Proyecto final

Fundamentos de Computación 1

Carpeta de Campo

Documentación

Borges Marcelo Agustin

Vega Hernández Carlos David

Fecha: junio 2023

Descripción general del proyecto:

El proyecto “Mascota Virtual” es un proyecto final desarrollado en el contexto de la asignatura de Fundamentos de la Computación 1. El objetivo principal del proyecto es crear una mascota virtual similar a un Tamagotchi, implementada en lenguaje C.

Para lograr este objetivo, se han utilizado diversas herramientas de programación, tales como estructuras, archivos, arrays y funciones. Estas herramientas permiten mantener un código organizado y modular, facilitando su comprensión y mantenimiento.

Además, se ha dado especial énfasis a la implementación de árboles binarios, uno de los temas principales de la asignatura. La integración de árboles binarios en el código del proyecto permite explorar y aplicar los conceptos teóricos aprendidos durante la cursada.

Características del proyecto:

1. Tema y Objetivo principal:

el proyecto “Mascota virtual” se centra en la creación de una mascota virtual interactiva, inspirada en el concepto del popular juego Tamagotchi. El objetivo es simular la responsabilidad de cuidar y mantener a una mascota virtual, proporcionando una experiencia de juego entretenida y educativa.

1. Barras de estado:

Se deben implementar barras de estado, como la barra de hambre, salud y ánimo de la mascota. Estas barras deben estar relacionadas entre sí, de modo que el nivel de hambre afecte la salud de la mascota, y el ánimo pueda influir en la interacción con minijuegos. El estado de la mascota se reflejará visualmente mediante las barras de estado, lo que permitirá al usuario evaluar y satisfacer las necesidades de la mascota.

1. Datos de usuario y niveles de dificultad:

El proyecto debe permitir a los usuarios ingresar los datos de la mascota, como también, el nombre de usuario.

Estos datos se utilizarán para personalizar la experiencia del juego. Además, se puede optar por niveles de dificultad, que varían la mortalidad de la mascota entre otras características, proporcionando desafíos adecuados a la preferencia y habilidad del usuario.

1. Avatares en ASCII:

El proyecto incluye avatares en ASCII para representar visualmente a la mascota virtual.

Estos avatares en ASCII agregan un componente visual atractivo y permiten al usuario conectarse emocionalmente con la mascota.

1. Menú interactivo y minijuegos:

Mediante la implementación de un menú interactivo, se proporciona al usuario diferentes opciones para interactuar con la mascota virtual. Estas opciones pueden incluir alimentar a la mascota, jugar con ella, tener un control de su salud, entre otros. Además, se incluyen minijuegos que permiten al usuario interactuar y entretenerse junto a la mascota virtual, afectando su estado de ánimo y brindando recompensas.

1. Sistema de alimentación y tienda de alimentos:

El proyecto incluye un sistema de alimentación variado que permite al usuario seleccionar y proporcionar alimentos a la mascota. Además de una tienda de alimentos donde el usuario puede adquirir diferentes tipos de comida para la mascota. Esto agrega un componente de gestión y toma de decisiones al juego.

1. Archivos externos:

Para ampliar la funcionalidad del proyecto, se utilizan archivos externos, asimismo su implementación permite la capacidad de leer y escribir archivos de texto (.txt) y archivos de valores separados por comas (.csv) para almacenar y recuperar datos relevantes del juego, como registros de puntación, datos del usuario, o información de la tienda de alimentos.

Implementación del tema relacionado a la materia:

Árboles Binarios: Inserción/Búsqueda/Eliminación.

En nuestro proyecto de mascota virtual en C, hemos decidido utilizar árboles binarios para gestionar el inventario de alimentos y suministros de nuestra mascota. Esta implementación nos permitirá organizar y acceder eficientemente a los elementos del inventario, además de facilitar la inserción, búsqueda y eliminación de los mismos. A continuación, explicaré los detalles de esta implementación en base a la información proporcionada.

1. Estructura de Nodo del Árbol Binario:

Para representar un nodo en nuestro árbol binario, utilizaremos una estructura que contendrá la información relevante de cada artículo del inventario. Los campos de esta estructura incluyen:

* ‘itemName’: El nombre del artículo.
* ‘quantity’: La cantidad disponible del artículo.
* ´price´: El precio del artículo.
* ‘foodType’: el tipo de comida.
* ‘left’: puntero a subárbol izquierdo.
* ‘right’: puntero a subárbol derecho.

1. La función **‘crearNodo’** se encarga de crear un nuevo nodo del árbol.

Toma valores del artículo y crea un nodo asignando memoria y copiando los valores proporcionados.

1. La función **‘insertarNodo’** se utiliza para insertar un nuevo nodo en el árbol. Toma como argumentos la raíz del árbol, los valores del artículo y realiza la inserción en la posición adecuada del árbol según el nombre del artículo. Si el artículo ya existe en el árbol, se incrementa la cantidad en lugar de insertar un nuevo nodo.
2. La función **‘buscarNodo’** se utiliza para buscar un nodo en el árbol según el nombre del artículo. Toma como argumentos la raíz del árbol y el nombre del artículo, y realiza una búsqueda recursiva en el árbol hasta encontrar el nodo correspondiente o llegar a una hoja.
3. La función **‘mostrarInventario’** se utiliza para mostrar el inventario en orden. Recorre el árbol en orden recursivamente, mostrando los detalles de cada artículo en el nodo actual.
4. La función **‘encontrarMinimo’** es una función auxiliar que encuentra el nodo con el valor mínimo en el subárbol. Se utiliza en la función **‘eliminarNodo’** para encontrar el sucesor inmediato de un nodo cuando se elimina.
5. La función **‘eliminarNodo’** se utiliza para eliminar un nodo del árbol según el nombre del artículo. Realiza una búsqueda recursiva para encontrar el nodo a eliminar y realiza la eliminación según los casos posibles (nodo sin hijos, nodo con un solo hijo o nodo con dos hijos). También actualiza la cantidad de dinero (**‘wallet**) al eliminar un nodo.

En la función ‘**alimentation’** se utiliza un bucle **‘do-while’** para mostrar un menú de opciones al usuario. Las opciones incluyen comprar un producto, mostrar el inventario, comer un producto y salir del programa.

En la opción “Comer producto”, se muestra el inventario actual y se ingresa el nombre del producto a comer. Se busca el nodo correspondiente y, si se encuentra, se actualiza la cantidad y se elimina si la cantidad llega a 0. Además, se actualiza el **‘monedero’** al agregar el precio del producto.

Arquitectura del proyecto