

Instituto Tecnológico de Costa Rica

Campus Tecnológico Alajuela

Escuela de Ingeniería en Computación



Tarea 2

IC-6200 Inteligencia Artificial

Programación Racket

Profesora:

Maria Auxiliadora Mora Cross

Estudiantes:

Rodolfo Cruz Vega - 2013235955

Jonathan Quesada Salas - 2020023583

I Semestre

2023

Descripción del sistema	3
Diseño de componentes	4
Estructuras de datos utilizadas	4
Mantenimiento del estado del juego	8
Jugadores	9
Factores de relevancia	9
1. Tablero	9
2. Fichas	9
3. Jugadores	9
4. Estrategia	9
6. Movimiento	10
¿Cómo jugar Dara normalmente?	10
¿Cómo jugar Dara con el código?	10
Conclusión	14

Descripción del sistema

El juego nigeriano de Dara es un juego de mesa tradicional que se juega en varios países africanos, incluyendo Nigeria, Ghana, Senegal y Togo. También se le conoce por otros nombres, como "Draughts" o "Dame" en otros países. Aunque hay variaciones regionales en las reglas, la versión nigeriana del juego es una de las más populares y se juega con un tablero cuadrado de 12x12 casillas.

El objetivo del juego es capturar todas las piezas del oponente o bloquearlas de manera que no puedan moverse. Cada jugador tiene 48 piezas de juego, que se colocan en los cuadrados oscuros en las tres primeras filas del tablero, dejando los cuadrados claros vacíos. El jugador que controla las piezas negras comienza el juego.

Las piezas se mueven en diagonal, una casilla a la vez, en una dirección hacia el lado opuesto del tablero. Las piezas pueden capturar a las del oponente saltando sobre ellas en diagonal, siempre y cuando haya una casilla vacía detrás de la pieza capturada. Si una pieza puede capturar, debe hacerlo, y si hay varias opciones, el jugador debe elegir la opción que le permita capturar la mayor cantidad de piezas del oponente.

Si una pieza llega a la última fila del tablero, se convierte en un "Rey" y se coloca otra pieza de juego del mismo color encima de ella para indicar su nueva condición. Los Reyes pueden moverse en diagonal en cualquier dirección y pueden capturar a las piezas del oponente en cualquier dirección, siempre y cuando haya una casilla vacía detrás de la pieza capturada.

El juego termina cuando un jugador ha capturado todas las piezas del oponente o las ha bloqueado de tal manera que no pueden moverse. Si un jugador no

puede hacer un movimiento válido en su turno, pierde automáticamente el juego.

El juego de Dara es muy popular en África y se juega tanto por diversión como por competición. En algunos países, como Nigeria, hay torneos regulares de Dara con jugadores profesionales y premios en efectivo. El juego es una forma de mantener viva la tradición africana y es una actividad social muy apreciada en muchas comunidades africanas.

Diseño de componentes

El código implementa el juego Dará-Nigeriano y utiliza una representación de tablero mediante vectores anidados en Racket.

Con lo que respecta en primer lugar, se definen varias constantes que se utilizarán en todo el código, como la cantidad de filas y columnas del tablero y las identificaciones de los jugadores.

Luego, se crea la variable board, que es una matriz 5x6 que contiene el estado actual del tablero. Cada celda del tablero contiene un valor entero que indica si la celda está vacía o si está ocupada por uno de los jugadores.

Estructuras de datos utilizadas

A continuación, se definen varias funciones que se utilizan para interactuar con el tablero:

1. set-cell!: Esta función toma un tablero, una fila, una columna y un jugador y establece la celda correspondiente en el tablero con el jugador dado.
2. get-cell: Esta función toma un tablero, una fila y una columna y devuelve el valor de la celda correspondiente en el tablero.

3. `get-row`: Esta función toma un tablero y un número de fila y devuelve un vector que contiene los valores de todas las celdas de esa fila.
4. `get-col`: Esta función toma un tablero y un número de columna y devuelve un vector que contiene los valores de todas las celdas de esa columna.
5. `check-empty`: Esta función toma un tablero, una fila y una columna y devuelve un valor booleano que indica si la celda correspondiente en el tablero está vacía.
6. `place-randomly`: Esta función toma un tablero y un jugador y coloca una pieza aleatoriamente en una celda vacía del tablero.
7. `place-piece`: Esta función toma un tablero y un jugador y le pide al jugador que ingrese una fila y una columna para colocar una pieza en el tablero.
8. `remove-piece`: Esta función toma un tablero, una fila y una columna y elimina la pieza correspondiente del tablero.
9. `move-piece`: Esta función toma un tablero, una fila y una columna de origen, una fila y una columna de destino y un jugador. Mueve la pieza desde la celda de origen a la celda de destino, siempre y cuando la celda de destino esté vacía.
10. `move-piece-one-direction`: Esta función toma un tablero, una fila y una columna de origen, una fila y una columna de destino y un jugador. Esta función mueve la pieza desde la celda de origen a la celda de destino en una dirección determinada, como hacia arriba, abajo, izquierda o derecha.
11. `move-piece-up`, `move-piece-down`, `move-piece-left` y `move-piece-right`: estas son funciones auxiliares que llaman a `move-piece-one-direction` con la dirección correspondiente.

12. choose-move: Esta función toma un tablero y un jugador y le pide al jugador que elija una pieza para mover y la dirección en la que moverla.

Cada función toma como argumento el tablero y otros valores específicos para realizar su tarea. Las funciones también modifican el tablero directamente utilizando la función set-cell!.

La estructura de datos utilizada es una lista de listas (una matriz) para representar el tablero. Cada celda del tablero contiene el número del jugador (1 o 2) o 0 si está vacía.

13. Las funciones 3-in-a-row-right, 3-in-a-row-left, 3-in-a-row-center, 3-in-a-col-up, 3-in-a-col-down y 3-in-a-col-center verifican si un jugador ha colocado 3 piezas en una fila o columna. Cada una de estas funciones toma como entrada el tablero, la fila y la columna de la celda que se está verificando y el número del jugador que se está verificando. Cada una de estas funciones devuelve #t si el jugador ha colocado 3 piezas en una fila o columna y #f de lo contrario.
14. Las funciones 3-in-a-row y 3-in-a-col verifican si un jugador ha colocado 3 piezas en una fila o columna. Cada una de estas funciones toma como entrada el tablero, la fila y la columna de la celda que se está verificando y el número del jugador que se está verificando. Cada una de estas funciones devuelve #t si el jugador ha colocado 3 piezas en una fila o columna y #f de lo contrario. Estas funciones llaman a las funciones 3-in-a-row-right, 3-in-a-row-left, 3-in-a-row-center, 3-in-a-col-up, 3-in-a-col-down y 3-in-a-col-center.
15. La función 3-pieces? verifica si un jugador ha colocado 3 piezas en una fila o columna. Esta función toma como entrada el tablero, la fila y la columna de la celda que se está verificando y el número del jugador que se está verificando. Esta función devuelve #t si el jugador ha colocado 3 piezas en una fila o columna y #f de lo contrario. Esta función llama a las funciones 3-in-a-row y 3-in-a-col.

16. La función `count-pieces` cuenta el número de piezas que quedan en el tablero para un jugador dado. Esta función toma como entrada el tablero y el número del jugador que se está contando. Esta función devuelve el número de piezas que quedan en el tablero para el jugador dado.
17. La función `game-over` verifica si el juego ha terminado. Esta función toma como entrada el tablero. Devuelve `#t` si un jugador tiene solo 2 piezas restantes en el tablero y `#f` de lo contrario.
18. La función `who-won` devuelve el jugador que ha ganado el juego. Esta función toma como entrada el tablero y devuelve el número del jugador que tiene solo 2 piezas restantes en el tablero.
19. La función `next-player` devuelve el siguiente jugador. Esta función toma como entrada el número del jugador actual y devuelve el número del siguiente jugador.
20. La función `play-game-iter` es una función que implementa el bucle principal del juego. Esta función se encarga de iterar el juego, mostrar el tablero, el jugador que tiene el turno, verificar si el juego ha terminado y mostrar el ganador si es que existe.

El código es un juego de mesa en el que dos jugadores colocan fichas en un tablero y tratan de capturar las fichas del otro jugador. El código contiene varias funciones que se utilizan para jugar el juego:

21. La función `display-board-frame` toma un tablero como entrada y lo muestra en la consola. La estructura de datos del tablero es una lista de listas. Cada lista representa una fila del tablero y contiene los valores de las celdas de esa fila.
22. Las funciones `take-turns-place-piece` y `take-turns-place-piece-random` colocan fichas para ambos jugadores de manera alterna, ya sea manualmente o de manera aleatoria, respectivamente. Ambas funciones modifican el tablero

pasado como argumento. La estructura de datos del tablero es la misma que en la función `display-board-frame`.

23. Las funciones `place-pieces` y `place-pieces-random` colocan 12 fichas para cada jugador de manera alterna, ya sea manualmente o de manera aleatoria, respectivamente. Ambas funciones utilizan la función `take-turns-place-piece` o `take-turns-place-piece-random` para colocar las fichas. La estructura de datos del tablero es la misma que en la función `display-board-frame`.
24. La función `choose-placement-mode` muestra un menú para que el usuario elija si quiere colocar las fichas manualmente o de manera aleatoria. La función utiliza las funciones `place-pieces` o `place-pieces-random` para colocar las fichas según la elección del usuario.
25. La función `play-game` es la función principal que se utiliza para jugar el juego. Primero, utiliza la función `choose-placement-mode` para colocar las fichas. Luego, establece al primer jugador en el turno y llama a la función `play-game-iter` para comenzar el bucle principal del juego. La estructura de datos del tablero es la misma que en la función `display-board-frame`.

Mantenimiento del estado del juego

El mantenimiento del estado del juego Nigeriano Dará es importante para garantizar el correcto desarrollo del juego, así como para prevenir errores y asegurar que el juego se juegue de acuerdo a las reglas establecidas.

En el juego Nigeriano Dará, los estados del juego incluyen la posición de cada una de las piezas en el tablero, el jugador actual y su turno, el número de piezas que ha colocado cada jugador, el número de piezas que ha capturado cada jugador y cualquier otra información relevante para el juego.

El seguimiento de estos estados es fundamental para garantizar que el juego se desarrolle correctamente. Por ejemplo, se debe llevar un registro de las piezas capturadas por cada jugador para asegurarse de que el jugador no coloque más piezas después de haber perdido todas las suyas. Además, llevar un registro del número de piezas colocadas por cada jugador es importante para determinar cuándo se ha completado la fase de colocación de piezas y cuándo comienza la fase de movimiento.

Mantener un registro del estado del juego también puede ayudar a resolver disputas si surge alguna durante el juego. Si hay alguna confusión sobre el número de piezas que un jugador ha colocado o capturado, o si hay alguna disputa sobre la posición de una pieza en particular en el tablero, el registro del estado del juego puede ser consultado para resolver el problema de manera justa.

Jugadores

Con respecto al juego nigeriano Dara es un juego para dos jugadores. Cada jugador tiene 12 piezas, que se colocan en los puntos del tablero. Los jugadores alternan turnos para mover sus piezas en el tablero. Por lo tanto, siempre hay dos jugadores en el juego: el jugador 1 y el jugador 2.

Factores de relevancia

El juego nigeriano Dara es un juego de estrategia de dos jugadores que se juega en un tablero de juego compuesto por 12 agujeros o espacios y dos áreas de almacenamiento de semillas, una para cada jugador. A continuación, se explican los factores clave del juego:

1. Tablero

El tablero es un elemento clave en el juego nigeriano Dara, ya que es donde se colocan las semillas y se mueven durante el juego. El tablero consta de 12 agujeros o espacios, seis para cada jugador, y dos áreas de almacenamiento de semillas, una para cada jugador.

2. Fichas

Las fichas son las piezas que se mueven en el juego nigeriano Dara. En total, hay 48 semillas, 24 para cada jugador. Las semillas son generalmente pequeñas, como los granos o las piedras.

3. Jugadores

El juego nigeriano Dará se juega entre dos jugadores. Cada jugador tiene su propio conjunto de semillas y su propia área de almacenamiento de semillas en el tablero.

4. Estrategia

La estrategia es un factor clave en el juego nigeriano Dará, ya que los jugadores deben tomar decisiones estratégicas sobre cómo mover sus semillas para capturar las semillas del oponente y ganar el juego. Los jugadores también deben considerar su propia defensa y asegurarse de que sus semillas están protegidas.

5. Captura

La captura es un elemento importante del juego nigeriano Dará, ya que los jugadores pueden capturar las semillas del oponente al mover una semilla hacia un agujero con semillas del oponente. Cuando un jugador captura las semillas del oponente, las agrega a su propia área de almacenamiento de semillas en el tablero.

6. Movimiento

El movimiento de las semillas es un factor crítico en el juego nigeriano Dará. Cada jugador tiene una serie de movimientos permitidos y restricciones en función de la posición de las semillas en el tablero. Los jugadores deben tomar decisiones cuidadosas sobre cómo mover sus semillas para maximizar sus posibilidades de ganar.

¿Cómo jugar Dara normalmente?

El juego Nigeriano Dara es un juego de estrategia en el que dos jugadores compiten por capturar las piezas del otro jugador. A continuación se presentan las reglas básicas del juego:

- El juego se juega en un tablero con 12 agujeros o espacios para piezas, seis en cada lado.
- Cada jugador comienza con 12 piezas colocadas en los seis agujeros más cercanos a su lado del tablero.
- El jugador 1 comienza el juego moviendo una de sus piezas a cualquier agujero vacío adyacente en sentido contrario a las agujas del reloj. Luego, el jugador 2 realiza su turno.
- Si un jugador mueve una de sus piezas a un agujero que contiene piezas del otro jugador, captura esas piezas y las retira del tablero. El jugador también puede continuar moviendo esa misma pieza para capturar más piezas si hay una disposición adecuada en el tablero.
- Si un jugador tiene la oportunidad de capturar piezas del oponente, debe hacerlo. Si hay varias opciones de captura, el jugador puede elegir cuál de ellas tomar.
- Si un jugador no puede mover ninguna de sus piezas, pierde el turno.
- El juego continúa hasta que un jugador no tenga más piezas o no pueda hacer más movimientos. El jugador con más piezas al final del juego gana.

¿Cómo jugar Dara con el código?

Con lo que respecta a la funcionalidad que tiene el código como tal se puede ver expresado en las siguientes imágenes con su respectiva explicación del desarrollo de la ejecución:

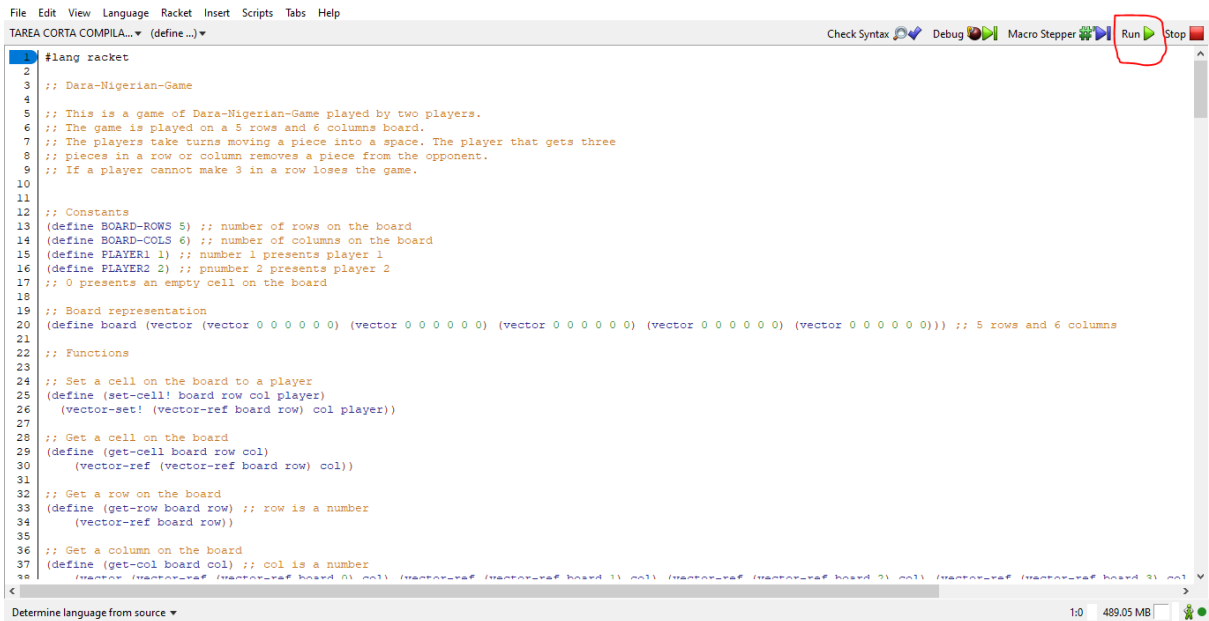


Figura 1: Iniciar el juego

En esta primera imagen se puede apreciar que el código está en un solo archivo, por lo cual la única indicación sobre dar pie a la ejecución del juego es darle click al botón superior izquierdo “Run”

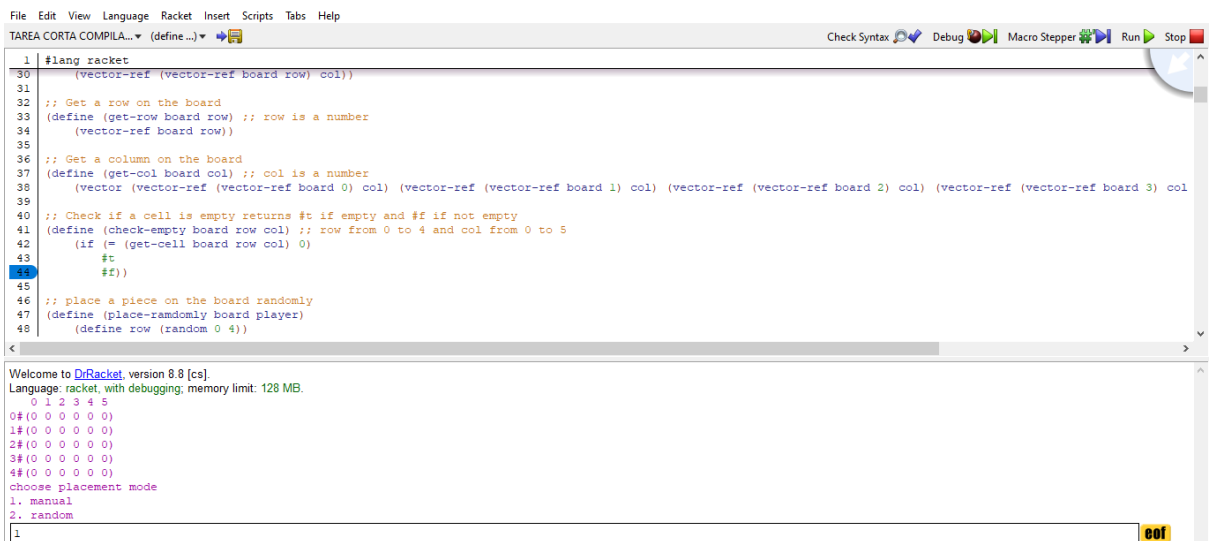


Figura 2: Cuando se ejecuta el juego

En esta segunda imagen se puede apreciar que cuando uno ejecuta el código puede escoger 2 opciones 1. manual y 2. random, en este caso escogemos la opción 1.

```

287 (play-game-iter board turn-player) ;; start game loop
288
289 (display-board-frame board)
290 (play-game board)
291
292
293 ..functions for testing
<
Welcome to DrRacket, version 8.8 [cs].
Language: racket, with debugging, memory limit: 128 MB.
0 1 2 3 4 5
0#(0 0 0 0 0 0)
1#(0 0 0 0 0 0)
2#(0 0 0 0 0 0)
3#(0 0 0 0 0 0)
4#(0 0 0 0 0 0)
choose placement mode
1. manual
2. random
1
Player 1
Enter column: 1
Enter row: 2
Player 2
Enter column: 3
Enter row: 1
0 1 2 3 4 5
0#(0 0 0 0 0 0)
1#(0 0 0 2 0 0)
2#(0 1 0 0 0 0)
3#(0 0 0 0 0 0)
4#(0 0 0 0 0 0)
Player 1
Enter column:
eof

```

Figura 3: Cuando se ejecuta la opción 1

En esta tercer imagen se puede apreciar que cuando se escoge la opción 1 se puede dar la posibilidad que entre en 2 jugadores puedan colocar sus respectivas fichas en el tablero de una manera secuencial, primero el jugador 1 y luego el jugador 2,

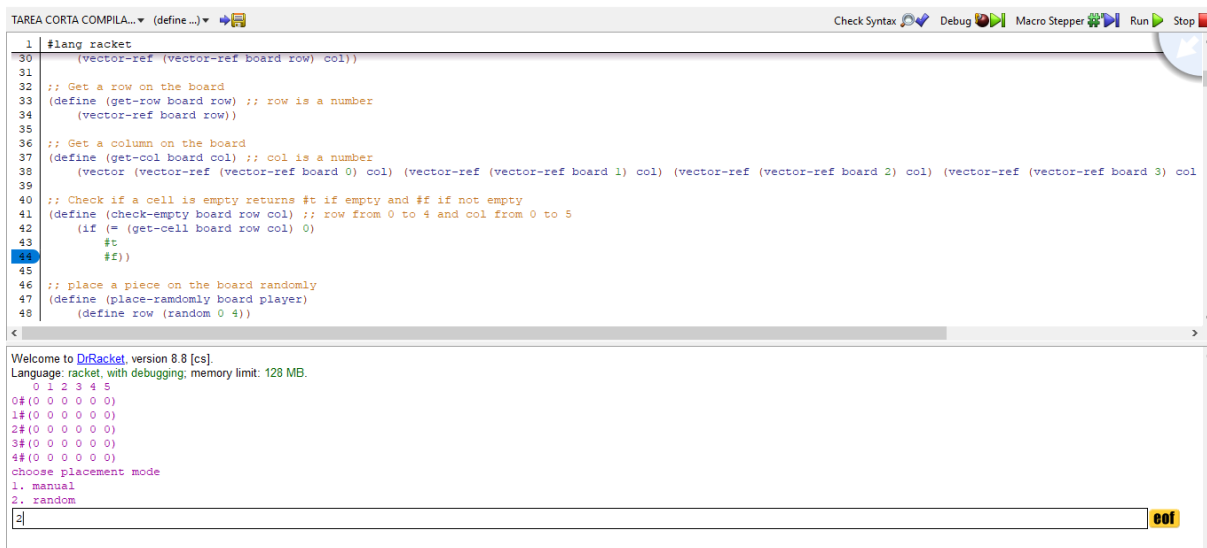
```

File Edit View Language Racket Insert Scripts Tabs Help
TAREA CORTA COMPILA... (define...) Check Syntax Debug Macro Stepper Run Stop
1 #lang racket
30 (vector-ref (vector-ref board row) col))
31
32 ;; Get a row on the board
33 (define (get-row board row) ;; row is a number
34 (vector-ref board row))
35
36 ;; Get a column on the board
37 (define (get-col board col) ;; col is a number
38 (vector (vector-ref (vector-ref board 0) col) (vector-ref (vector-ref board 1) col) (vector-ref (vector-ref board 2) col) (vector-ref (vector-ref board 3) col)
39
40 ;; Check if a cell is empty returns #t if empty and #f if not empty
41 (define (check-empty board row col) ;; row from 0 to 4 and col from 0 to 5
42 (if (= (get-cell board row col) 0)
43 #t
44 #f))
45
46 ;; place a piece on the board randomly
47 (define (place-randomly board player)
<
3. Move piece left
4. Move piece right
0
Invalid move
0 1 2 3 4 5
0#(1 0 0 0 0 0)
1#(2 1 0 2 0 0)
2#(0 1 0 0 0 0)
3#(0 2 0 0 0 0)
4#(0 0 0 0 0 0)
2
Choose piece to move for player 2
Enter row 0-4: 4
Enter col 0-5: 4
Enter move: 1. Move piece up
2. Move piece down
3. Move piece left

```

Figura 4: Opciones

En esta cuarta imagen es en el caso que se quiera mover la dicha pieza en un lugar de arriba, izquierda y abajo.

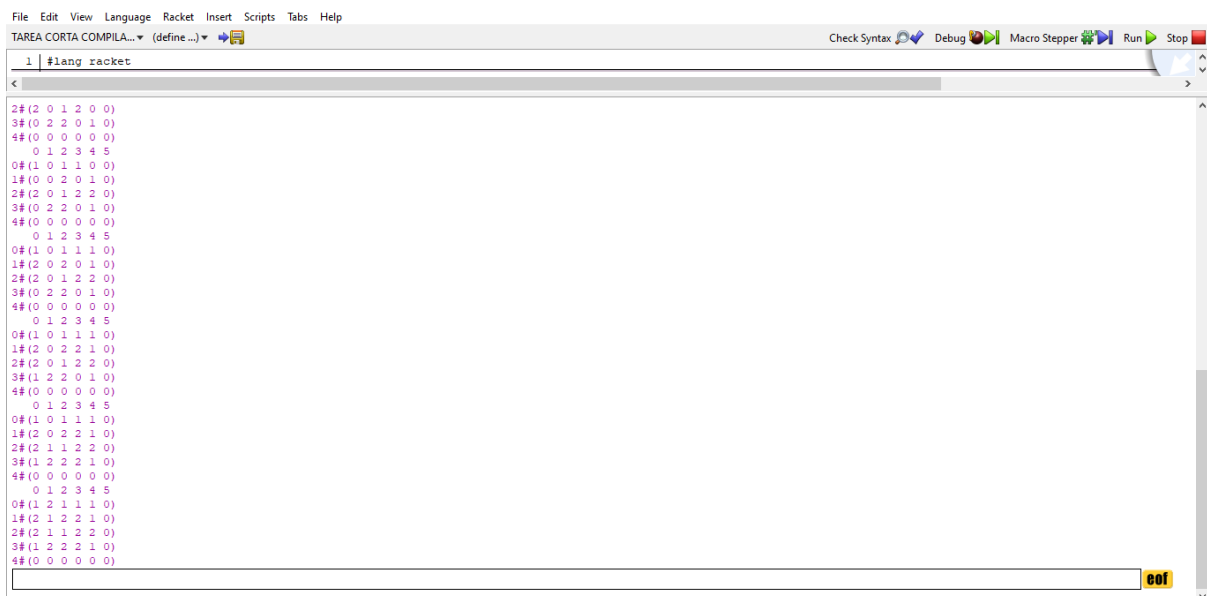


```
1 #lang racket
30 (vector-ref (vector-ref board row) col))
31
32 ;; Get a row on the board
33 (define (get-row board row) ;; row is a number
34   (vector-ref board row))
35
36 ;; Get a column on the board
37 (define (get-col board col) ;; col is a number
38   (vector (vector-ref (vector-ref board 0) col) (vector-ref (vector-ref board 1) col) (vector-ref (vector-ref board 2) col) (vector-ref (vector-ref board 3) col)
39
40 ;; Check if a cell is empty returns #t if empty and #f if not empty
41 (define (check-empty board row col) ;; row from 0 to 4 and col from 0 to 5
42   (if (= (get-cell board row col) 0)
43       #t
44       #f))
45
46 ;; place a piece on the board randomly
47 (define (place-randomly board player)
48   (define row (random 0 4)))
```

Welcome to DrRacket, version 8.8 [cs].
Language: racket, with debugging; memory limit: 128 MB.
0 1 2 3 4 5
0#(0 0 0 0 0 0)
1#(0 0 0 0 0 0)
2#(0 0 0 0 0 0)
3#(0 0 0 0 0 0)
4#(0 0 0 0 0 0)
choose placement mode
1. manual
2. random
2]

Figura 5: Elección de la opción 2

En esta quinta imagen se puede apreciar la ejecución del código, pero con la diferencia que al inicio de la ejecución se escoge la opción 2 la cuales “random” la cual se hará la colocación de las fichas de una manera aleatoria para evitar ponerlas todas y hacerlo más rápido (Fue un toque personal del grupo)



```
2#(2 0 1 2 0 0)
3#(0 2 2 0 1 0)
4#(0 0 0 0 0 0)
0 1 2 3 4 5
0#(1 0 1 1 0 0)
1#(0 0 2 0 1 0)
2#(2 0 1 2 2 0)
3#(0 2 2 0 1 0)
4#(0 0 0 0 0 0)
0 1 2 3 4 5
0#(1 0 1 1 1 0)
1#(2 0 2 0 1 0)
2#(2 0 1 2 2 0)
3#(0 2 2 0 1 0)
4#(0 0 0 0 0 0)
0 1 2 3 4 5
0#(1 0 1 1 1 0)
1#(2 0 2 2 1 0)
2#(2 0 1 2 2 0)
3#(1 2 2 0 1 0)
4#(0 0 0 0 0 0)
0 1 2 3 4 5
0#(1 0 1 1 1 0)
1#(2 0 2 2 1 0)
2#(2 1 1 2 2 0)
3#(1 2 2 2 1 0)
4#(0 0 0 0 0 0)
0 1 2 3 4 5
0#(1 2 1 1 1 0)
1#(2 1 2 2 1 0)
2#(2 1 1 2 2 0)
3#(1 2 2 2 1 0)
4#(0 0 0 0 0 0)
```

Figura 6: Salida de la opción 2

En esta sexta imagen se puede apreciar que cuando se ejecuta la opción 2 del programa se genera de una manera automática todo lo referente a la colocación de las fichas.

Conclusión

En conclusión, el juego Nigeriano Dara es un juego de estrategia fascinante y desafiante que se juega en África y otras partes del mundo. Con un tablero y semillas como únicos requisitos, este juego puede ser jugado por personas de todas las edades y habilidades, y proporciona horas de diversión y entretenimiento.

Dara es un juego que promueve el pensamiento crítico, la estrategia y la planificación a largo plazo, así como la habilidad para leer el movimiento del oponente y anticipar sus movimientos futuros. Además, es un juego que se puede jugar individualmente o en parejas, lo que lo hace adecuado para diferentes situaciones y preferencias.

Aunque puede parecer simple al principio, el juego Dara tiene una gran cantidad de estrategias y movimientos posibles que pueden ser explorados y perfeccionados con la práctica. Por lo tanto, es un juego que puede mantener a los jugadores interesados y desafiados durante mucho tiempo.

En resumen, el juego Nigeriano Dara es un juego emocionante y desafiante que tiene una larga historia y tradición en África. Si estás buscando un juego que desafíe tus habilidades estratégicas y te proporcione horas de diversión, entonces Dara es definitivamente una excelente opción.