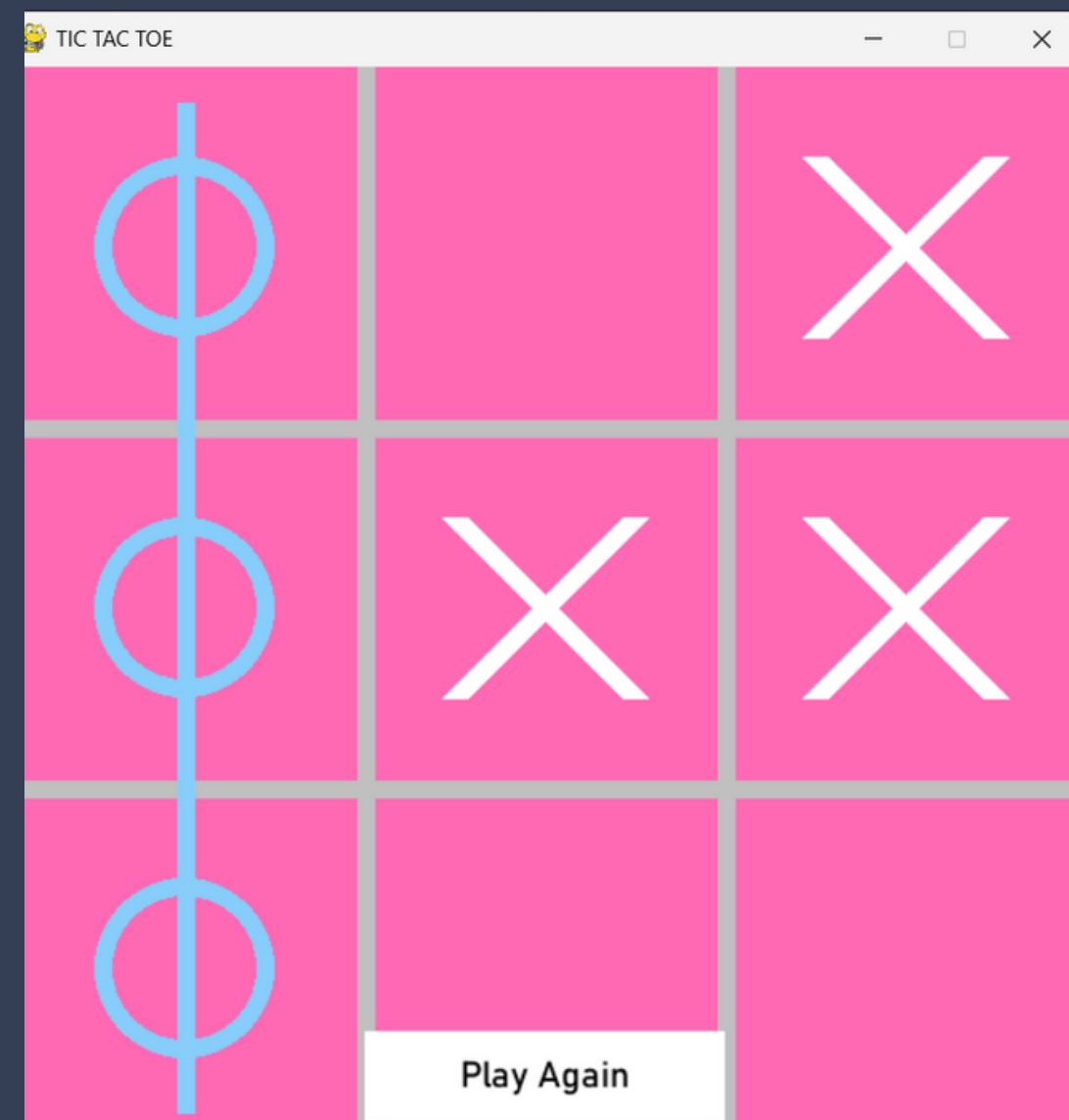
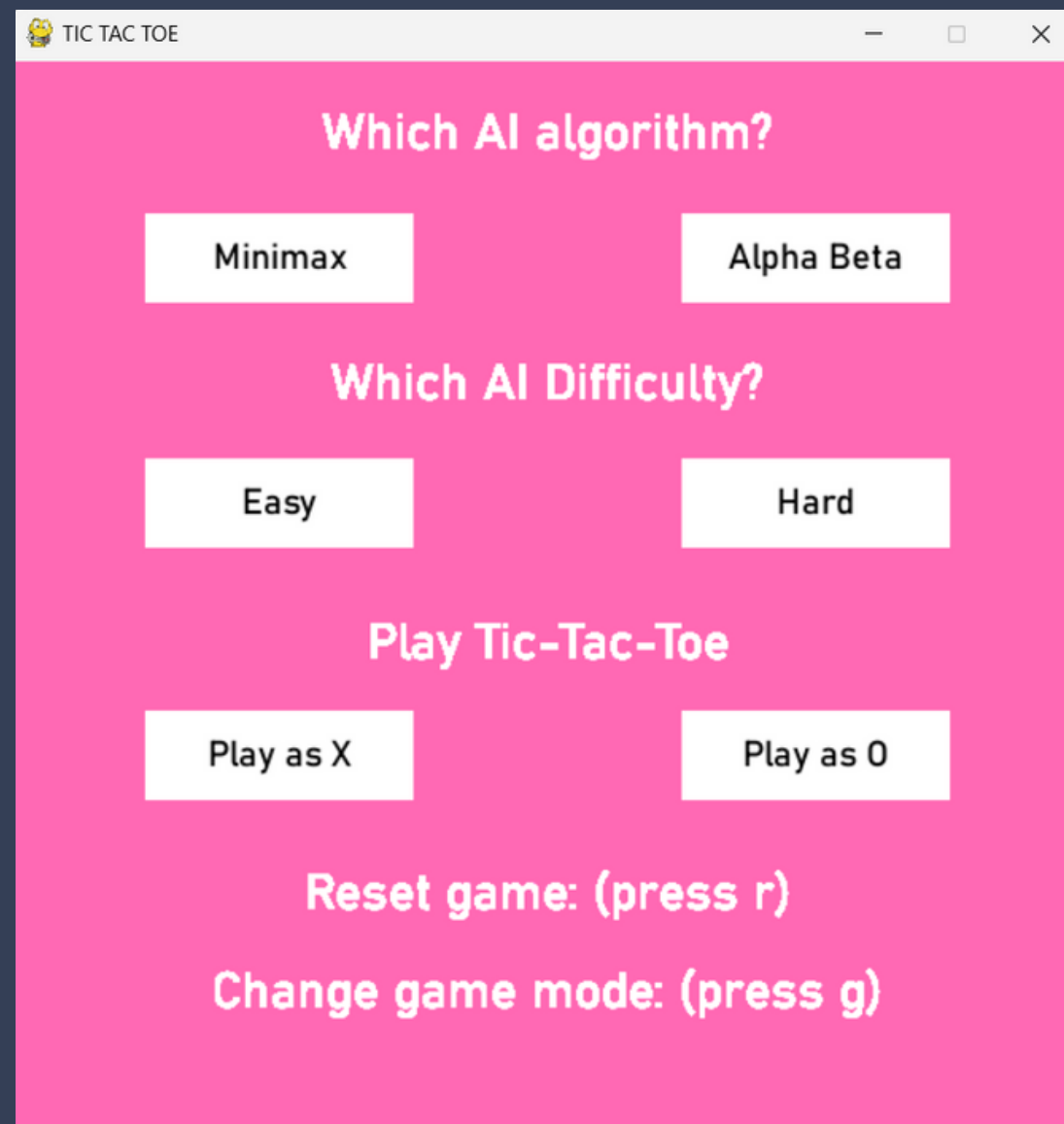
DESKRIPSI KASUS

PERMAINAN TIC TAC TOE

Tic Tac Toe merupakan permainan papan yang berisi kotak-kotak berukuran 3x3. Pada program ini, permainan hanya memiliki satu mode, yaitu mode melawan AI dengan beberapa pilihan. Diantaranya pilihan kesulitan yaitu easy dan hard, pilihan algoritma yang ingin digunakan, yaitu algoritma "Minimax" dan "Alpha Beta Pruning", dan pilihan simbol X atau O yang akan dimainkan player. Dalam game ini juga terdapat 3 kondisi, yaitu menang, kalah, dan draw.

KELOMPOK BINGUNG

TAMPILAN GAME



KELOMPOK BINGUNG

METODE PERMASALAHAN

METODE

1.ALGORITMA MINIMAX

- 1.Inisialisasi pohon permainan : Membuat pohon permainan di mana setiap simpul mewakili keadaan permainan saat itu.
- 2.Evaluasi simpul daun : Evaluasi dengan memberikan skor berdasarkan hasil permainan. Misalnya, skor positif untuk kemenangan, skor negatif untuk kekalahan, dan nol untuk seri.
- 3.Penilaian pohon : Penilaian skor dengan pemain 'X' memaksimalkan skor, sedangkan pemain 'O' meminimalkannya.
- 4.Penyaringan skor : Secara rekursif, setiap tingkat pohon akan memaksimalkan atau meminimalkan skor di bawahnya, bergantung pada giliran.

METODE

2. ALGORITMA MINIMAX

5. Pemilihan langkah terbaik : Pemain akan memilih langkah terbaik dengan memilih langkah yang menghasilkan skor terbaik pada tingkat pohon awal.
6. Melakukan langkah terbaik : Pemain akan melakukan langkah terbaik yang telah ditentukan algoritma Minimax.

METODE

2. ALGORITMA ALPHA - BETA

1. Inisialisasi pohon permainan : Membuat pohon permainan yang mewakili semua kemungkinan langkah di Tic-Tac-Toe.
2. Evaluasi simpul daun : Evaluasi setiap simpul daun dengan memberikan skor berdasarkan hasil permainan. Misalnya, skor positif untuk kemenangan, skor negatif untuk kekalahan, dan nol untuk seri.
3. Penilaian pohon dengan Alpha-Beta Pruning :
 - a. Lakukan penilaian skor setiap simpul dengan algoritma Alpha-Beta. Mulai dengan pemain 'X' sebagai pemain maksimal dan pemain 'O' sebagai pemain minimal.

METODE

2. ALGORITMA ALPHA - BETA

- b. Selama pencarian, perbarui nilai alpha dan beta pada setiap simpul sesuai dengan nilai yang ditemukan. Jika pemain maksimal, perbarui alpha dengan nilai maksimum dari alpha dan nilai yang ditemukan. Jika pemain minimal, perbarui beta dengan nilai minimum dari beta dan nilai yang ditemukan.
- c. Selama pencarian, lakukan pemotongan alpha-beta. Jika pada suatu titik alpha menjadi lebih besar atau sama dengan beta, maka dapat menghentikan pencarian lebih lanjut .
- 4. Pemilihan langkah terbaik: Pemain 'X' akan memilih langkah terbaik dengan memilih langkah yang menghasilkan skor terbaik .
- 5. Melakukan langkah terbaik: Pemain 'X' akan mengambil langkah terbaik yang telah dihitung oleh algoritma Alpha-Beta.

KELOMPOK BINGUNG

ANALISIS HASIL

ALGORITMA MINIMAX

A. Basis Kasus

- a. Jika pemain X menang , maka dikembalikan skor 1.
- b. Jika pemain O menang , maka dikembalikan skor -1.
- c. Jika permainan berakhir dengan seri, maka dikembalikan skor 0.

B. Rekursi

- a. Algoritma memasuki mode maksimisasi atau minimisasi tergantung pada giliran.
- b. Mode maksimisasi (X) bertujuan untuk memaksimalkan skor, sementara mode minimisasi (O) bertujuan untuk meminimalkan skor.
- c. Algoritma mencari sel kosong yang tersedia dan mencoba setiap kemungkinan langkah.

ALGORITMA MINIMAX

- d. Untuk setiap kemungkinan langkah, algoritma membuat salinan papan permainan dan melakukan langkah (X atau O) pada sel yang sedang dievaluasi.
- e. Algoritma melakukan rekursi pada papan permainan yang telah dimodifikasi.
- f. Setelah pemanggilan rekursif, algoritma memperoleh skor dari langkah tersebut.

C. Perbandingan Skor

- a. Dalam mode maksimisasi, jika skor saat ini lebih tinggi dari skor maksimum sejauh ini, skor maksimum dan langkah terbaik diperbarui.
- b. Dalam mode minimisasi, jika skor saat ini lebih rendah dari skor minimum sejauh ini, skor minimum dan langkah terbaik diperbarui.

ALGORITMA MINIMAX

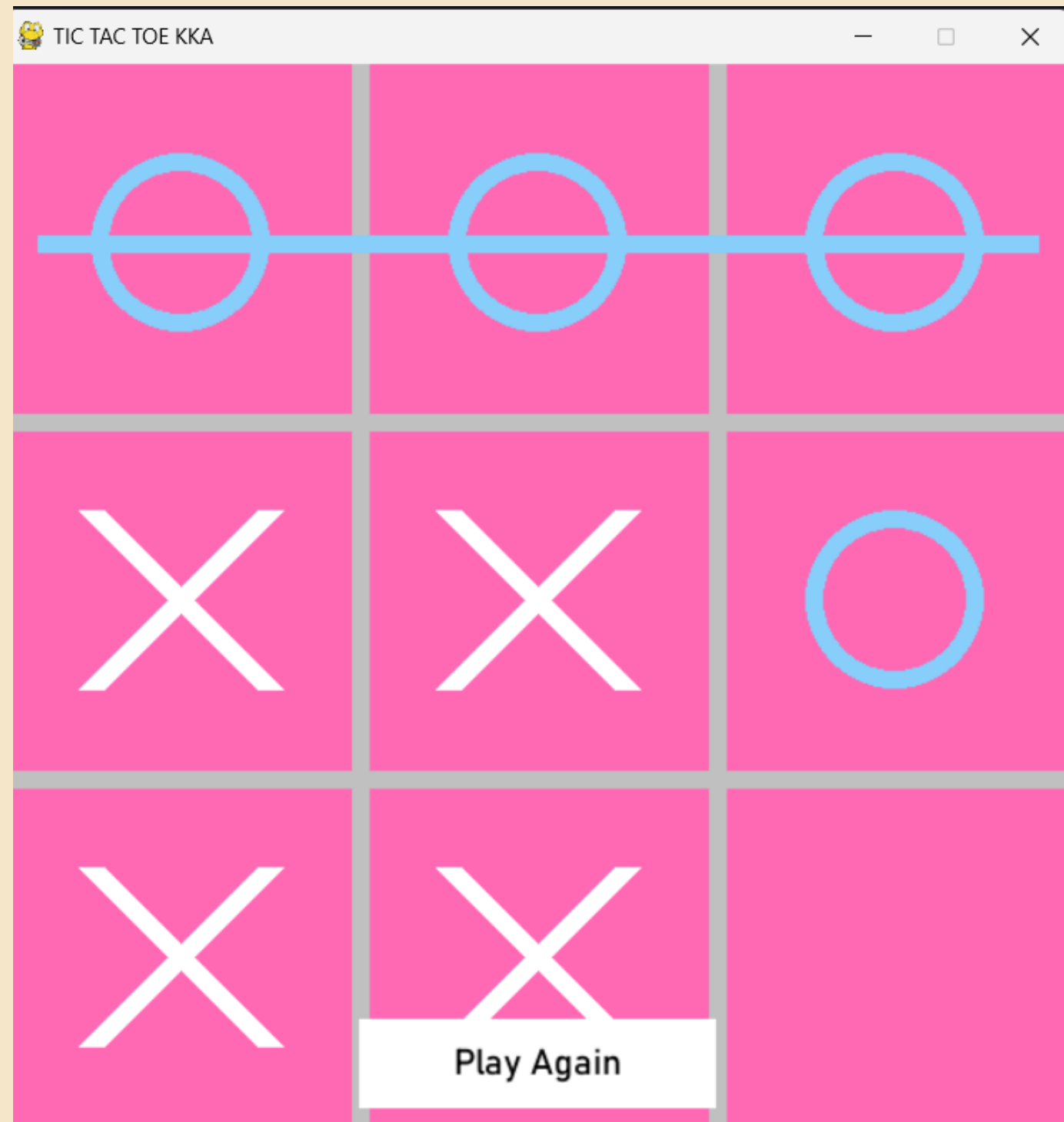
D. Kembalikan Hasil

- a. Setelah iterasi selesai, algoritma mengembalikan skor maksimum (dalam mode maksimisasi) atau skor minimum (dalam mode minimisasi) yang ditemukan selama pencarian.

E. Jenis Kesulitan

- a. Algoritma memiliki opsi untuk mengatur kesulitan permainan ("easy" atau "hard").
- b. Dalam mode "easy", algoritma membatasi pencarian hanya pada lima sel kosong pertama.

HASIL DARI KONDISI AI (SIMBOL O) MEMENANGKAN GAME.



**HASIL TAMPILAN
PROGRAM MENUNJUKKAN
POSISI DAN NILAI EVAL AI.**

```
PS C:\Users\Acer\Documents\Semester 3\KKA\Final Project\Tic-Tac-Toe> |
r 3\KKA\Final Project\Tic-Tac-Toe\main.py"
pygame 2.5.2 (SDL 2.28.3, Python 3.12.0)
Hello from the pygame community. https://www.pygame.org/contribute.ht
Minimax algorithm Chosen
AI Plays Easy
AI has chosen to mark the square in pos (0, 0) with an eval of: 0
AI has chosen to mark the square in pos (1, 2) with an eval of: 0
AI has chosen to mark the square in pos (0, 1) with an eval of: 0
AI has chosen to mark the square in pos (0, 2) with an eval of: -1
```

ALGORITMA ALPHA - BETA

A. Basis Kasus

Pada awal fungsi, terdapat pengecekan apakah permainan telah berakhir. Jika ada pemenang, fungsi mengembalikan 1 (kemenangan X), -1 (kemenangan O), atau 0 (seri), sesuai dengan aturan Tic-Tac-Toe.

B. Rekursi

Fungsi ini memiliki dua mode: maximizing (maksimisasi) dan minimizing (minimisasi), yang digunakan bergantian. Pemain X selalu berusaha untuk memaksimalkan skor, sementara pemain O berusaha untuk meminimalkan skor.

ALGORITMA ALPHA - BETA

C. Iterasi

Selama mode maksimisasi, fungsi mencari langkah terbaik yang dapat diambil oleh pemain X. Selama mode minimisasi, fungsi mencari langkah terbaik yang dapat diambil oleh pemain O. Ini dilakukan dengan iterasi melalui semua kotak kosong yang tersedia pada papan permainan.

D. Pemangkasan Alpha-Beta

Pada setiap iterasi, fungsi memanggil dirinya sendiri secara rekursif dengan perubahan papan yang disalin (untuk mencoba setiap langkah), dan memperbarui nilai alpha dan beta sesuai dengan mode yang berlaku.

ALGORITMA ALPHA - BETA

- Selama mode maksimisasi, jika α lebih besar dari atau sama dengan β , maka ada pemangkasan β (break).
- Selama mode minimisasi, jika β kurang dari atau sama dengan α , maka ada pemangkasan α (break).

E. Print Statements

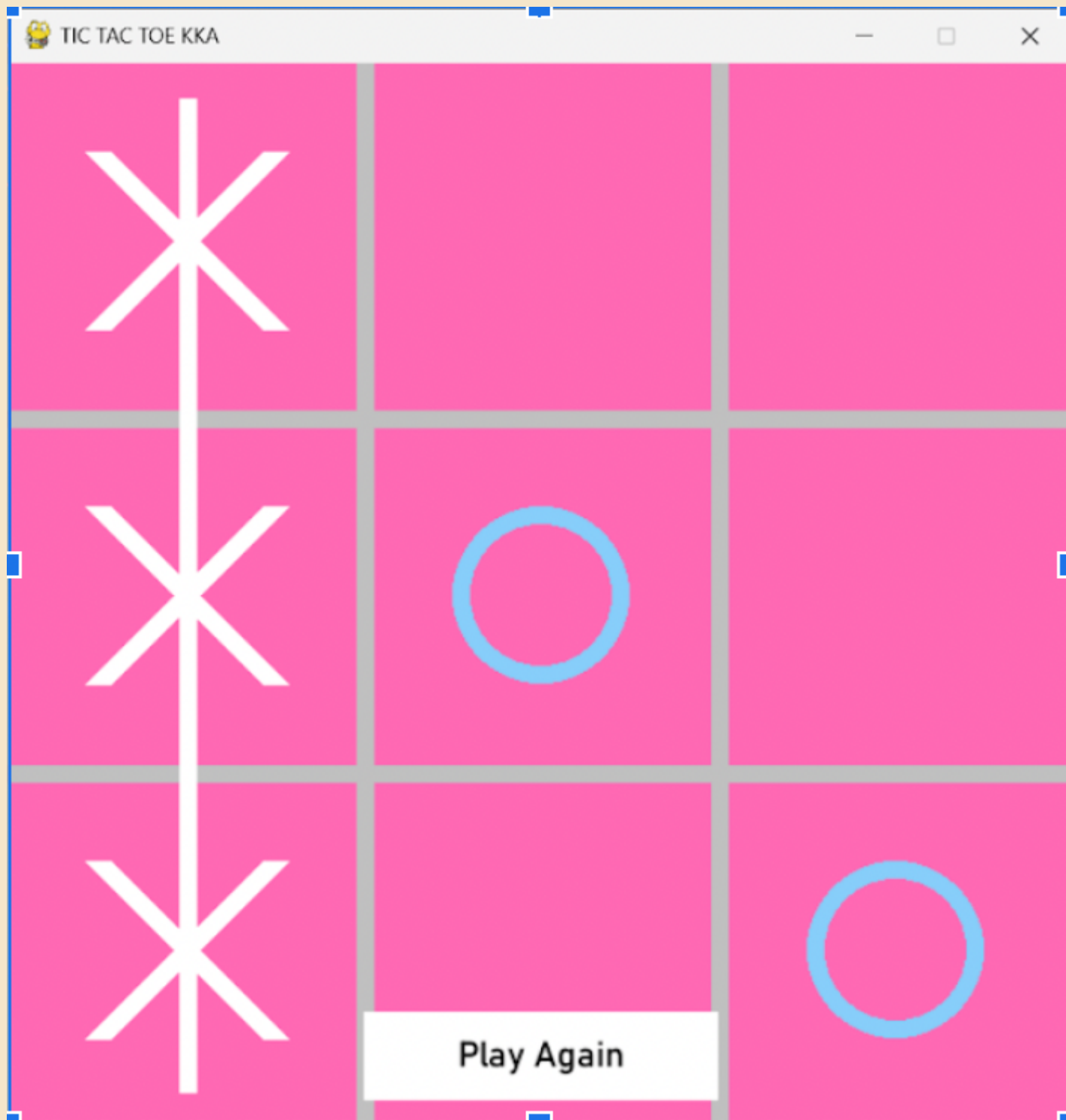
Terdapat pencetakan pernyataan yang digunakan untuk melacak pemangkasan (cut) dalam algoritma. Ini digunakan untuk pemahaman lebih lanjut tentang bagaimana algoritma bekerja.

ALGORITMA ALPHA - BETA

E. Return Values

Fungsi mengembalikan nilai skor terbaik yang dapat dicapai dalam mode yang berlaku (maksimisasi atau minimisasi) dan langkah terbaik yang harus diambil oleh pemain pada saat itu.

HASIL DARI KONDISI AI (SIMBOL X) MEMENANGKAN GAME.



HASIL ITERASI PROGRAM MENENTUKAN NILAI ALPHA DAN MELAKUKAN PEMANGKASAN ALPHA.

```
PROBLEMS OUTPUT DEBUG CONSOLE

alpha value is 0
alpha value is 1
alpha value is 1
alpha cut
alpha value is 1
alpha cut
alpha value is 1
Beta value is 1
alpha value is 0
alpha cut
alpha value is 0
alpha value is 0
alpha cut
```

HASIL TAMPILAN PROGRAM MENUNJUKKAN POSISI DAN NILAI EVAL AI.

```
alpha value is 1
alpha cut
alpha value is 1
alpha value is 1
alpha cut
alpha value is 1
alpha value is 1
alpha cut
alpha value is 1
alpha cut
alpha value is 1
AI has chosen to mark the square in pos (1, 0) with an eval of: 1
```

THANK YOU