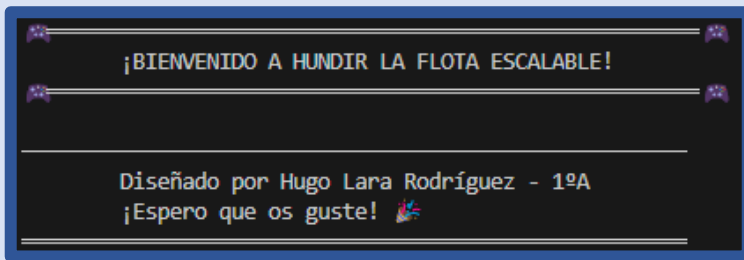


Proyecto Final de PROGRAMACIÓN – 1er curso

Grado en Ingeniería Matemática e Inteligencia Artificial

Hundir la flota

Hola, bienvenido al juego Hundir la Flota mejorado. Como proyecto me falta algo de originalidad por coger el predeterminado, así que espero que se supla con las funcionalidades avanzadas que he implementado. Dejo por aquí una lista con algunas de las cosas con las que cuenta; por favor Pruébalo para conocerlo por completo: (los requisitos básicos del proyecto están cumplidos, estos que menciono son extras o estos requisitos bastante mejorados)



(pantalla introductoria)

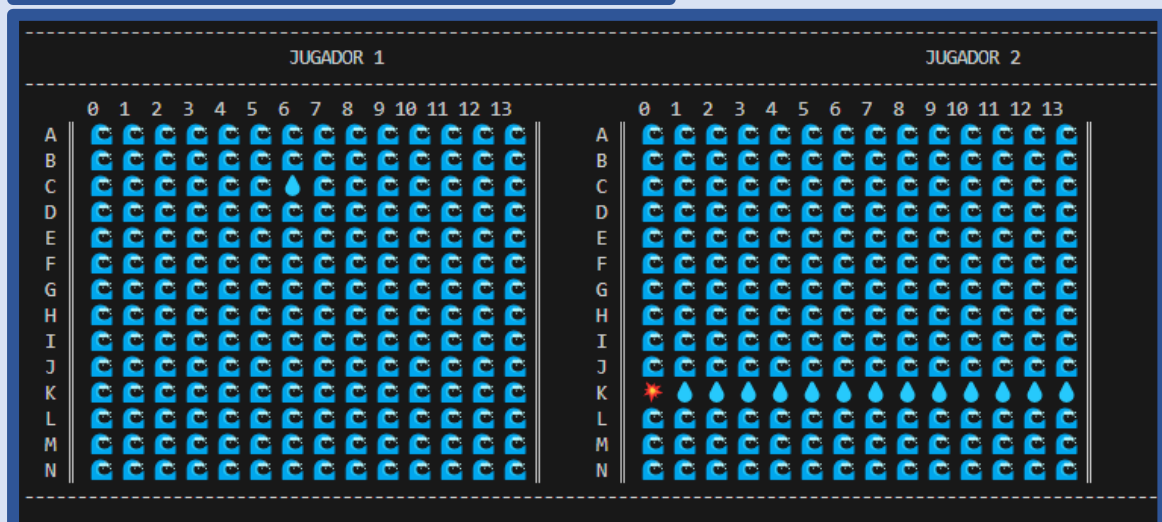
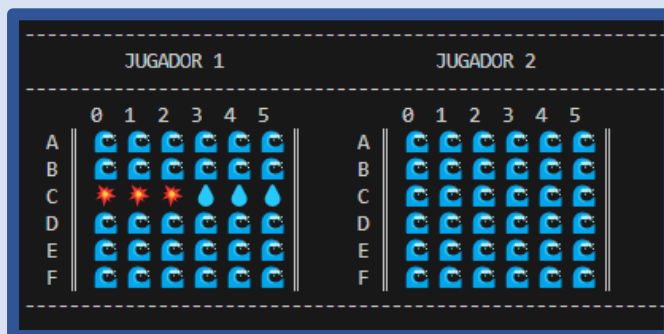


Importante: En este documento figura lo más importante, pero sigue habiendo otras funcionalidades:

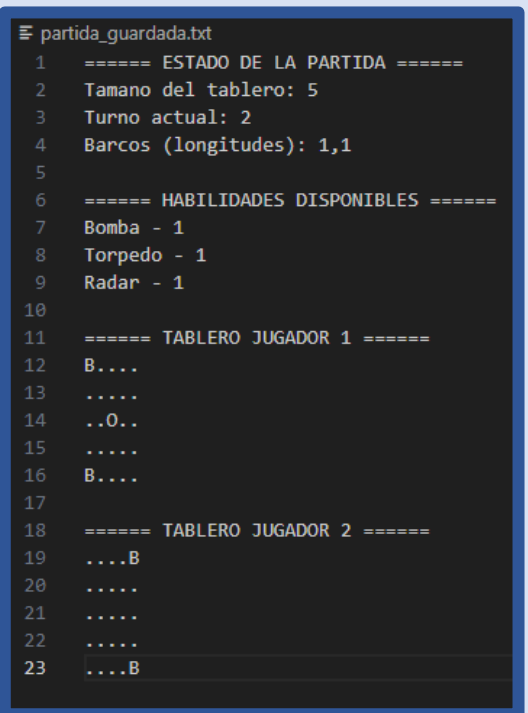
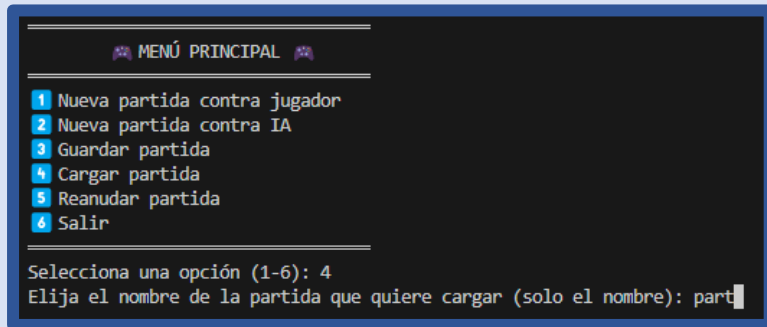
¡Pruébalas!

Generales:

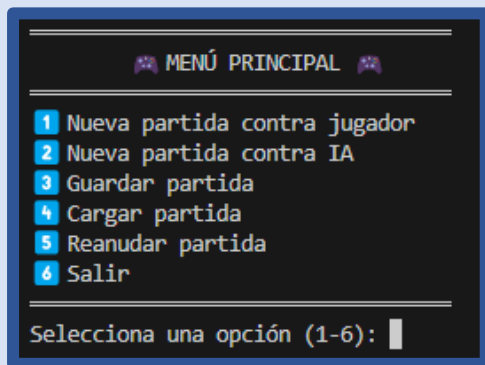
- Tablero totalmente modificable: en vez de una anchura rígida, el juego siempre te pregunta por la anchura del tablero con el que quieres jugar (de 5x5 a 24x24 por temas de la visualización). Además, los emojis que representan cada situación son escalables, se pueden cambiar desde un diccionario en constantes.py



- Sistema de guardado y carga de partidas con nombre propio del jugador para la partida y guardado que cuida la UX, para poder ver el estado de la partida fuera de la misma (ej.: en partida_guardada.txt). Además, se podrá elegir el nombre de la partida a guardar, pudiendo crear cuantas partidas guardadas se quieran



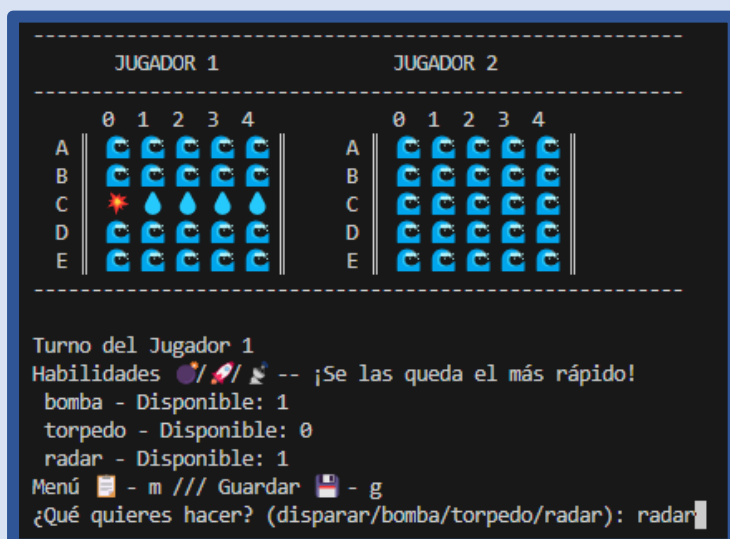
- Menú principal con opciones añadidas



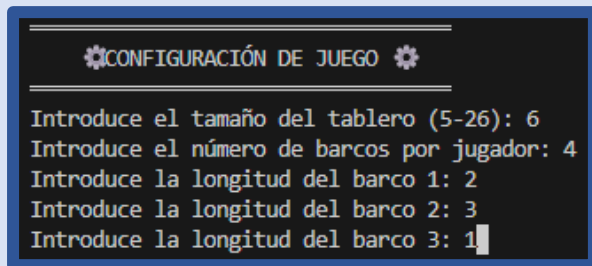
- Diferentes niveles de dificultad según la configuración

Funcionalidades del Juego:

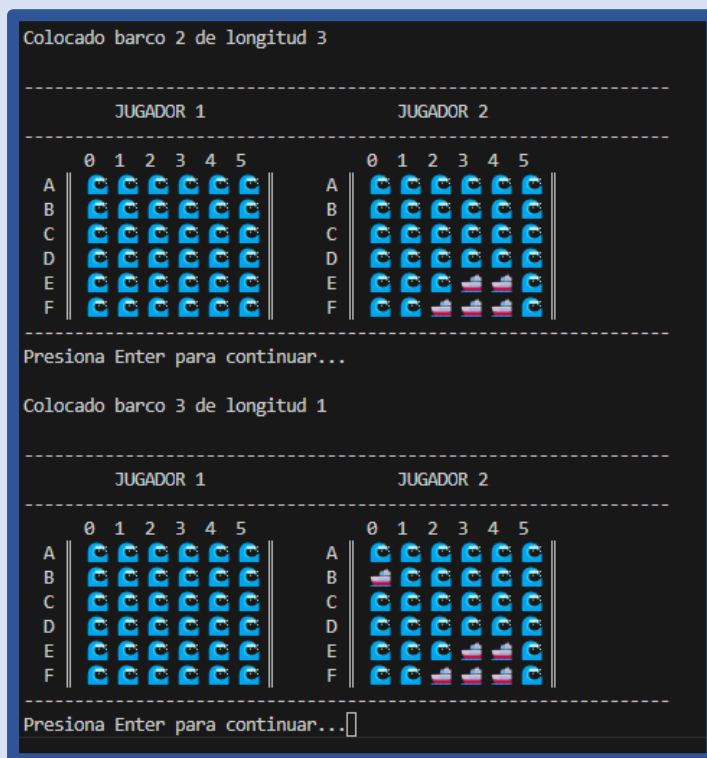
- Habilidades Especiales: disparo (habitual), bomba (ataque en área 3x3), torpedo (ataque en línea (fila o columna)), radar (detecta barcos en área 3x3), ... (son totalmente escalables y se pueden añadir más, ya que SON FUNCIONES INDEPENDIENTES)



- El número de barcos es totalmente libre y modificable, así como sus longitudes. Se pregunta todo al iniciar partida, y luego se guarda también

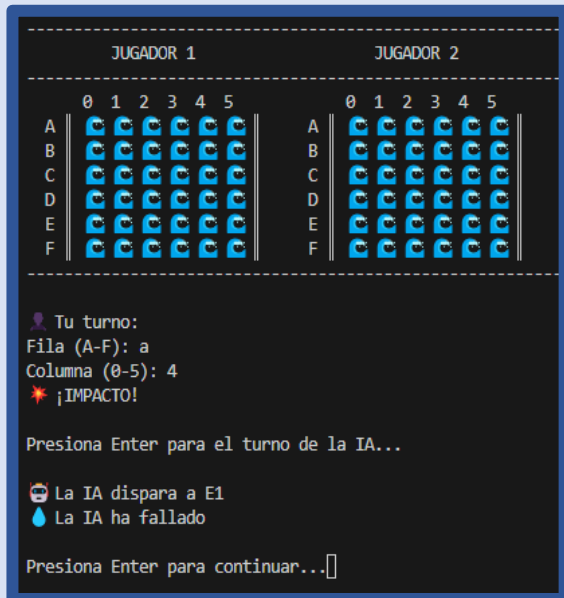


- Posibilidad de colocación manual de los barcos o aleatoria (por si estás vago, mejor que una colocación predeterminada)



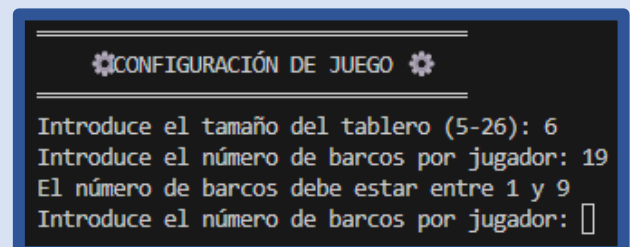
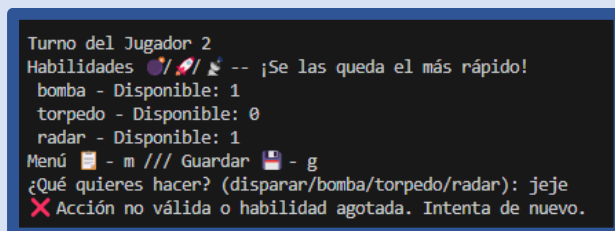
UX:

- Visualización clara de dos tableros simultáneamente de manera horizontal
- Interfaz en consola con símbolos Unicode para mejor visualización con emojis (🔴, 🚀, 📡, ...)
- Borrado periódico de la consola para mantener una interfaz limpia



Errores:

- Validación de todas las entradas por el jugador y manejo de errores en todas las operaciones (está todo capado)



Cambios respecto a la funcionalidad del hundir la flota clásico:

- Sistema de armas, siendo el que llegue primero quien se la queda. Un turno por jugador, aunque se dé al objetivo (en tableros pequeños es más justo con las habilidades)

Características técnicas:

Este proyecto consta de los siguientes elementos de desarrollo de software:

- $4 + 35 + 46 + 703 + 55 = 843$ líneas de código aproximadamente
- 5 módulos
- 28 funciones

- 1 MEGA clases
- Infinitos ficheros de datos (puedes guardar infinitas partidas con el nombre que quieras)
- Librerías:
 - random, time, os

Funcionalidad del programa:

La parte principal del programa se basa en una única clase, ya que así es más fácil compartir información entre funciones en forma de atributos; siendo estos los tableros (lista de listas), el tamaño, los barcos (una lista con la longitud de cada barco), el turno actual, el estado de la partida, los símbolos para cada elemento y un diccionario con las habilidades y sus usos.

Básicamente la lógica del juego es la siguiente:

El programa comienza con `iniciar_juego()` el cual controla el flujo principal, que funciona como menú, permitiendo elegir todas las opciones. Se puede elegir crear una nueva partida, entonces, se utiliza el método `configurar_juego()`; si se aceptan los valores pedidos (básicamente el tamaño del tablero y número/longitud de barcos) se entra en `colocar_barcos_jugadores()` donde cada jugador tiene la opción de colocar sus barcos manual o aleatoriamente, buena funcionalidad para vagos jeje. Luego, si todo va bien, se entra en `jugar_partida()` que se encarga de gestionar los turnos alternando entre jugadores, donde se realizan las acciones como disparar o usar habilidades especiales (bomba, torpedo, radar, ...) mediante sus métodos. Todo ello se va imprimiendo y borrando de la pantalla, de manera limpia con el método de `imprimir_tableros()`, mostrando la situación de los tableros ocultando los barcos del oponente. Para el modo contra IA, he creado una lógica específica, más aislada y compleja: se empieza con `jugar_contra_ia()` que implementa la colocación automática de barcos con `colocar_barcos_ia()` y disparo mediante `disparo_ia()`. Todo ello, la IA y no IA, es validado mediante los métodos `es_posicion_valida()` y `pedir_coordenadas()`, asegurando que los movimientos sean legales. La partida continúa hasta que `quedan_barcos()` ejecutado en cada ronda diga lo contrario, terminando la partida. Luego, está también la funcionalidad para `guardar_partida()` y `cargar_partida()`, permitiendo continuar partidas previas.