

Proyecto 1, Tron

1st Sergio Álvarez Chanto, 2024134332
CE, Estructuras de Datos 1, 2024

Link github: <https://github.com/DAniels0100/TronGame>

I. TABLA DE CONTENIDO

- 1) Tabla de contenido
- 2) Introducción
- 3) Descripción del problema
- 4) Descripción de las soluciones
 - a) Solucion a la interfaz grafica
 - b) Solucion a la grid de nodos
 - c) Solucion a la funcionalidad de la moto
 - d) Solucion a las otras motos
- 5) Diagrama de clases

II. INTRODUCTION

En este proyecto se busca crear un juego de carrera de motos de luz basado en la película "Tron", esto mediante la utilización de estructuras de datos lineales, como linked list, stacks y queues.

El juego al basarse en la película "Tron" debe respetar ciertos comportamientos observables en esta, por ejemplo, cuando una moto colisiona con la estela de otra, esta explota. Además, en este proyecto se presentan poderes e ítems que el jugador puede usar a su voluntad.

III. DESCRIPCION DEL PROBLEMA

El problema planteado es el lograr crear un juego inspirado en la película de "Tron", usando principalmente estructuras de datos lineales. El jugador debe poder manejarse en un grid de nodos de una linked list, donde aparecen distintos items y poderes mientras esquiva otras motos (en este caso bots). Si una moto colisiona con la estela de otra, esta desaparece soltando los objetos que poseia en el momento de la colision. Además, el jugador debe poder reorganizar sus poderes para usarlos a conveniencia.

IV. DESCRIPCION DE LAS SOLUCIONES

A. Solucion a la interfaz grafica

Para el desarrollo del juego se utilizo c shrap con Windows forms como interfaz grafica. El juego se corre sobre un picture box, donde estan las distintas motos y poderes.

B. Solucion a la grid de nodos

Para representar la superficie sobre la cual se desplazan las motos, se creo una grid de nodos que se crea en "zigzag" para que los nodos se vayan conectando con los nodos que se encuentran "arriba" y "abajo". Otra forma de lograr esto, seria con una matriz bidimensional.

En esta solucion el principal problema encontrado fue el como

ir conectando los nodos, ya que en un array bidimensional se pueden modelar estas dos dimensiones facilmente, mientras que con una sola linked list es unidimensional. Aportar que los nodos tenian que tener cuatro apuntadores para conocer todos los nodos vecinos.

C. Solucion a la funcionalidad de la moto

Las motos de luz debian responder a cinco posibles eventos, estos siendo desplazarse, colisionar, explotar, recolectar poderes e items, usar y ser afectados por los poderes o items. Para el desplazarse, colisionar y explotar al colisionar, se usaron las cordenadas de los segmentos de las motos. El funcionamiento del desplazamiento es que los nodos que no son el "head" siguen al nodo "head". En la colision, se verifica si el nodo "head" colisiona con cualquier otro nodo, al igual la recoleccion verifica si el nodo "head" colisiono con un poder o item. Las motos explotan al haber colisionado con otra moto. Por otra parte, la interaccion entre las motos y los poderes, para los poderes se utilizaron stacks para almacenar los poderes que el jugador obtiene y con una manipulacion con dos stacks se realizan las reorganizaciones de poderes. Por el lado de los items, estos actuan instantaneamente en el jugador, no necesitan que este los active. Los items se guardan en una queue de cuales son las modificaciones a hacerle a la moto.

D. Solucion a las otras motos

Las otras motos, contra las cuales el jugador se enfrenta son bots que imitan el comportamiento. La solucion aportada fue copiar el funcionamiento de desplazamiento e interaccion con los objetos del mapa. Los bots se desplazan aleatoriamente con un cambio de direccion cada pocos segundos. Estos bots tambien funcionan bajo la misma logica de colision, por lo cual, al colisionar desaparecen y sueltan sus objetos.

V. DIAGRAMA DE CLASES

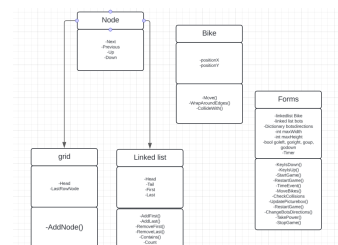


Fig. 1. Diagrama de clases UML

VI. REFERENCIAS

- [1]Moo ICT, "Csharp Tutorial - Create a Classic Snakes Game in Visual Studio with Windows Forms [UPDATED]," YouTube, Jun. 25, 2021. [Online]. Available: <https://www.youtube.com/watch?v=TzaCn1ZPaIt=1588s>. [].
- [2]Microsoft, "LinkedList<T> Class," Microsoft Learn, Aug. 18, 2023. [Online]. Available: <https://learn.microsoft.com/en-us/dotnet/api/system.collections.generic.linkedlist-1?view=net-8.0>.
- [3]Microsoft, "Stack Class," Microsoft Learn, Aug. 18, 2023. [Online]. Available: <https://learn.microsoft.com/es-es/dotnet/api/system.collections.stack?view=net-8.0>.
- [4]Microsoft, "Queue Class," Microsoft Learn, Aug. 18, 2023. [Online]. Available: <https://learn.microsoft.com/es-es/dotnet/api/system.collections.queue?view=net-8.0>.