

Hundir la flota

Representación gráfica en Racket

Carlos Lucena Robles

Curso 4º de Grado de Ingeniería Informática
Programación Declarativa. 1º Cuatrimestre
Escuela Politécnica Superior
Universidad de Córdoba

Curso académico 2023 - 2024
Córdoba, 8 de enero de 2024



Introducción

- 1 Introducción
- 2 Descripción
- 3 Descripción modular del código
- 4 Resultados
- 5 Conclusiones
- 6 Referencias

Introducción

Descripción del juego

- 1 Introducción
 - Descripción del juego
 - Historia

Introducción

Descripción del juego

Hundir la flota. *Battleship*

- Juego tradicional de guerra de estrategia (y suerte) por turnos.
- Los dos jugadores colocan su flota en su tablero de forma secreta al contrincante.
- Cada jugador en su turno intenta hundir los barcos del contrincante.
- El objetivo es *Hundir la flota* del contrincante.

Introducción

Descripción del juego

Hundir la flota. *Battleship*

- Juego tradicional de guerra de estrategia (y suerte) por turnos.
- Los dos jugadores colocan su flota en su tablero de forma secreta al contrincante.
- Cada jugador en su turno intenta hundir los barcos del contrincante.
- El objetivo es *Hundir la flota* del contrincante.

Introducción

Descripción del juego

Hundir la flota. *Battleship*

- Juego tradicional de guerra de estrategia (y suerte) por turnos.
- Los dos jugadores colocan su flota en su tablero de forma secreta al contrincante.
- Cada jugador en su turno intenta hundir los barcos del contrincante.
- El objetivo es *Hundir la flota* del contrincante.

Introducción

Descripción del juego

Hundir la flota. *Battleship*

- Juego tradicional de guerra de estrategia (y suerte) por turnos.
- Los dos jugadores colocan su flota en su tablero de forma secreta al contrincante.
- Cada jugador en su turno intenta hundir los barcos del contrincante.
- El objetivo es *Hundir la flota* del contrincante.



Figura 1: Juego de mesa de hundir la flota. Año: 1980

Introducción

Historia

- 1 Introducción
 - Descripción del juego
 - Historia

Introducción

Historia

Orígenes

- Su posible origen está en el juego francés *L'Attaque* (publ. Hermance Edan, 1909)
- La primera versión comercializada, *Salvo*, fue en 1931 por *Starex Company*.
- En 1967, *Milton Bradley Company*, introduce una versión con tableros, barcos y otros elementos de plástico.

Introducción

Historia

Orígenes

- Su posible origen está en el juego francés *L'Attaque* (publ. Hermance Edan, 1909)
- La primera versión comercializada, *Salvo*, fue en 1931 por *Starex Company*.
- En 1967, *Milton Bradley Company*, introduce una versión con tableros, barcos y otros elementos de plástico.

Introducción

Historia

Orígenes

- Su posible origen está en el juego francés *L'Attaque* (publ. Hermance Edan, 1909)
- La primera versión comercializada, *Salvo*, fue en 1931 por *Starex Company*.
- En 1967, *Milton Bradley Company*, introduce una versión con tableros, barcos y otros elementos de plástico.



Figura 2: Juego de mesa *L'Attaque*. Año: 1909



Figura 3: Juego de mesa *Salvo*. Año: 1931



Figura 4: Juego de mesa *Battleship*. Año: 1967

Introducción

Historia

Evolución

- En 1977, Milton lanza la primera versión computarizada: *Electronic Battleship*.
- En 1984, **Hasbro** adquiere Milton Bradley Company, y por tanto los derechos de *Battleship*
- La **versión de Milton** es la que se ha expandido globalmente y la más conocida hasta hoy.
- Se han creado múltiples variantes, desde juegos de mesa hasta digitales.

Introducción

Historia

Evolución

- En 1977, Milton lanza la primera versión computarizada: *Electronic Battleship*.
- En 1984, **Hasbro** adquiere Milton Bradley Company, y por tanto los derechos de *Battleship*
- La *versión de Milton* es la que se ha expandido globalmente y la más conocida hasta hoy.
- Se han creado múltiples variantes, desde juegos de mesa hasta digitales.

Introducción

Historia

Evolución

- En 1977, Milton lanza la primera versión computarizada: *Electronic Battleship*.
- En 1984, **Hasbro** adquiere Milton Bradley Company, y por tanto los derechos de *Battleship*
- La **versión de Milton** es la que se ha expandido globalmente y la más conocida hasta hoy.
- Se han creado múltiples variantes, desde juegos de mesa hasta digitales.

Introducción

Historia

Evolución

- En 1977, Milton lanza la primera versión computarizada: *Electronic Battleship*.
- En 1984, **Hasbro** adquiere Milton Bradley Company, y por tanto los derechos de *Battleship*
- La **versión de Milton** es la que se ha expandido globalmente y la más conocida hasta hoy.
- Se han creado múltiples variantes, desde juegos de mesa hasta digitales.



Figura 5: Primera versión computarizada de *Battleship*. Año: 1977



Figura 6: Versión actual de *Battleship* de Hasbro.

Descripción

- 1 Introducción
- 2 Descripción**
- 3 Descripción modular del código
- 4 Resultados
- 5 Conclusiones
- 6 Referencias

Descripción

Tableros

2 Descripción

- Tableros
- Flota y restricciones en la colocación
- Condición de victoria
- Contrincante

Descripción

Tableros

Tableros

- Tamaño 10x10 casillas
- Columnas identificadas con letras de la A - J
- Filas numeradas del 1 al 10
- Hay dos tableros:
 - **Tablero del jugador:** Donde el jugador coloca sus barcos y se registran los disparos de la máquina.
 - **Tablero del contrincante:** Donde la máquina coloca sus barcos y donde el jugador realiza sus disparos.

Descripción

Tableros

Tableros

- Tamaño 10x10 casillas
- Columnas identificadas con letras de la A - J
- Filas numeradas del 1 al 10
- Hay dos tableros:
 - **Tablero del jugador:** Donde el jugador coloca sus barcos y se registran los disparos de la máquina.
 - **Tablero del contrincante:** Donde la máquina coloca sus barcos y donde el jugador realiza sus disparos.

Descripción

Tableros

Tableros

- Tamaño 10x10 casillas
- Columnas identificadas con letras de la A - J
- Filas numeradas del 1 al 10
- Hay dos tableros:
 - **Tablero del jugador:** Donde el jugador coloca sus barcos y se registran los disparos de la máquina.
 - **Tablero del contrincante:** Donde la máquina coloca sus barcos y donde el jugador realiza sus disparos.

Descripción

Tableros

Tableros

- Tamaño 10x10 casillas
- Columnas identificadas con letras de la A - J
- Filas numeradas del 1 al 10
- Hay **dos tableros**:
 - **Tablero del jugador**: Donde el jugador coloca sus barcos y se registran los disparos de la máquina.
 - **Tablero del contrincante**: Donde la máquina coloca sus barcos y donde el jugador realiza sus disparos.

Descripción

Tableros

Tableros

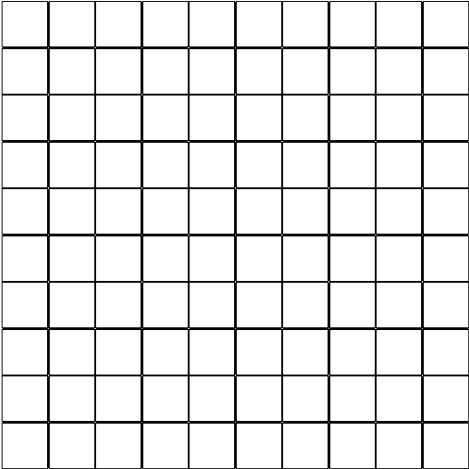
- Tamaño 10x10 casillas
- Columnas identificadas con letras de la A - J
- Filas numeradas del 1 al 10
- Hay **dos tableros**:
 - **Tablero del jugador**: Donde el jugador coloca sus barcos y se registran los disparos de la máquina.
 - **Tablero del contrincante**: Donde la máquina coloca sus barcos y donde el jugador realiza sus disparos.

Descripción

Tableros

Tableros

- Tamaño 10x10 casillas
- Columnas identificadas con letras de la A - J
- Filas numeradas del 1 al 10
- Hay **dos tableros**:
 - **Tablero del jugador**: Donde el jugador coloca sus barcos y se registran los disparos de la máquina.
 - **Tablero del contrincante**: Donde la máquina coloca sus barcos y donde el jugador realiza sus disparos.



	A	B	C	D	E	F	G	H	I	J
1										
2										
3										
4										
5										
6										
7										
8										
9										
10										

Figura 7: Representación gráfica de un tablero

Descripción

Flota y restricciones en la colocación

2 Descripción

- Tableros
- Flota y restricciones en la colocación
- Condición de victoria
- Contrincante

Flota y restricciones en la colocación

- 1 x Portaaviones de 5 casillas
- 1 x Acorazado de 4 casillas
- 2 x Crucero de 3 casillas
- 3 x Destructor de 2 casillas
- 3 x Submarino de 1 casilla



Flota y restricciones en la colocación

- 1 x Portaaviones de 5 casillas
- 1 x Acorazado de 4 casillas
- 2 x Crucero de 3 casillas
- 3 x Destructor de 2 casillas
- 3 x Submarino de 1 casilla



Flota y restricciones en la colocación

- 1 x Portaaviones de 5 casillas
- 1 x Acorazado de 4 casillas
- 2 x Crucero de 3 casillas
- 3 x Destructor de 2 casillas
- 3 x Submarino de 1 casilla



Descripción

Flota y restricciones en la colocación

Flota

- 1 x Portaaviones de 5 casillas
- 1 x Acorazado de 4 casillas
- 2 x Crucero de 3 casillas
- 3 x Destructor de 2 casillas
- 3 x Submarino de 1 casilla



Figura 8: Representación de la flota

Flota y restricciones en la colocación

- 1 x Portaaviones de 5 casillas
- 1 x Acorazado de 4 casillas
- 2 x Crucero de 3 casillas
- 3 x Destructor de 2 casillas
- 3 x Submarino de 1 casilla



Restricciones en la colocación de los barcos

- Se pueden colocar vertical u horizontalmente.
- No se permiten casillas con barco **contiguas** (ni diagonalmente).

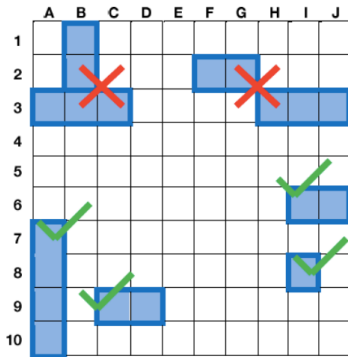


Figura 9:
Restricciones en la
colocación

Restricciones en la colocación de los barcos

- Se pueden colocar vertical u horizontalmente.
- **No** se permiten casillas con barco **contiguas** (ni diagonalmente).

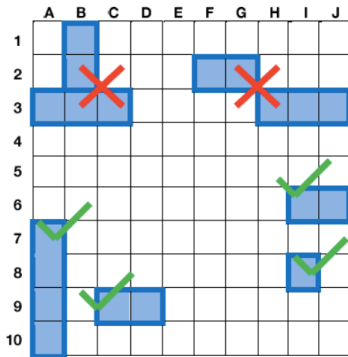


Figura 9:
Restricciones en la
colocación

Descripción

Condición de victoria

2 Descripción

- Tableros
- Flota y restricciones en la colocación
- Condición de victoria
- Contrincante

Descripción

Condición de victoria

Condición de victoria

Gana el primer jugador que **hunda la flota** del contrincante.

Descripción

Contrincante

2 Descripción

- Tableros
- Flota y restricciones en la colocación
- Condición de victoria
- **Contrincante**
 - Algoritmo aleatorio
 - Algoritmo Hunt/Target
 - Algoritmo Hunt/Target probabilístico

Descripción

Contrincante

Contrincante

- El único oponente será la **CPU** (máquina).
- Habrá tres tipos de algoritmos diferentes que podrá usar:
 - Algoritmo aleatorio
 - Algoritmo Hunt/Target
 - Algoritmo Hunt/Target probabilístico

Descripción

Contrincante

Contrincante

- El único oponente será la CPU (máquina).
- Habrá tres tipos de algoritmos diferentes que podrá usar:
 - Algoritmo aleatorio
 - Algoritmo Hunt/Target
 - Algoritmo Hunt/Target probabilístico

Descripción

Contrincante

Contrincante

- El único oponente será la CPU (máquina).
- Habrá tres tipos de algoritmos diferentes que podrá usar:
 - **Algoritmo aleatorio**
 - Algoritmo Hunt/Target
 - Algoritmo Hunt/Target probabilístico

Descripción

Contrincante

Contrincante

- El único oponente será la CPU (máquina).
- Habrá tres tipos de algoritmos diferentes que podrá usar:
 - **Algoritmo aleatorio**
 - **Algoritmo Hunt/Target**
 - Algoritmo Hunt/Target probabilístico

Descripción

Contrincante

Contrincante

- El único oponente será la **CPU** (máquina).
- Habrá tres tipos de algoritmos diferentes que podrá usar:
 - **Algoritmo aleatorio**
 - **Algoritmo Hunt/Target**
 - **Algoritmo Hunt/Target probabilístico**

Descripción

Contrincante

2 Descripción

- Tableros
- Flota y restricciones en la colocación
- Condición de victoria
- **Contrincante**
 - Algoritmo aleatorio
 - Algoritmo Hunt/Target
 - Algoritmo Hunt/Target probabilístico

Descripción

Contrincante

Algoritmo aleatorio

- Realiza el disparo escogiendo una fila y una columna de forma aleatoria.
- No escogerá casillas **disparadas con anterioridad** o **casillas contiguas** a barcos hundidos.

Debilidades

- Cuando un barco es alcanzado, el algoritmo no termina de hundirlo.
- Recorre la mayoría de las casillas del tablero (muy ineficiente).

Descripción

Contrincante

Algoritmo aleatorio

- Realiza el disparo escogiendo una fila y una columna de forma aleatoria.
- No escogerá casillas **disparadas con anterioridad** o **casillas contiguas** a barcos hundidos.

Debilidades

- Cuando un barco es alcanzado, el algoritmo no termina de hundirlo.
- Recorre la mayoría de las casillas del tablero (muy ineficiente).

Descripción

Contrincante

Algoritmo aleatorio

- Realiza el disparo escogiendo una fila y una columna de forma aleatoria.
- No escogerá casillas **disparadas con anterioridad** o **casillas contiguas** a barcos hundidos.

Debilidades

- Cuando un barco es alcanzado, el algoritmo no termina de hundirlo.
- Recorre la mayoría de las casillas del tablero (muy ineficiente).

Descripción

Contrincante

Algoritmo aleatorio

- Realiza el disparo escogiendo una fila y una columna de forma aleatoria.
- No escogerá casillas **disparadas con anterioridad** o **casillas contiguas** a barcos hundidos.

Debilidades

- Cuando un barco es alcanzado, el algoritmo no termina de hundirlo.
- Recorre la mayoría de las casillas del tablero (muy ineficiente).

Descripción

Contrincante

2 Descripción

- Tableros
- Flota y restricciones en la colocación
- Condición de victoria
- **Contrincante**
 - Algoritmo aleatorio
 - **Algoritmo Hunt/Target**
 - Algoritmo Hunt/Target probabilístico

Descripción

Contrincante

Algoritmo Hunt/Target

- Usa **dos modos** (*funciones*) alternos que usan una lista de objetivos conjuntamente para determinar la posición de un barco y hundirlo:
 - **Modo hunt** para buscar un objetivo
 - **Modo target** para hundir el barco

Descripción

Contrincante

Algoritmo Hunt/Target

- Usa **dos modos** (*funciones*) alternos que usan una lista de objetivos conjuntamente para determinar la posición de un barco y hundirlo:
 - **Modo hunt** para buscar un objetivo
 - **Modo target** para hundir el barco

Descripción

Contrincante

Algoritmo Hunt/Target

- Usa **dos modos** (*funciones*) alternos que usan una lista de objetivos conjuntamente para determinar la posición de un barco y hundirlo:
 - **Modo hunt** para buscar un objetivo
 - **Modo target** para hundir el barco

Descripción

Contrincante

Modo Hunt

- Dispara aleatoriamente una de las casillas del tablero que cumplan:
 - **no** disparadas con anterioridad ni contiguas a barcos hundidos.
 - **pared** **par** y **espacio** disponible para el barco de menor longitud sin hundir.
- Si el disparo es sobre un **barco**, se crea una lista con sus casillas vecinas N - S - E - O

Descripción

Contrincante

Modo Hunt

- Dispara aleatoriamente una de las casillas del tablero que cumplan:
 - **no** disparadas con anterioridad ni contiguas a barcos hundidos.
 - **pared** **par** y **espacio** disponible para el barco de menor longitud sin hundir.
- Si el disparo es sobre un **barco**, se crea una lista con sus casillas vecinas N - S - E - O

Descripción

Contrincante

Modo Hunt

- Dispara aleatoriamente una de las casillas del tablero que cumplan:
 - **no** disparadas con anterioridad ni contiguas a barcos hundidos.
 - **pared** **par** y **espacio** disponible para el barco de menor longitud sin hundir.
- Si el disparo es sobre un **barco**, se crea una lista con sus casillas vecinas N - S - E - O

Descripción

Contrincante

Modo Hunt

- Dispara aleatoriamente una de las casillas del tablero que cumplan:
 - **no** disparadas con anterioridad ni contiguas a barcos hundidos.
 - **pared** **par** y **espacio** disponible para el barco de menor longitud sin hundir.
- Si el disparo es sobre un **barco**, se crea una **lista con sus casillas vecinas N - S - E - O**

Pariedad par

Que casillas (mínimas) debe contener parte de un barco colocado de cualquier forma en el tablero.

Nota

Se aplica para el **barco de menor longitud sin hundir** para garantizar tocar a todos los barcos en menos disparos.

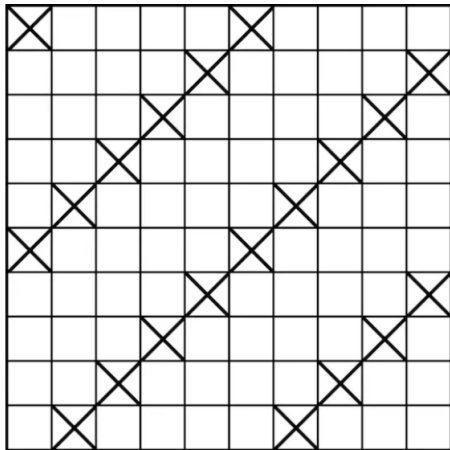


Figura 10: Patrón de disparo para un barco de 5 casillas

Espacio disponible

La casilla seleccionada debe contener al menos **una dirección con suficientes casillas vacías** para albergar el barco de menor longitud sin hundir.

Descripción

Contrincante

Modo target

- Dispara a las casillas de la **lista objetivos** para precisar la **orientación** del barco:
 - Si el disparo es en la misma fila: **horizontal**.
 - Si el disparo es en la misma columna: **navyblue**.
- Una vez precisada la orientación, por cada acierto, se añade a la lista de objetivos la **siguiente casilla** de la misma fila (horizontal) o de la misma columna (vertical).
- Una vez que el barco es **hundido**, se vuelve al modo hunt para repetir el proceso.

Descripción

Contrincante

Modo target

- Dispara a las casillas de la **lista objetivos** para precisar la **orientación** del barco:
 - Si el disparo es en la misma fila: **horizontal**.
 - Si el disparo es en la misma columna: **navyblue**.
- Una vez precisada la orientación, por cada acierto, se añade a la lista de objetivos la **siguiente casilla** de la misma fila (horizontal) o de la misma columna (vertical).
- Una vez que el barco es **hundido**, se vuelve al modo hunt para repetir el proceso.

Descripción

Contrincante

Modo target

- Dispara a las casillas de la **lista objetivos** para precisar la **orientación** del barco:
 - Si el disparo es en la misma fila: **horizontal**.
 - Si el disparo es en la misma columna: **navyblue**.
- Una vez precisada la orientación, por cada acierto, se añade a la lista de objetivos la **siguiente casilla** de la misma fila (horizontal) o de la misma columna (vertical).
- Una vez que el barco es **hundido**, se vuelve al modo hunt para repetir el proceso.

Descripción

Contrincante

Modo target

- Dispara a las casillas de la **lista objetivos** para precisar la **orientación** del barco:
 - Si el disparo es en la misma fila: **horizontal**.
 - Si el disparo es en la misma columna: **navyblue**.
- Una vez precisada la orientación, por cada acierto, se añade a la lista de objetivos la **siguiente casilla** de la misma fila (horizontal) o de la misma columna (vertical).
- Una vez que el barco es **hundido**, se vuelve al modo hunt para repetir el proceso.

Descripción

Contrincante

Modo target

- Dispara a las casillas de la **lista objetivos** para precisar la **orientación** del barco:
 - Si el disparo es en la misma fila: **horizontal**.
 - Si el disparo es en la misma columna: **navyblue**.
- Una vez precisada la orientación, por cada acierto, se añade a la lista de objetivos la **siguiente casilla** de la misma fila (horizontal) o de la misma columna (vertical).
- Una vez que el barco es **hundido**, se vuelve al modo hunt para repetir el proceso.

Notas sobre la *lista de objetivos*

- Hay que *barajarla* al principio para que los barcos verticales no tengan desventaja.
- Es necesaria para *guardar los objetivos* después de un cambio de turno, o retroceder si el disparo no ha sido en los extremos
- *Solo* contiene casillas no disparadas ni contiguas a barcos hundidos y dentro de los límites del tablero.

Notas sobre la lista de objetivos

- Hay que *barajarla* al principio para que los barcos verticales no tengan desventaja.
- Es necesaria para **guardar los objetivos** después de un cambio de turno, o retroceder si el disparo no ha sido en los extremos
- **Solo** contiene casillas no disparadas ni contiguas a barcos hundidos y dentro de los límites del tablero.

Notas sobre la lista de objetivos

- Hay que *barajarla* al principio para que los barcos verticales no tengan desventaja.
- Es necesaria para **guardar los objetivos** después de un cambio de turno, o retroceder si el disparo no ha sido en los extremos
- **Solo** contiene casillas no disparadas ni contiguas a barcos hundidos y dentro de los límites del tablero.

Fortalezas

- Cuando un barco es alcanzado es hundido.
- Se realizan menos disparos (es más eficiente).
- Durante la evolución de la partida, el algoritmo es más rápido en hundir barcos.

Debilidades

- Todavía sigue usando aleatoriedad.
- Necesita una lista de objetivos.
- Al existir el barco de una casilla, en la mayoría de partidas el modo hunt no se beneficia de sus filtros.

Fortalezas

- Cuando un barco es alcanzado es hundido.
- Se realizan menos disparos (es más eficiente).
- Durante la evolución de la partida, el algoritmo es más rápido en hundir barcos.

Debilidades

- Todavía sigue usando aleatoriedad.
- Necesita una lista de objetivos.
- Al existir el barco de una casilla, en la mayoría de partidas el modo hunt no se beneficia de sus filtros.

Fortalezas

- Cuando un barco es alcanzado es hundido.
- Se realizan menos disparos (es más eficiente).
- Durante la evolución de la partida, el algoritmo es más rápido en hundir barcos.

Debilidades

- Todavía sigue usando aleatoriedad.
- Necesita una lista de objetivos.
- Al existir el barco de una casilla, en la mayoría de partidas el modo hunt no se beneficia de sus filtros.

Fortalezas

- Cuando un barco es alcanzado es hundido.
- Se realizan menos disparos (es más eficiente).
- Durante la evolución de la partida, el algoritmo es más rápido en hundir barcos.

Debilidades

- Todavía sigue usando aleatoriedad.
- Necesita una lista de objetivos.
- Al existir el barco de una casilla, en la mayoría de partidas el modo hunt no se beneficia de sus filtros.

Fortalezas

- Cuando un barco es alcanzado es hundido.
- Se realizan menos disparos (es más eficiente).
- Durante la evolución de la partida, el algoritmo es más rápido en hundir barcos.

Debilidades

- Todavía sigue usando aleatoriedad.
- Necesita una lista de objetivos.
- Al existir el barco de una casilla, en la mayoría de partidas el modo hunt no se beneficia de sus filtros.

Descripción

Contrincante

2 Descripción

- Tableros
- Flota y restricciones en la colocación
- Condición de victoria
- **Contrincante**
 - Algoritmo aleatorio
 - Algoritmo Hunt/Target
 - **Algoritmo Hunt/Target probabilístico**

Descripción

Contrincante

Algoritmo Hunt/Target probabilístico

- Usa la **misma idea** de los dos *modos* que el algoritmo Hunt/Target normal.
- **Mejora** el modo hunt y modo target para eliminar la *aleatoriedad*.
- Se mantiene el uso de la lista de objetivos
- Se usan las **funciones de densidad de probabilidad** para tomar las decisiones.

Nota

Se puede usar una versión de este algoritmo sin lista de objetivos. Todas las decisiones serían tomadas con probabilidad.

Descripción

Contrincante

Algoritmo Hunt/Target probabilístico

- Usa la **misma idea** de los dos *modos* que el algoritmo Hunt/Target normal.
- **Mejora** el modo hunt y modo target para eliminar la *aleatoriedad*.
- Se mantiene el uso de la lista de objetivos
- Se usan las **funciones de densidad de probabilidad** para tomar las decisiones.

Nota

Se puede usar una versión de este algoritmo sin lista de objetivos. Todas las decisiones serían tomadas con probabilidad.

Descripción

Contrincante

Algoritmo Hunt/Target probabilístico

- Usa la **misma idea** de los dos *modos* que el algoritmo Hunt/Target normal.
- **Mejora** el modo hunt y modo target para eliminar la *aleatoriedad*.
- Se mantiene el uso de la lista de objetivos
- Se usan las **funciones de densidad de probabilidad** para tomar las decisiones.

Nota

Se puede usar una versión de este algoritmo sin lista de objetivos. Todas las decisiones serían tomadas con probabilidad.

Descripción

Contrincante

Algoritmo Hunt/Target probabilístico

- Usa la **misma idea** de los dos *modos* que el algoritmo Hunt/Target normal.
- **Mejora** el modo hunt y modo target para eliminar la *aleatoriedad*.
- Se mantiene el uso de la lista de objetivos
- Se usan las **funciones de densidad de probabilidad** para tomar las decisiones.

Nota

Se puede usar una versión de este algoritmo sin lista de objetivos. Todas las decisiones serían tomadas con probabilidad.

Descripción

Contrincante

Algoritmo Hunt/Target probabilístico

- Usa la **misma idea** de los dos *modos* que el algoritmo Hunt/Target normal.
- **Mejora** el modo hunt y modo target para eliminar la *aleatoriedad*.
- Se mantiene el uso de la lista de objetivos
- Se usan las **funciones de densidad de probabilidad** para tomar las decisiones.

Nota

Se puede usar una versión de este algoritmo sin lista de objetivos. Todas las decisiones serían tomadas con probabilidad.

Descripción

Contrincante

Funciones de densidad de probabilidad

- Se considera que en el centro del tablero habrá más **densidad de barcos**.
- Por cada casilla que haya un barco se asigna un **peso**.
- Se calculan todas las posibles posiciones en todas las casillas con todos los barcos (*superposición*)
- Al sumarse los pesos por cada posible variación en cada casilla, se obtiene una **matriz de probabilidades**.

Descripción

Contrincante

Funciones de densidad de probabilidad

- Se considera que en el centro del tablero habrá más **densidad de barcos**.
- Por cada casilla que haya un barco se asigna un **peso**.
- Se calculan todas las posibles posiciones en todas las casillas con todos los barcos (*superposición*)
- Al sumarse los pesos por cada posible variación en cada casilla, se obtiene una **matriz de probabilidades**.

Descripción

Contrincante

Funciones de densidad de probabilidad

- Se considera que en el centro del tablero habrá más **densidad de barcos**.
- Por cada casilla que haya un barco se asigna un **peso**.
- Se calculan todas las posibles posiciones en todas las casillas con todos los barcos (*superposición*)
- Al sumarse los pesos por cada posible variación en cada casilla, se obtiene una **matriz de probabilidades**.

Descripción

Contrincante

Funciones de densidad de probabilidad

- Se considera que en el centro del tablero habrá más **densidad de barcos**.
- Por cada casilla que haya un barco se asigna un **peso**.
- Se calculan todas las posibles posiciones en todas las casillas con todos los barcos (*superposición*)
- Al sumarse los pesos por cada posible variación en cada casilla, se obtiene una **matriz de probabilidades**.

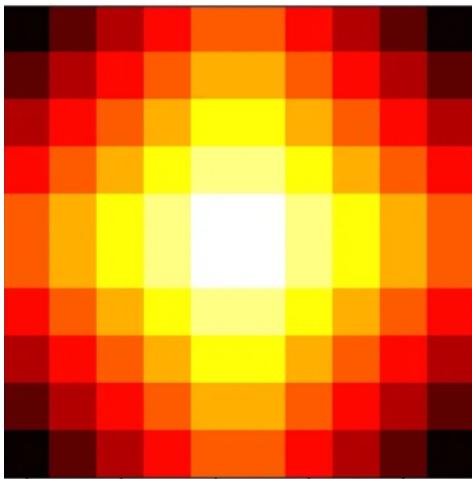


Figura 11: Ejemplo de representación de la matriz de probabilidad en un mapa de calor para un portaaviones

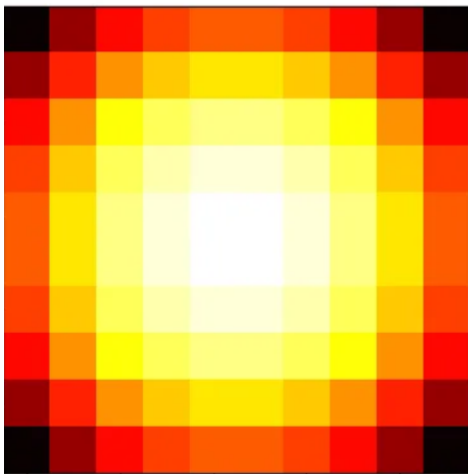


Figura 12: Ejemplo de representación de la matriz de probabilidad en un mapa de calor con todos los barcos

Descripción

Contrincante

Modo Hunt

- Al principio, la **matriz de probabilidades** es actualizada.
 - Las casillas disparadas o contiguas a barcos hundidos tendrán **probabilidad 0**.
 - Al calcular todas las posibles posiciones de los barcos, estos **no** pueden superponerse en casillas disparadas o contiguas a barcos hundidos.
- El objetivo será aquella casilla con **mayor probabilidad**.
 - Si el disparo **falla**, se volverá a intentar con la siguiente mayor en el siguiente turno.
 - Si el disparo **acierta**, se añaden las casillas vecinas a la lista de objetivos y se pasa el relevo al **modo target**.

Descripción

Contrincante

Modo Hunt

- Al principio, la **matriz de probabilidades** es actualizada.
 - Las casillas disparadas o contiguas a barcos hundidos tendrán **probabilidad 0**.
 - Al calcular todas las posibles posiciones de los barcos, estos **no** pueden superponerse en casillas disparadas o contiguas a barcos hundidos.
- El objetivo será aquella casilla con **mayor probabilidad**.
 - Si el disparo **falla**, se volverá a intentar con la siguiente mayor en el siguiente turno.
 - Si el disparo **acierta**, se añaden las casillas vecinas a la lista de objetivos y se pasa el relevo al **modo target**.

Descripción

Contrincante

Modo Hunt

- Al principio, la **matriz de probabilidades** es actualizada.
 - Las casillas disparadas o contiguas a barcos hundidos tendrán **probabilidad 0**.
 - Al calcular todas las posibles posiciones de los barcos, estos **no** pueden superponerse en casillas disparadas o contiguas a barcos hundidos.
- El objetivo será aquella casilla con **mayor probabilidad**.
 - Si el disparo **falla**, se volverá a intentar con la siguiente mayor en el siguiente turno.
 - Si el disparo **acierta**, se añaden las casillas vecinas a la lista de objetivos y se pasa el relevo al **modo target**.

Descripción

Contrincante

Modo Hunt

- Al principio, la **matriz de probabilidades** es actualizada.
 - Las casillas disparadas o contiguas a barcos hundidos tendrán **probabilidad 0**.
 - Al calcular todas las posibles posiciones de los barcos, estos **no** pueden superponerse en casillas disparadas o contiguas a barcos hundidos.
- El objetivo será aquella casilla con **mayor probabilidad**.
 - Si el disparo **falla**, se volverá a intentar con la siguiente mayor en el siguiente turno.
 - Si el disparo **acierta**, se añaden las casillas vecinas a la lista de objetivos y se pasa el relevo al **modo target**.

Descripción

Contrincante

Modo Hunt

- Al principio, la **matriz de probabilidades** es actualizada.
 - Las casillas disparadas o contiguas a barcos hundidos tendrán **probabilidad 0**.
 - Al calcular todas las posibles posiciones de los barcos, estos **no** pueden superponerse en casillas disparadas o contiguas a barcos hundidos.
- El objetivo será aquella casilla con **mayor probabilidad**.
 - Si el disparo **falla**, se volverá a intentar con la siguiente mayor en el siguiente turno.
 - Si el disparo **acierta**, se añaden las casillas vecinas a la lista de objetivos y se pasa el relevo al **modo target**.

Descripción

Contrincante

Modo Hunt

- Al principio, la **matriz de probabilidades** es actualizada.
 - Las casillas disparadas o contiguas a barcos hundidos tendrán **probabilidad 0**.
 - Al calcular todas las posibles posiciones de los barcos, estos **no** pueden superponerse en casillas disparadas o contiguas a barcos hundidos.
- El objetivo será aquella casilla con **mayor probabilidad**.
 - Si el disparo **falla**, se volverá a intentar con la siguiente mayor en el siguiente turno.
 - Si el disparo **acierta**, se añaden las casillas vecinas a la lista de objetivos y se pasa el relevo al **modo target**.

Descripción

Contrincante

Modo Target

- Usa la misma lógica que el modo target anterior.
- La matriz de probabilidades es usada para decidir en que orden disparar sobre las casillas vecinas.
- Si el disparo es sobre agua, se vuelven a calcular las probabilidades para elegir la casilla con mayor probabilidad.
- Cuando la orientación es elegida, se actúa de la forma anterior (no se usa la matriz de probabilidades).

Descripción

Contrincante

Modo Target

- Usa la misma lógica que el modo target anterior.
- La matriz de probabilidades es usada para decidir en que orden disparar sobre las casillas vecinas.
- Si el disparo es sobre agua, se vuelven a calcular las probabilidades para elegir la casilla con mayor probabilidad.
- Cuando la orientación es elegida, se actúa de la forma anterior (no se usa la matriz de probabilidades).

Descripción

Contrincante

Modo Target

- Usa la misma lógica que el modo target anterior.
- La matriz de probabilidades es usada para decidir en que orden disparar sobre las casillas vecinas.
- Si el disparo es sobre agua, se vuelven a calcular las probabilidades para elegir la casilla con mayor probabilidad.
- Cuando la orientación es elegida, se actúa de la forma anterior (no se usa la matriz de probabilidades).

Descripción

Contrincante

Modo Target

- Usa la misma lógica que el modo target anterior.
- La matriz de probabilidades es usada para decidir en que orden disparar sobre las casillas vecinas.
- Si el disparo es sobre agua, se vuelven a calcular las probabilidades para elegir la casilla con mayor probabilidad.
- Cuando la orientación es elegida, se actúa de la forma anterior (no se usa la matriz de probabilidades).

Fortalezas

- Si la flota es colocada aleatoriamente (densidad alta en el centro), el **algoritmo es muy eficaz**
- Permite ajustar la estrategia cambiando artificialmente los pesos según se desee
- **Fácilmente adaptable** para algoritmos más avanzados (Machine Learning)

Debilidades

- Usa una **estructura de datos auxiliar** extra (matriz de probabilidades).
- Supone un gasto mayor en recursos en calcular cada vez la matriz de probabilidades.
- Es **débil** si los barcos son colocados alejados del centro.

Fortalezas

- Si la flota es colocada aleatoriamente (densidad alta en el centro), el **algoritmo es muy eficaz**
- Permite ajustar la estrategia cambiando artificialmente los pesos según se desee
- **Fácilmente adaptable** para algoritmos más avanzados (Machine Learning)

Debilidades

- Usa una **estructura de datos auxiliar** extra (matriz de probabilidades).
- Supone un gasto mayor en recursos en calcular cada vez la matriz de probabilidades.
- Es **débil** si los barcos son colocados alejados del centro.

Fortalezas

- Si la flota es colocada aleatoriamente (densidad alta en el centro), el **algoritmo es muy eficaz**
- Permite ajustar la estrategia cambiando artificialmente los pesos según se desee
- **Fácilmente adaptable** para algoritmos más avanzados (Machine Learning)

Debilidades

- Usa una **estructura de datos auxiliar** extra (matriz de probabilidades).
- Supone un gasto mayor en recursos en calcular cada vez la matriz de probabilidades.
- Es **débil** si los barcos son colocados alejados del centro.

Fortalezas

- Si la flota es colocada aleatoriamente (densidad alta en el centro), el **algoritmo es muy eficaz**
- Permite ajustar la estrategia cambiando artificialmente los pesos según se desee
- **Fácilmente adaptable** para algoritmos más avanzados (Machine Learning)

Debilidades

- Usa una **estructura de datos auxiliar** extra (matriz de probabilidades).
- Supone un gasto mayor en recursos en calcular cada vez la matriz de probabilidades.
- Es **débil** si los barcos son colocados alejados del centro.

Fortalezas

- Si la flota es colocada aleatoriamente (densidad alta en el centro), el **algoritmo es muy eficaz**
- Permite ajustar la estrategia cambiando artificialmente los pesos según se desee
- **Fácilmente adaptable** para algoritmos más avanzados (Machine Learning)

Debilidades

- Usa una **estructura de datos auxiliar** extra (matriz de probabilidades).
- Supone un gasto mayor en recursos en calcular cada vez la matriz de probabilidades.
- Es **débil** si los barcos son colocados alejados del centro.

Fortalezas

- Si la flota es colocada aleatoriamente (densidad alta en el centro), el **algoritmo es muy eficaz**
- Permite ajustar la estrategia cambiando artificialmente los pesos según se desee
- **Fácilmente adaptable** para algoritmos más avanzados (Machine Learning)

Debilidades

- Usa una **estructura de datos auxiliar** extra (matriz de probabilidades).
- Supone un gasto mayor en recursos en calcular cada vez la matriz de probabilidades.
- Es **débil** si los barcos son colocados alejados del centro.

Descripción modular del código

- 1 Introducción
- 2 Descripción
- 3 Descripción modular del código**
- 4 Resultados
- 5 Conclusiones
- 6 Referencias

Descripción modular del código

Estructuras de datos

3 Descripción modular del código

- Estructuras de datos
 - Barcos
 - Tableros/flota
 - Mapas de disparo
 - Matriz de probabilidades
- Estructura de archivos

Descripción modular del código

Estructuras de datos

3 Descripción modular del código

- Estructuras de datos
 - Barcos
 - Tableros/flota
 - Mapas de disparo
 - Matriz de probabilidades
- Estructura de archivos

Descripción modular del código

Estructuras de datos

Barcos

- Vectores de vectores (matrices)
- Cada fila representa a un **tipo de barco** (portaaviones, acorazado, crucero, destructor o submarino).
- Las columnas son los siguientes campos por cada tipo de barco:
 - ID: número para identificar el tipo de barco
 - Nombre: cadena con el nombre del tipo
 - Número de barcos (sin colocar): número de barcos en total
 - Tamaño del barco: número de casillas que ocupan
 - Color: cadena con el color que son representados
 - Barcos restantes (sin hundir): número de barcos a flote
 - Lista con el tamaño de cada barco: controla el tamaño de cada barco

Descripción modular del código

Estructuras de datos

3 Descripción modular del código

- Estructuras de datos
 - Barcos
 - Tableros/flota
 - Mapas de disparo
 - Matriz de probabilidades
- Estructura de archivos

Descripción modular del código

Estructuras de datos

Tableros/flota

- Vectores de vectores (matrices)
- Representan **el tablero de cada jugador.**
- Guardan la información de la colocación de los barcos.
- El agua es representada con un 0 mientras que cada barco es una lista:
 - ID: número del tipo de barco.
 - SubID: número que identifica al barco dentro de su mismo tipo.
 - Posición proa: sublista con las coordenadas de la casilla dónde se sitúa la proa del barco.
 - Tamaño del barco: número que indica el número de casillas que ocupa
 - Orientación: booleano que indica su posición.

Descripción modular del código

Estructuras de datos

3 Descripción modular del código

- Estructuras de datos
 - Barcos
 - Tableros/flota
 - Mapas de disparo
 - Matriz de probabilidades
- Estructura de archivos

Descripción modular del código

Estructuras de datos

Mapas de disparos

- Vectores de vectores (matrices)
- Representan los **disparos que ha realizado el jugador** en una casilla del tablero del contrincante.
- Mismo tamaño que los tableros.
- Contienen la siguiente información:
 - 0: casilla sin descubrir.
 - 1: casilla de barco descubierta
 - -1: casilla de agua descubierta
 - -2: exclusión alrededor de barco hundido (solo es usado por los algoritmos de la máquina)

Descripción modular del código

Estructuras de datos

3 Descripción modular del código

- Estructuras de datos
 - Barcos
 - Tableros/flota
 - Mapas de disparo
 - Matriz de probabilidades
- Estructura de archivos

Descripción modular del código

Estructuras de datos

Matriz de probabilidades

- Vectores de vectores (matrices)
- Representa la **probabilidad de cada casilla de contener un barco.**
- Mismo tamaño que los tableros.
- Solo es usado por el algoritmo Hunt/Target probabilístico
- Contiene por cada casilla la suma de los pesos.

Descripción modular del código

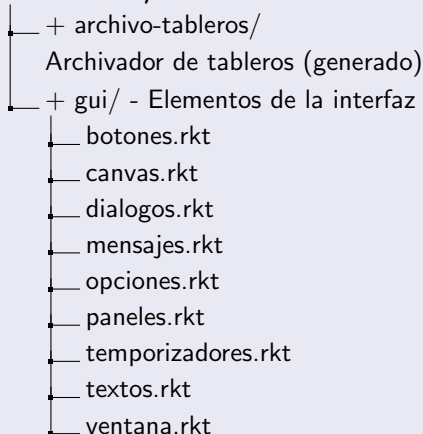
Estructura de archivos

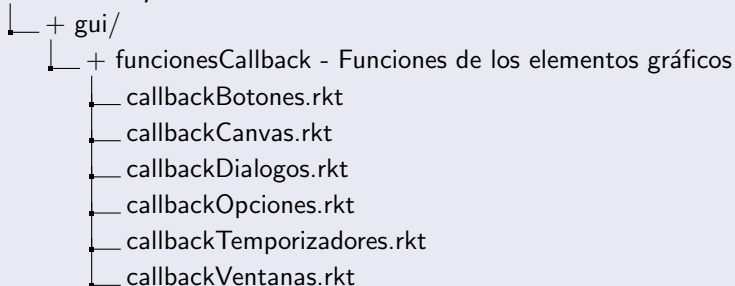
- 3 Descripción modular del código
 - Estructuras de datos
 - Estructura de archivos

Descripción modular del código

Estructura de archivos

HundirFlota/



HundirFlota/

HundirFlota/

- funciones.rkt
Funciones variadas
- funcionesGUI.rkt.
Funciones variadas de GUI
- turno.rkt
Función cambio de turno

HundirFlota/

- funcionesLimpiar.rkt
Limpieza de canvases
- funcionesColocar.rkt
Colocación de barcos y su lógica
- funcionesDibujar.rkt
Dibujado en canvas
- funcionesDisparar.rkt
Lógica de disparar
- funcionesGenerar.rkt
Generación aleatoria en la colocación de barcos
- funcionesLogica.rkt
Lógica del juego

HundirFlota/

- + imagenes/
Recursos gráficos
- + ayuda/
Recursos de ayuda
- main.rkt
Programa principal
- estructuras.rkt
Estructuras principales
- macros.rkt
Macros globales

Resultados

- 1 Introducción
- 2 Descripción
- 3 Descripción modular del código
- 4 Resultados**
- 5 Conclusiones
- 6 Referencias

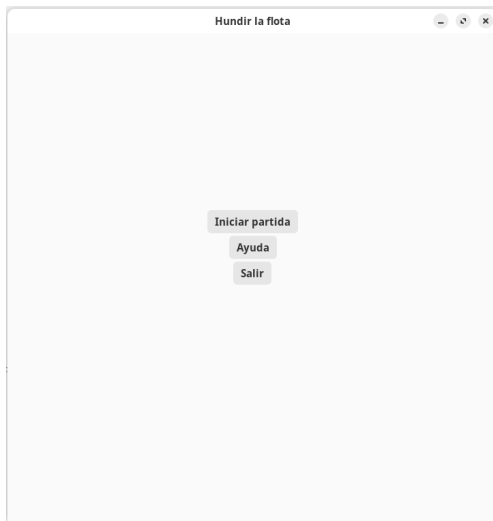


Figura 13: Menú de inicio de la aplicación

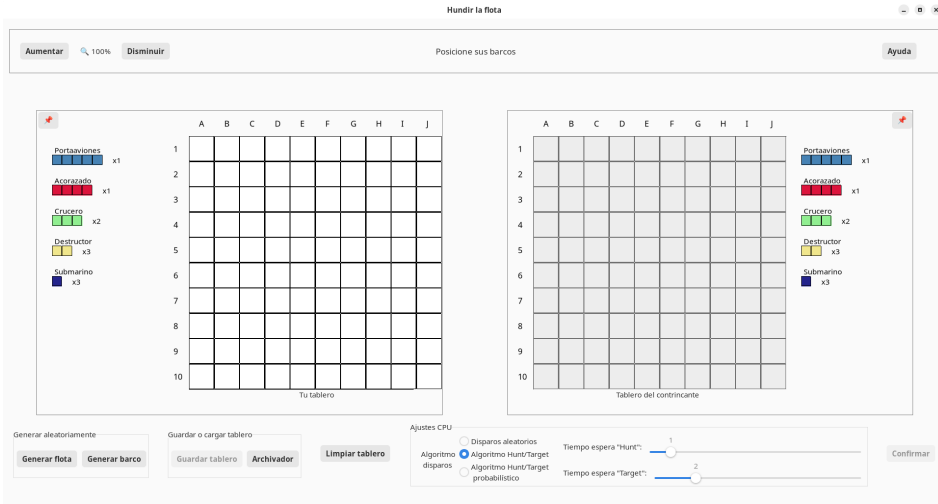


Figura 14: Pantalla principal de la aplicación

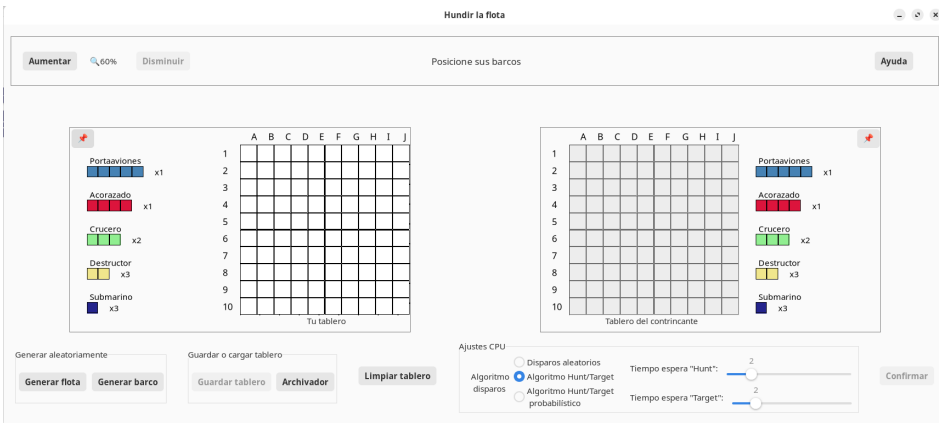


Figura 15: Disminuir el tamaño de la interfaz

Figura 16: Ejemplo de colocación de los barcos aleatoria

Hundir la flota

Aumentar 100% Disminuir

Posicione sus barcos

Ayuda

Portaaviones

¡Colocados!

Acorazado

¡Colocados!

Crucero

¡Colocados!

Destructor

¡Colocados!

Submarino

x3

	A	B	C	D	E	F	G	H	I	J
1										
2										
3										
4										
5										
6										
7										
8										
9										
10										

Tu tablero

Portaaviones

x1

Acorazado

x1

Crucero

x2

Destructor

x3

Submarino

x3

	A	B	C	D	E	F	G	H	I	J
1										
2										
3										
4										
5										
6										
7										
8										
9										
10										

Tablero del contrinicante

Generar aleatoriamente

Generar flota Generar barco

Guardar o cargar tablero

Guardar tablero Archivar

Limpiar tablero

Ajustes CPU

☐ Disparos aleatorios
☒ Algoritmo Hunt/Target
☐ Algoritmo Hunt/Target probabilístico

Tiempo espera "Hunt": 1
 Tiempo espera "Target": 2

Confirmar

Figura 17: Ejemplo de colocación de los barcos manual

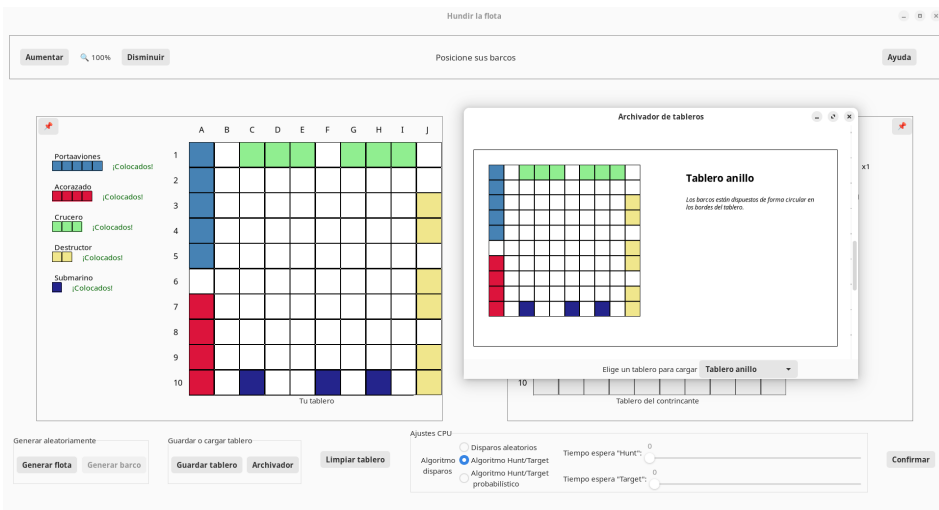


Figura 18: Uso del archivador

Hundir la flota

Aumentar 100% Disminuir

Posicione sus barcos

Ayuda

Tu tablero

Tablero del contrinante

Guardar tablero en archivador

Datos del tablero

Nombre del tablero: Tablero ejemplo

Descripcion del tablero

Esto es un ejemplo de tablero

Fichero con el tablero guardado correctamente

Guardar

Nombre del fichero: tablero4

Generar aleatoriamente

Generar flota

Generar barco

Guardar o cargar tablero

Guardar tablero

Archivador

Limpiar tablero

Ajustes CPU

☐ Disparos aleatorios

☒ Algoritmo Hunt/Target

☐ Algoritmo Hunt/Target probabilístico

Tiempo espera "Hunt": 0

Tiempo espera "Target": 0

Confirmar

Figura 19: Guardar un tablero en el archivador

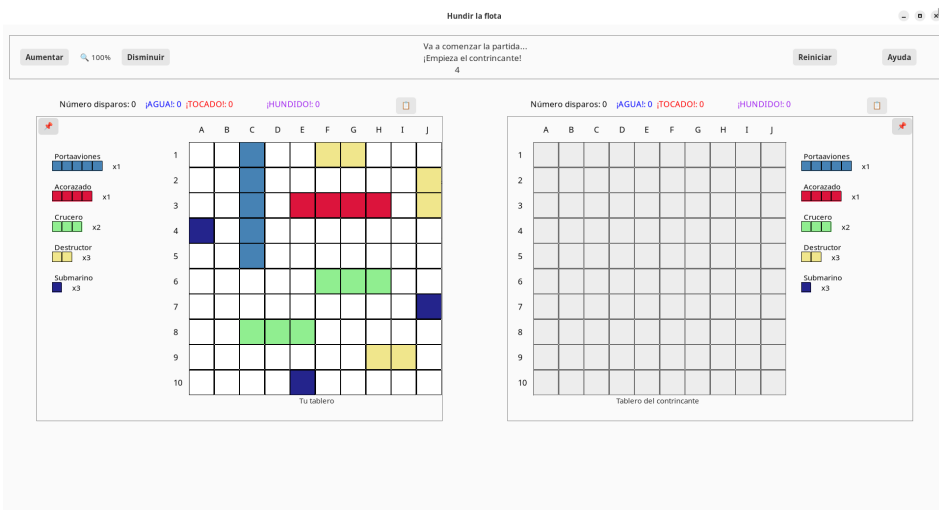


Figura 20: Inicio de partida

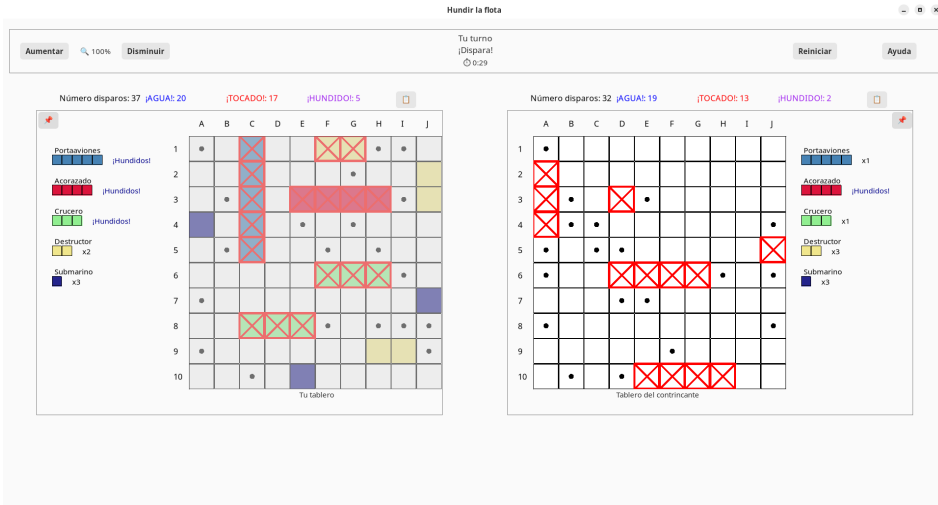


Figura 21: Turno del jugador

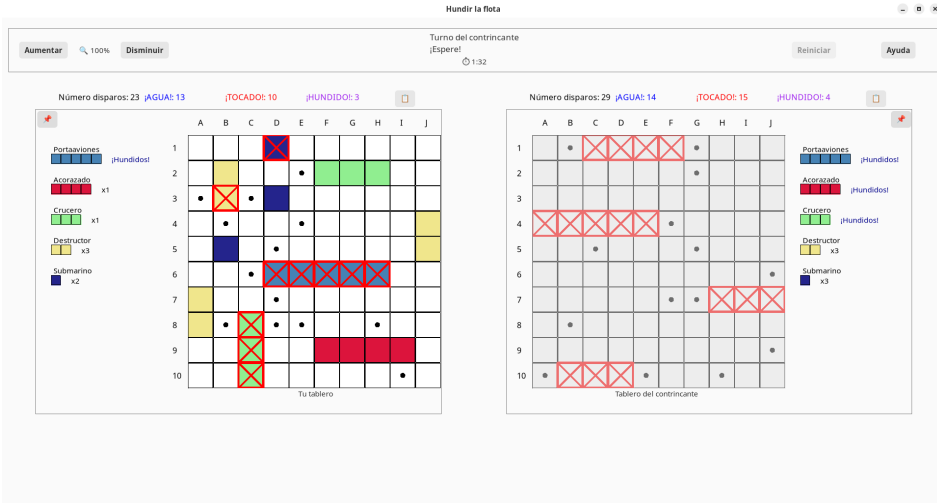


Figura 22: Turno de la máquina (CPU)

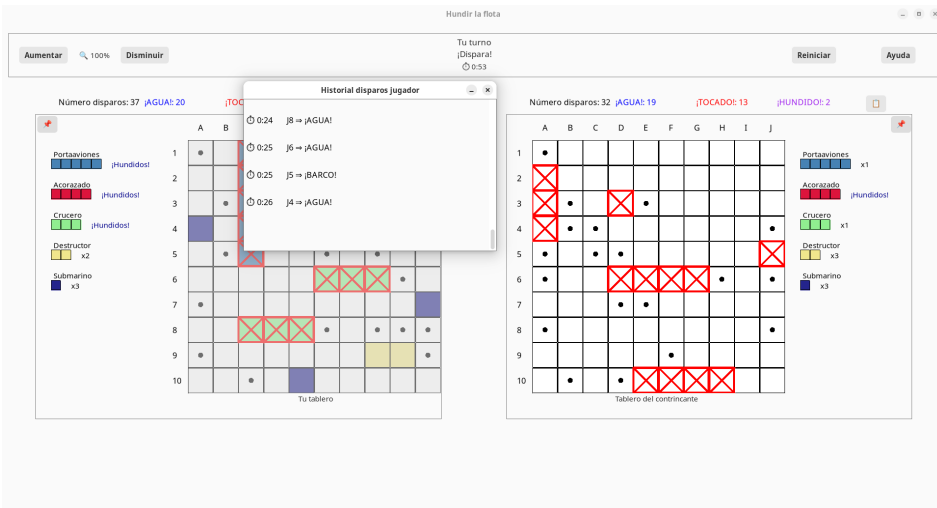


Figura 23: Uso del historial de disparos



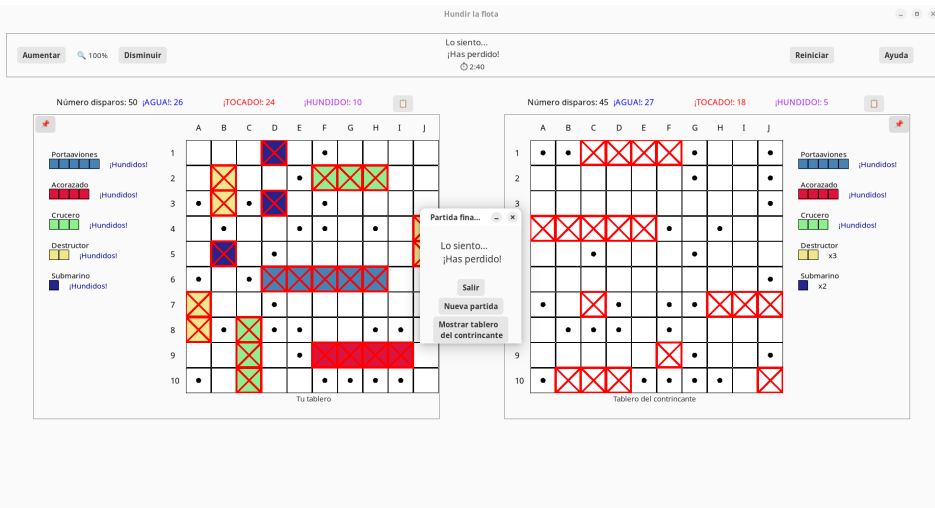


Figura 25: Final de una partida

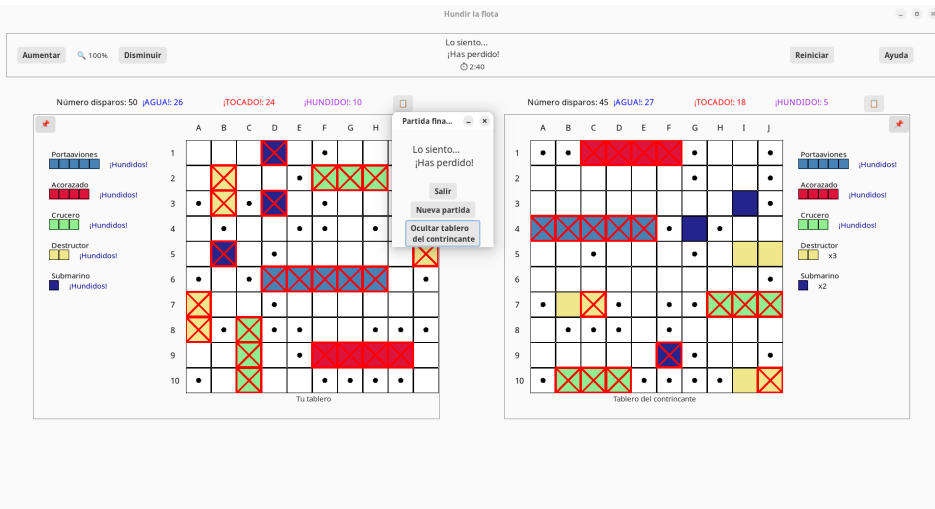


Figura 26: Mostrar el tablero de la CPU al finalizar

Conclusiones

- 1 Introducción
- 2 Descripción
- 3 Descripción modular del código
- 4 Resultados
- 5 Conclusiones**
- 6 Referencias

Conclusiones

Racket es un lenguaje diferente diseñado para ser usado con la programación declarativa, pero no ha quedado duda que es lo suficientemente potente e intuitivo como cualquier otro lenguaje más popular.

Sus librerías gráficas, *GUI* y *Draw* permiten construir aplicaciones de forma sencilla y deja la puerta abierta para indagar más profundamente si se necesita desarrollar características más complejas.

Usar Racket como lenguaje para desarrollar aplicaciones está a la altura de otros *frameworks* como *Qt* o *Gtk*.

Referencias

- 1 Introducción
- 2 Descripción
- 3 Descripción modular del código
- 4 Resultados
- 5 Conclusiones
- 6 Referencias**

Referencias

Sitios de interés:



Racket Documentation (v8.11)

The Racket Graphical Interface Toolkit

Consultado el 11 de enero de 2024



Racket Documentation (v8.11)

The Racket Drawing Toolkit

Consultado el 11 de enero de 2024



Battleship (s.f.)

Obtenido de enciclopedia libre Wikipedia

Consultado el 11 de enero de 2024

Referencias

Sitios de interés:



James W. Stelly (2021)

Racket Programming the Fun Way

Consultado el 11 de enero de 2024



Nick Berry (2011)

Battleship algorithms

Consultado el 11 de enero de 2024



Aydin Schwartz (2022)

Coding an Intelligent Battleship Agent

Consultado el 11 de enero de 2024

Referencias

Sitios de interés:



PLT (Version 200, June 2002)

[PLT Miscellaneous Libraries: Reference Manua](#)

Consultado el 11 de enero de 2024



Francisco Javier Rodríguez Lozano (Curso académico 2013 - 2014. UCO)

[Representación Gráfica en Scheme](#)

Consultado el 11 de enero de 2024

Referencias

Sitios de interés:



battleship-game.org

Hundir la flota online

Consultado el 11 de enero de 2024



Domingo Gallardo

Seminario de Scheme

Consultado el 11 de enero de 2024



Philip F.

Strategies in battleship

Consultado el 11 de enero de 2024

Referencias

Sitios de interés:



Bas Steunebrink

Create executable for "Pretty Big" legacy program

Consultado el 11 de enero de 2024



u/fazzah

Battleship game algorithm comparison with results

Consultado el 11 de enero de 2024



Hasbro

Battleship rules

Consultado el 11 de enero de 2024