



Programación II

Proyecto final Battleship

Sección: 2.03

Integrantes:

Código	Apellidos y Nombres	% Participación
	Pérez Bonany, Martín	100
202110269	Picon Abucci, Francesca Catterina	100

Profesor: Estanislao Contreras Chávez

3/12/2021

LIMA - PERÚ

Índice de contenidos

Índice de contenidos	2
	3
1. Fundamento teórico	4
1.1. Concepto 1	4
1.2. Concepto 2	4
2. Metodología y Desarrollo	5
2.1. Diagrama de Clases	5
2.2. Descripción de las Clases	5
3. Conclusiones	6
Bibliografía	7

1. Fundamento teórico

1.1. POO

La Programación Orientada a Objetos (POO) es un paradigma de programación, es decir, un modelo o un estilo de programación que nos da unas guías sobre cómo trabajar con él. Se basa en el concepto de clases y objetos.

1.1.1. Clase

Una clase es la declaración de un tipo de objeto. Las clases son similares a los tipos abstractos de datos y equivalen a modelos que describen cómo se construyen ciertos tipos de objetos.

1.1.2. Objeto

Un objeto en POO representa alguna entidad de la vida real, es decir, alguno de los objetos que pertenecen al negocio con que estamos trabajando o al problema con el que nos estamos enfrentando, y con los que podemos interactuar.

1.2. Herencia

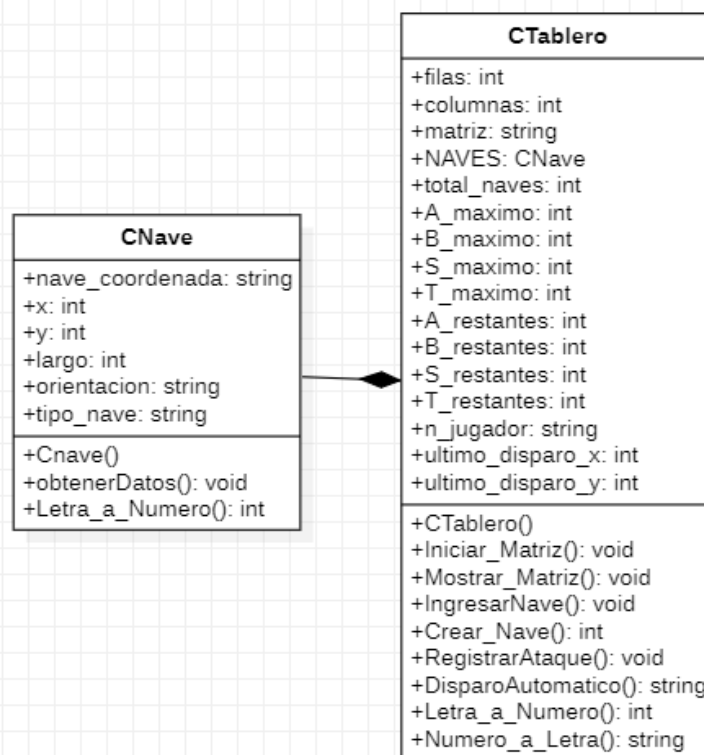
La herencia permite que se puedan definir nuevas clases basadas de unas ya existentes a fin de reutilizar el código, generando así una jerarquía de clases dentro de una aplicación. Si una clase deriva de otra, esta hereda sus atributos y métodos y puede añadir nuevos atributos, métodos o redefinir los heredados.

2. Metodología y Desarrollo

Después de haber creado las clases, en el main.cpp pusimos algunos datos básicos como el tamaño del tablero, la ruta del juego o su estado. Después usamos funciones para ingresar y crear las naves de manera aleatoria y así posicionarlas en el tablero; creamos también algunas funciones de guardado que tienen la función de crear, buscar y guardar archivos.

Una vez finalizada la declaración de funciones, llegamos a la función main donde vamos a ingresar el nombre del equipo y el número de jugador al cual pertenece, posterior a eso inicia el estado de juego para jugar y empezamos a usar las funciones. Vamos captando los tokens y en base a eso decidimos si toca atacar, esperar o ya ganamos/perdimos, y en cada uno de estos eventos prácticamente llamamos a su función únicamente. Todo eso se va repitiendo en un while hasta que detecte 100 jugadas o que alguien ha ganado, y con eso finaliza el juego.

2.1. Diagrama de Clases



2.2. Descripción de las Clases

Usamos dos clases para el proyecto, una para las naves y otra para los tableros. En el caso de las naves, creamos una clase con su posición, orientación, tipo y tamaño y después unas funciones para obtener sus datos mediante getters & setters. En el caso del tablero, hicimos la clase con sus filas, columnas y matriz para crear el tablero y después un vector con las naves así como la cantidad de cada tipo de estas restantes en el tablero. Asimismo hicimos unas funciones para primero iniciar la matriz e imprimirla en la terminal. Después funciones para mostrar el tablero, otra para crear la nave ,otra para registrar en el tablero las X e Y's cuando sucede algún evento, y por último una función de Disparo Aleatorio que es la que va a manejar al bot.

3. Conclusiones

Este proyecto presenta una alta dificultad para nosotros debido al hecho de usar un tipo básico de lo que sería Inteligencia Artificial, tomando en consideración que el programa debía de jugar solo y tener la mejor estrategia posible. Investigando modelos y mejores estrategias decidimos emplear el método hide & seek, que se basa en que se lanzan los disparos de manera aleatoria, y si le da al objetivo procede a disparar arriba o a los lados, pero si no le da al objeto sigue disparando de manera aleatoria.

Aprendimos de este trabajo que era mejor trabajar juntos para tener un mejor entendimiento de lo que estaba ocurriendo en nuestro código y de esa manera frente a cualquier inconveniente podríamos hacer una lluvia de ideas para solucionarlo.

Bibliografía

Miriam Martínez Canelo. (2020, November 2). *¿Qué es la Programación Orientada a Objetos?* Profile Software Services.
<https://profile.es/blog/que-es-la-programacion-orientada-a-objetos/>

¿Qué es la Herencia en programación orientada a objetos? (2019, June 5). IfgeekthenNTTdata.
<https://ifgeekthen.nttdata.com/es/herencia-en-programacion-orientada-objetos>