

19 de marzo de 2024

Cristian valentin anghel

.

Programació orientada a objectes

Mascota virtual

**Índex**

[**Abstract** 2](#_Toc161741885)

[**Diagrama de clase** 2](#_Toc161741886)

[**Código** 2](#_Toc161741887)

[**Tamagotchi.cpp** 2](#_Toc161741888)

[**Farm.cpp** 4](#_Toc161741889)

[**Farm.h** 5](#_Toc161741890)

[**Orc.h** 6](#_Toc161741891)

[**Mascota.cpp** 6](#_Toc161741892)

[**Orc.cpp** 8](#_Toc161741893)

[**OrcBoss.cpp** 8](#_Toc161741894)

[**Mascota.h** 9](#_Toc161741895)

[**Farm.h** 9](#_Toc161741896)

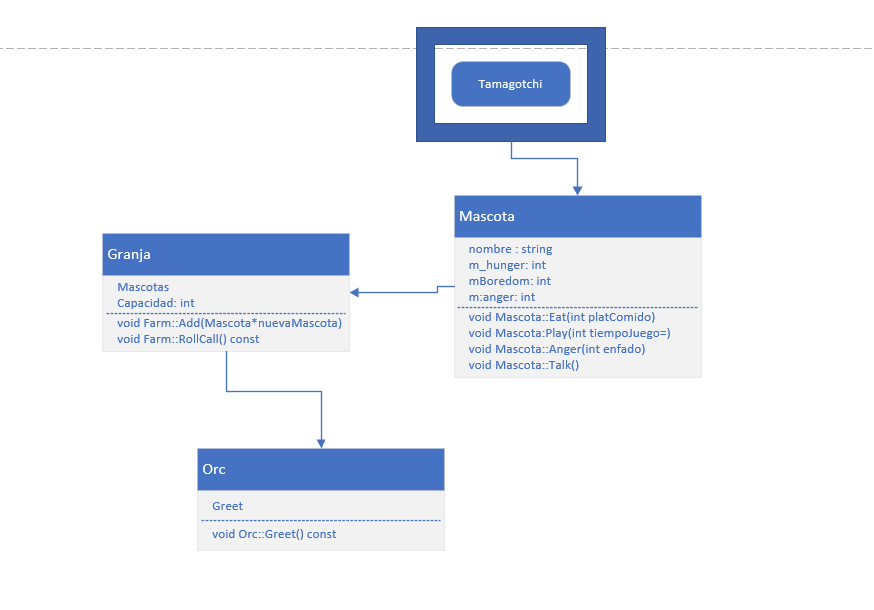
[**OrcBoss.h** 10](#_Toc161741897)

[**Display** 10](#_Toc161741898)

# **Abstract**

On this Project, we had to make create a farm which you had the chance to play with them, to feed them, talk and anger them. Also you had the opportunity to create new animals, list them and see their stats. My farm had a limit of 3 animals. Lastly you could talk with an orcus.

# **Diagrama de clase**

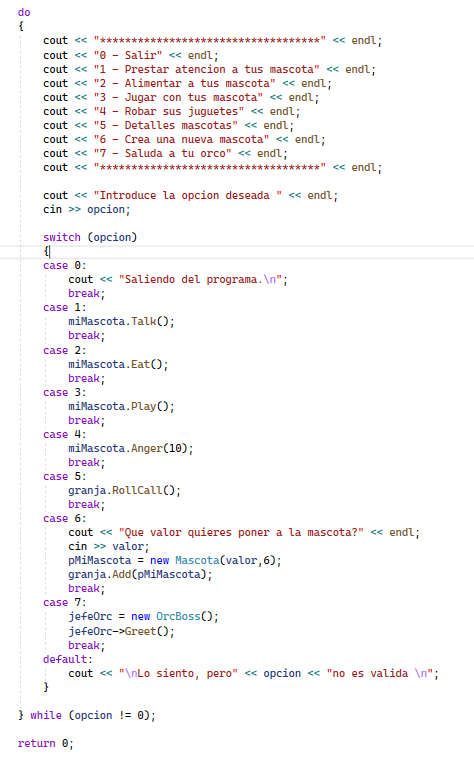


# **Código**

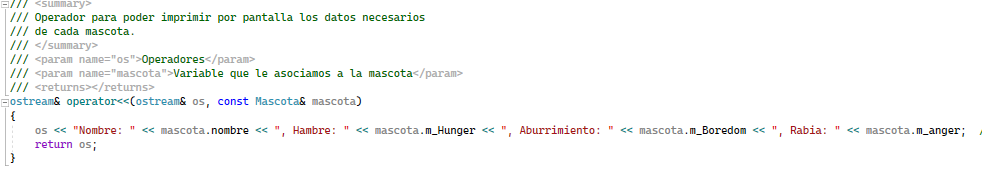
## **Tamagotchi.cpp**



Este es el main, donde incluimos los diferentes .h en las librerías y el iostream para poder hacer las entradas y salidas del programa. Después declararemos los operadores con las constantes de Mascota. También tenemos el ascci bonito y crearemos las variables de miMascota y pMimascota para declararlas en el switch case más tarde. Por ultimo crearemos el array de la granja con un límite de 3 mascota, el integer para la opción del menú con el valor de la mascota y el OrcBoss.

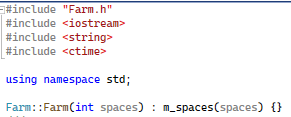


Aquí seguiremos con un do while, donde tendremos el menú principal dentro y los diferentes swithc case que tienen dentro. Todas llaman a funciones externas.

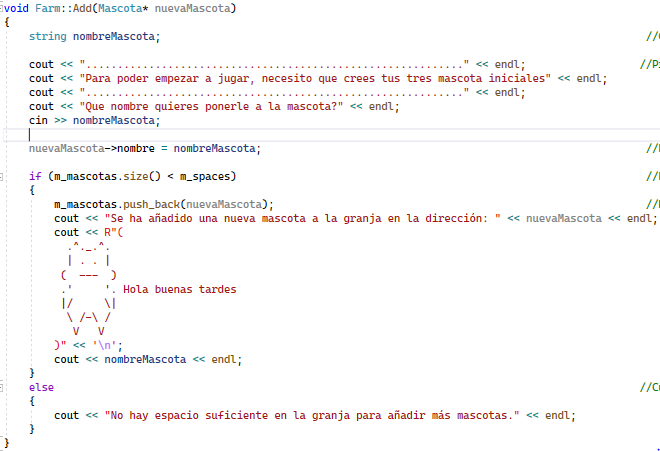


Aquí tendremos los operadores, donde nos saldrá por pantalla el nombre, el hambre, el aburrimiento y la rabia de cada mascota.

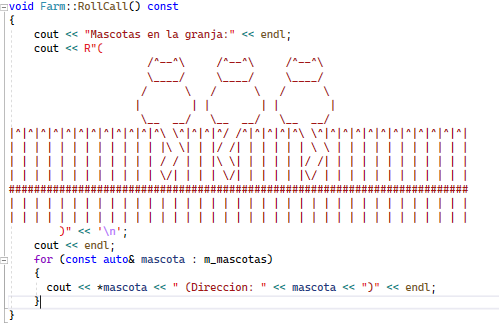
## **Farm.cpp**



En el farm.cpp tendremos estas librerías necesarias, menos ctime, crearemos un constructor de la granja con los espacios dentro



Después tenemos la función principal para crear una nueva mascota, donde haremos un string para poder coger el nombre de la mascota que crearemos, le pregunta al usuario el nombre. Tendremos un bucle if else, donde hará el pushback y un ascci bonito. Cuando la granja llegue a una cantidad máxima de 3 mascota, nos podrá este otro mensaje donde nos dirá que no hay más espacio para poder añadir más macotas en la granja.



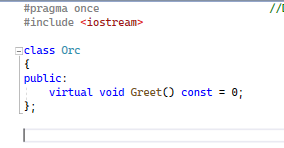
Esta será la función principal para poder enseñar las mascotas de la granja que tenemos, un simple bucle de for con las constantes de las mascotas y dentro un cout para enseñar la dirección de los punteros.

## **Farm.h**



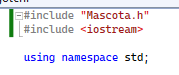
Para la granja, tendremos que hacerle la clase con sus funciones públicas y privadas. Pondremos cada cosa que puede llegar a hacer la mascota

## **Orc.h**

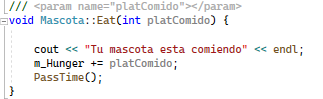


Crearemos una clase llamada Orc con un método virtual llamado Greet, una función constante y será implementada en cualquiera clase que herede Orc.

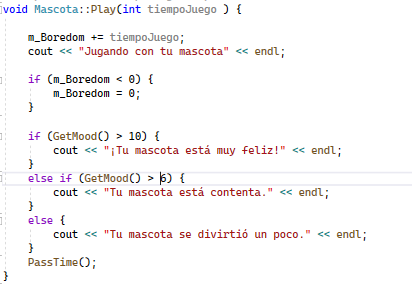
## **Mascota.cpp**



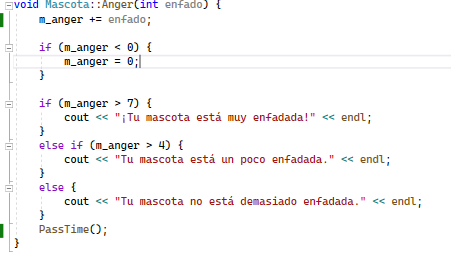
En mascota incluyremos el .h



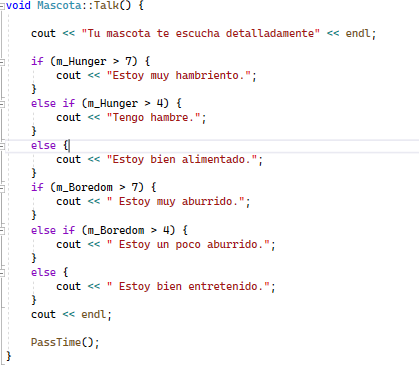
La función para poder comer, incluiremos como un integer el platocomido, le haremos un += para poder ir subiéndole el hambre de uno en uno y el passtime para que el hambre vaya variando con el tiempo jugando.



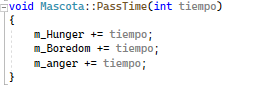
Esta será la función para poder jugar con la mascota donde tendremos el += para que vaya sumando y luego un if else donde nos saldrá un texto diferente cuando los valores vayan sumando. Y por último el passtime para que el boredom vaya variando con el tiempo jugando.



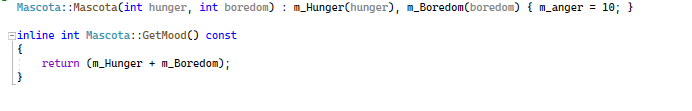
La función de Rabia, tendrá el += para que vaya sumando, if else para que depende de qué valor tenga nos ponga una cosa u otra y el passtime para que el enfado vaya variando con el tiempo jugando.



Función para poder hablar con la mascota y nos diga el estado anímico.

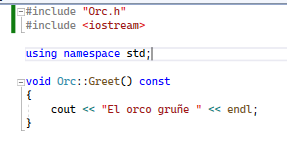


Función para que vaya pasando el tiempo y los valores vaya variando automáticamente jugando el juego



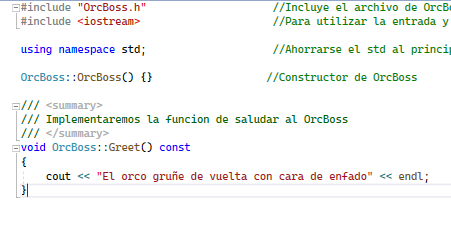
Creamos el constructor de Mascota, donde pondremos los integers de las funciones que hará la mascota y establezco un valor inicial de 10 de rabia. La función de Getmood, nos devolverá la suma de los miembros de m\_Hunger y m\_Boredom.

## **Orc.cpp**



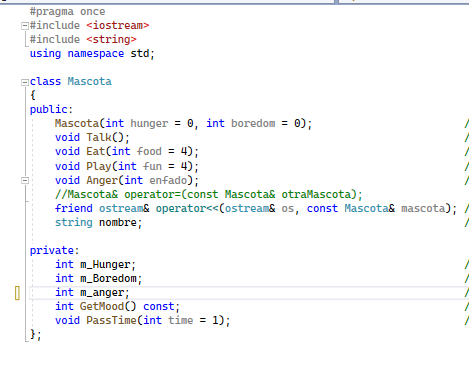
Implementacion de un metod llamado Greet para la clase Orc donde imprimirá El orco gruñe

## **OrcBoss.cpp**



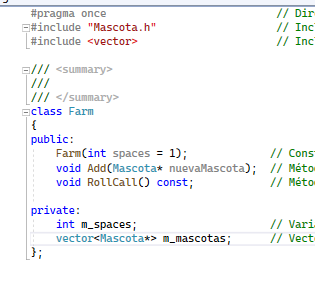
Define la clase OrcBoss y su método Greet que imprime el mensaje querido

## **Mascota.h**



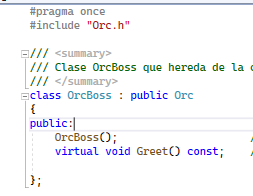
Aquí tendremos las clase Mascota con el public y privado, con sus respectivas variables

## **Farm.h**



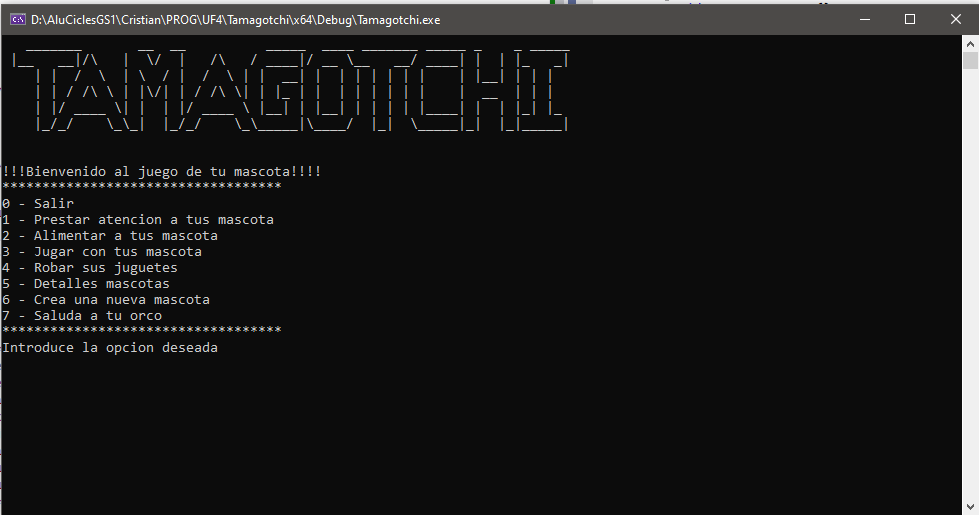
Dentro de la granja, tendremos las clases públicas y privadas con las funciones de añadir y nombrar mascotas.

## **OrcBoss.h**

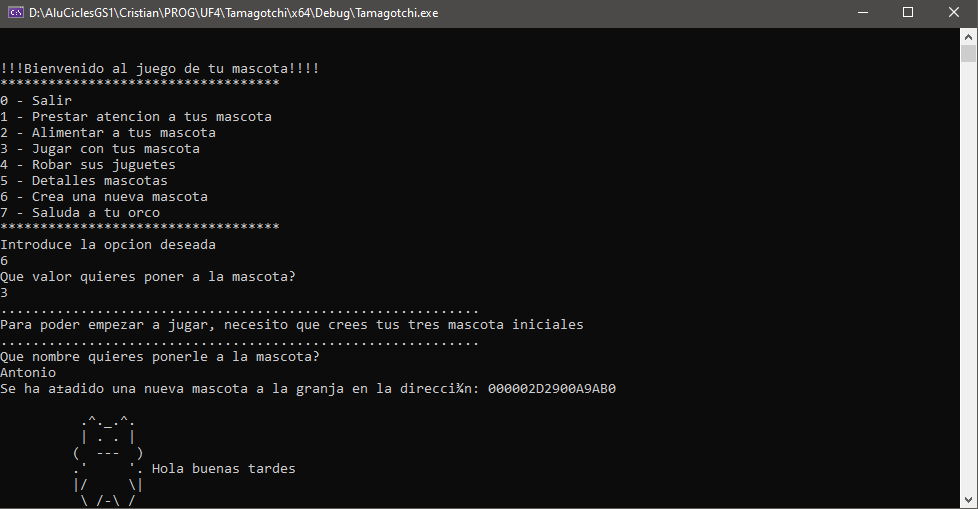


Define la clase OrcBoss que hereda de la clase Orc y declara un constructor y una función Greet

# **Display**



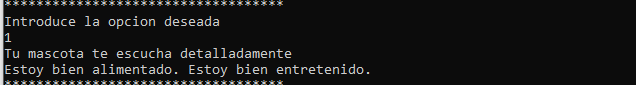
Menú al iniciar



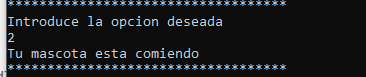
Añadir mascota



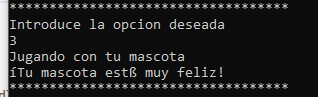
Ver mascotas en la granja



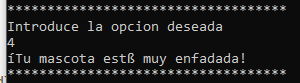
Prestar atención



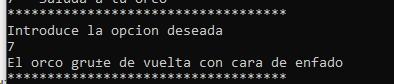
Dar de comer



Jugar con tu mascota



Enfadar a tu mascota



Saludar al orco