

*The Earth is the cradle of Humanity.
But one doesn't always live in the cradle.*

Konstantin Tsiolkovsky

*"The best computer is a man,
and it's the only one that can be mass-produced
by unskilled labor."*

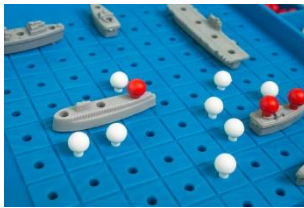
Wernher von Braun

*"There's a silly notion that failure's not an option at NASA.
Failure is an option here.
If things are not failing, you are not innovating enough."*

Elon Musk

Project 2

Computer Engineering



Sea Battle

Hit or miss game

Professors:

Prof. Jeff Schmidt Peralta (G1),

Prof. Milton Villegas Lemus (G2),

Prof. Leonardo Araya Martínez (G3),

Prof. Luis Alonso Barboza Artavia (G4),

Prof. Jason Leitón Jiménez (G5)

Based on the proposal by Jason Leitón Jiménez

Costa Rica, May 20th 2022

Introducción

Battleship (batalla naval) es un juego de mesa de estrategia muy popular. La idea principal del juego es disponer de un tablero y **colocar barcos de diferentes tamaños**. Un segundo jugador buscará tratar de **hundir los barcos contrarios indicando una coordenada del tablero**. **El ganador del juego es quien hunda primero todos los barcos del otro jugador**. Una de las primeras versiones se elaboró en 1930 utilizando papel y lápiz. En 1967, Milton Bradley realizó el juego con piezas plásticas. Desde su lanzamiento se estima que se han vendido más de 100 millones de copias del juego. Curiosamente Battleship fue uno de los primeros juegos producidos para computadoras, ya que en 1979 se lanzó para la Z80 Compucolor. Desde ahí, se han realizado versiones del juego para diversas consolas de videojuegos y redes sociales.



Figura 1. Tablero del jugador. En la parte inferior se colocan los barcos para que el otro jugador trate de hundirlos. En la parte superior se colocan las adivinanzas del jugador. Se diferencia un acierto con color rojo.

Aunque se cree que el juego surge por la Primera Guerra Mundial, antes de este periodo existían juegos de mesa con una idea similar. En 1890 se publicó el juego de estrategia Basilinda. A diferencia de Battleship, en el tablero se colocaban cañones, un capitán y un ejército de 18 hombres.

Arduino UNO

El **Arduino Uno es una placa microcontrolador** basado en el microchip ATmega328P desarrollado por Arduino.cc. Esta empresa se dedica a fabricar kits de placas de desarrollo para construir dispositivos digitales. La principal característica de los productos que realizan es que son de **hardware y software abierto**.

Esta placa de desarrollo está **equipada con diversos pines los cuales se pueden programar como entrada o salida**, tanto **analógicas como digitales**. El propósito de esto es brindarle opciones al programador a la hora del desarrollo de un dispositivo.

Específicamente tiene 14 pines digitales y 6 analógicos. Para realizar la programación, se realiza por medio de un **USB tipo B**. Adicionalmente, puede ser alimentado por medio de una **batería de 9V** o por el **cable USB**.



Figura 2. Placa de desarrollo Arduino UNO.

Diodo emisor de luz (LED)

Un diodo emisor de luz (LED por sus siglas *light-emitting diode*) es un componente de dos terminales que **emite color cuando fluye corriente por él**. Es un elemento muy popular en cualquier dispositivo ya que sirven como medio visual para presentar un estado o respuesta que se le quiera dar a un usuario. Este componente **puede ser controlado por medio de un dispositivo externo para encenderlo o apagarlo**.

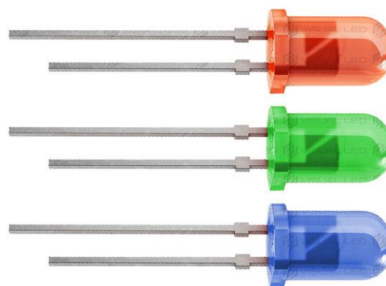


Figura 3. Tres diodos emisor de luz

Este segundo proyecto está ambientado en el juego Battleship, donde se jugará **contra la computadora o contra otra persona**. Adicionalmente, se podrá jugar por medio de un **control (botones) hecho por los estudiantes para seleccionar las coordenadas**, así como **recibir señales visuales (leds) de las acciones realizadas en el juego**. Esta comunicación se realizará por medio de un Arduino.

Objetivo general

- Diseñar una solución computacional relacionada con conceptos de recursividad, iteración, almacenamiento en disco y estructuras de datos, con el fin de aplicar la teoría en problemas específicos, considerando las buenas prácticas de programación.

Objetivos específicos

- Manipular estructuras de datos en dos dimensiones para modelar problemas abstractos, utilizando las listas de Python.
- Interactuar con memoria secundaria para almacenar datos con el fin de recuperarlos posteriormente mediante archivos.
- Implementar un programa con elementos de multimedia con el fin de realizar la interfaz de una manera amigable con el usuario, utilizando bibliotecas para gráficos y audio.

Descripción

El proyecto está basado en el juego Battleship, el cual contará con al menos dos pantallas principales cuya función se detalla continuación:

- 1) Pantalla Inicial: En esta pantalla se solicitará el nombre del jugador y la cantidad de naves de cada tipo para colocarlas en tablero (naves de tipo1, tipo2, tipo3). Además, existirá algún mecanismo para abrir la sección de ayuda, salón de la fama y abrir un juego guardado. En la figura 2 se muestra un ejemplo sencillo de esta pantalla inicial.



Figura 3. Pantalla Inicial

- 1.1) Ayuda: Esta sección es una pantalla que contiene las indicaciones de cómo utilizar el juego, debe guiar al usuario para utilizar la aplicación, además de indicar la información de los desarrolladores.
- 1.2) Salón de la fama: En esta pantalla aparecerán los 10 jugadores con mejores puntajes (los que ganaron más rápido), deberán estar ordenados de acuerdo con el tiempo.

- 1.3) Abrir: El juego proporcionará la posibilidad de cargar una partida previamente guardada, con el fin de que el usuario no inicie un nuevo juego, sino que continúe con que guardó previamente. Para esta sección se aconseja utilizar archivos de textos.
- 2) Pantalla de juego: Esta pantalla consiste en un tablero de 10x10 para el jugador y otro tablero de 10x10 para la computadora. El jugador colocará la cantidad de naves correspondientes de acuerdo con la configuración inicial. Cabe destacar que las naves del computador se colocarán aleatoriamente. En esta sección se deberá proporcionar el mecanismo para que el usuario guarde el estado actual de la pantalla, con el fin de que se pueda recuperar posteriormente.

Modo de juego

Una vez que el jugador haya configurado su nombre y la cantidad de naves de cada tipo, el juego deberá mostrar tanto el tablero del computador con las naves ya colocadas, como el del jugador para que este pueda iniciar a colocar las naves según su estrategia. La figura 4 muestra un ejemplo de los tableros y la colocación de naves (En este ejemplo se colocan naves de un espacio).

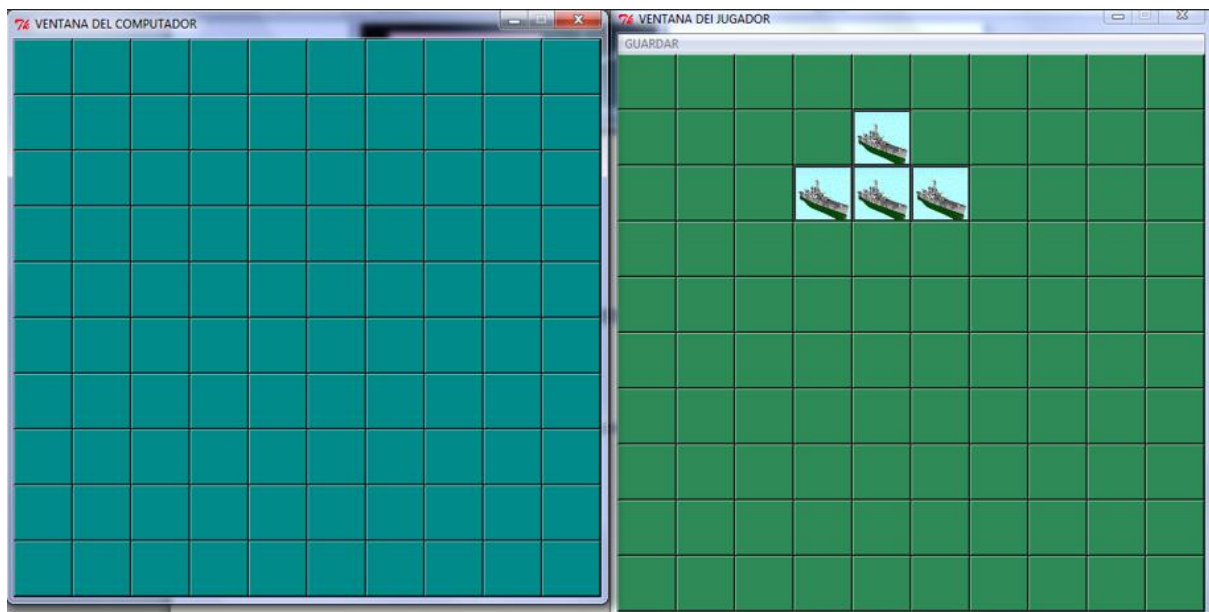


Figura 4. Tableros de juego

Cuando el jugador haya terminado de colocar todas las naves, el juego dará inicio. El usuario deberá elegir una casilla del tablero del computador para tratar de adivinar donde están las naves, en caso de que acierte se realizará una animación con sonido de explosión, mientras que si falla el sonido será de pérdida de turno. En el caso de que se acierte, se continuará con el turno, sin embargo, si falla cederá el turno al contrincante. En el caso del computador, se debe elegir la casilla aleatoriamente

(tomando en cuenta que no haya sido escogida antes). El juego debe ser robusto y realizar todas las validaciones para que el usuario no pueda hacer trampa, por ejemplo, saltarse de turno o jugar cuando no debe.

El juego será capaz de mostrar las casillas que han sido elegidas tanto por el usuario como por el computador, además, cada acierto o fallo será notificado de manera gráfica como se muestra en la figura 5.

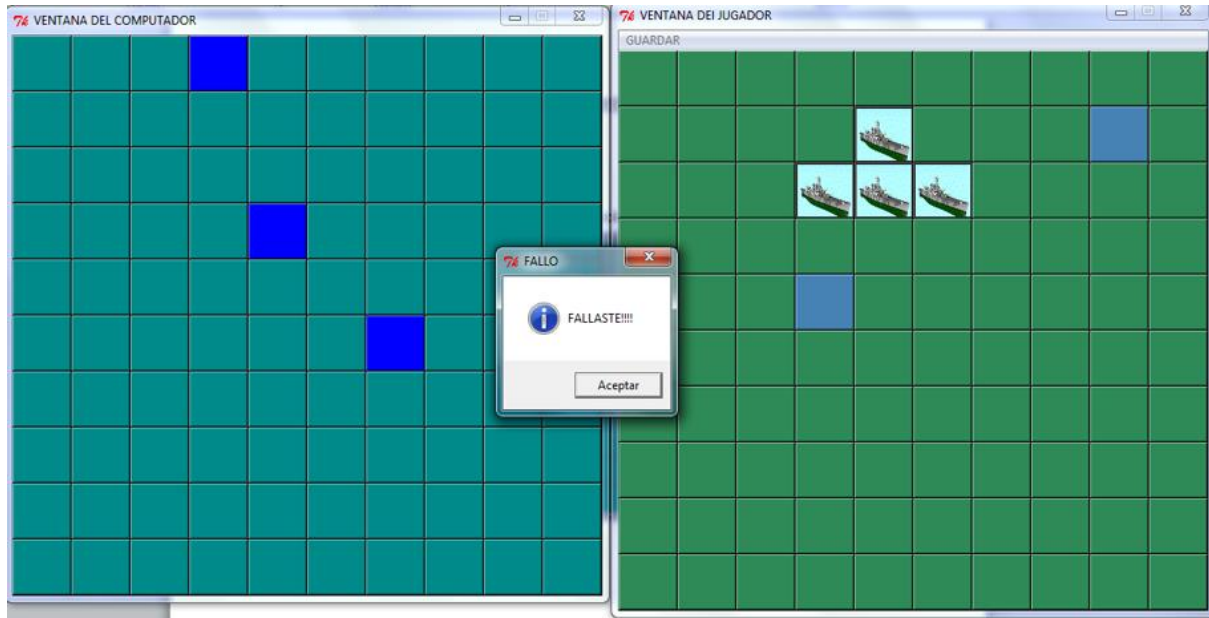


Figura 5. Casillas elegidas y notificación de fallo

El juego acaba cuando el computador o el jugador hayan adivinado todas las naves del oponente. Cuando esto suceda la aplicación notificará al usuario las estadísticas básicas, las cuales corresponden al nombre, minutos que duró la partida, cantidad de fallos, cantidad de aciertos y total de jugadas, en la figura 6 se muestra un ejemplo de las estadísticas.

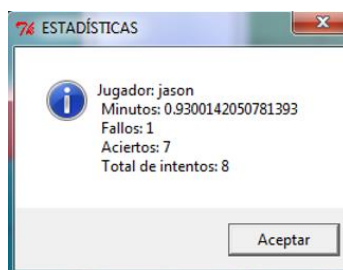


Figura 6. Estadísticas de juego

También se mostrará el lugar en el que se estableció dentro del salón de la fama, posteriormente que se haya mostrado las estadísticas, en la figura 7 se muestra un ejemplo.

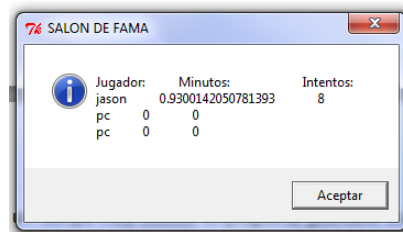


Figura 7. Salón de la fama

Tipos de naves

Existen tres tipos de naves que el usuario puede colocar, el primer tipo son aquellas naves que abarcan un espacio del tablero, el segundo tipo son las naves que cubren 2 espacios del tablero, mientras que el tercer tipo son naves con un tamaño de 4 espacios del tablero. Cabe resaltar, que el usuario puede colocar las naves de manera horizontal o vertical y la cantidad que haya sido especificada en la pantalla inicial. Es importante mencionar que cada tipo de nave debe ser representada con una imagen diferente de acuerdo con el tamaño. El juego validará que la configuración inicial sea posible, ya que el usuario puede cometer un error, por ejemplo, colocar 50 naves del tipo 3.

Sonidos

Durante toda la ejecución se debe tener sonido de fondo que agregue emoción al usuario, además, cuando se haga fallo y acierto debe aparecer una animación y sonido de acuerdo con la situación. Cabe destacar que cuando el jugador entre en el salón de la fama se ejecutará una animación de felicitación y un sonido acorde.

Puntos extras

Este apartado corresponde a 30 puntos extras, cada grupo puede elegir uno de los siguientes enfoques.

Software

El juego podrá jugarse entre dos usuarios, cada uno, utilizando máquinas distintas. Para este apartado se aconseja utilizar sockets para comunicar procesos.

Hardware

El juego tendrá un mando realizado por cada grupo, el cual consiste en un control para manipular los movimientos dentro del tablero. Cada grupo deberá hacer la maqueta de un control, el cual debe tener los siguientes elementos:

- Botón de seleccionar: al presionar el botón se selecciona la casilla en el tablero.
- Botón derecha: la selección se mueve a la derecha.
- Botón izquierda: la selección se mueve a la izquierda.
- Botón abajo: la selección se mueve a hacia abajo.
- Botón arriba: la selección se mueve a hacia arriba.
- Led fallo: Se enciende si el usuario falla.
- Led acierto: Se enciende si el usuario acierta.
- Led gane: Se enciende si el jugador gana.
- Led salón de fama: Se enciende si queda dentro del salón de la fama.

El circuito que se recomienda para realizar la componente de hardware es el que se muestra en la figura 8.

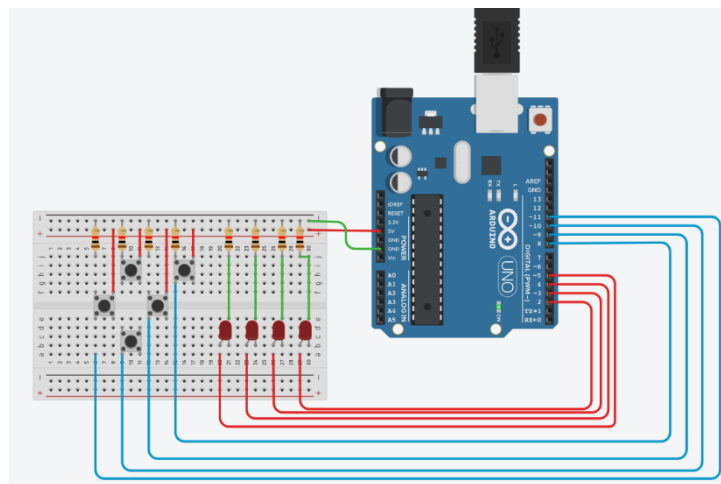


Figura 8. Circuito propuesto

Cabe resaltar que es de manera obligatoria realizar la maqueta del control para interactuar con la aplicación. No se permite presentar el circuito únicamente en la protoboard.

Evaluación

Ítem	Valor (%)
Tableros (Lógica e interfaz)	15
Cargar y guardar	10
Sonidos	8
Animaciones	7
Pantalla inicial	5
About	5
Salón de la fama	10
Tipos de naves	10
Estadísticas	5
Documentación	25
Puntos extra	30

Notas

- El proyecto programado es en **parejas** y se debe entregar a más tardar el martes 14 de junio 2022 hasta las 13:00 en forma electrónica. Cada grupo debe incluir un archivo *readme.txt* con la versión de Python (≥ 3.3) a utilizar para la revisión y alguna otra indicación que se considere importante. Se debe subir al tec digital.
- Cualquier duda, omisión o contradicción en la especificación se debe aclarar con el profesor y se difundirá por el tec digital y el grupo de Telegram.
- Se debe de enviar la documentación (archivo .pdf y realizado en latex) ésta debe contar con lo siguiente: Introducción, conclusiones, recomendaciones, diagramas de módulos, plan de pruebas, resultados y literatura o fuentes consultadas.

Además, debe adjuntar un programa que genere la auto-documentación con el método *print* de Python para los principales módulos de su programa. El resto de la auto-documentación debe estar disponible en la defensa.

- El código debe estar suficientemente documentado de tal forma que el grupo se pueda orientar en él fácilmente durante la defensa. Debe contar con auto documentación interna para ser generada mediante el método *print* de Python.
- El proyecto se debe defender previa cita con el profesor. El profesor publicará unas fechas de defensa. El grupo debe inscribirse para defender el proyecto, de no asistir a la defensa, únicamente obtendrá la nota correspondiente a la documentación del proyecto.
- Cualquier clase de copia de código será sancionada de acuerdo con el reglamento vigente y se llevará hasta la consecuencia de amonestación con carta al expediente. Código adoptado para el manejo de interfaz debe especificarse claramente la fuente y reconocer los créditos. Está prohibida la copia de código que involucre la solución lógica general del algoritmo.
- Si en la defensa no demuestra su autoría con el dominio propio de esa calidad solo se le otorgarán los puntos correspondientes a los obtenidos en la documentación.