

**Trabajo:**

Hito 1

**Curso:**

Complejidad Algorítmica

**Sección:**

WX61

**Integrantes:**

Luis Alejandro Galindo Honores - u201516808

Omar Chavez Olivera – u201523075

Mauricio Rivas Arroyo – u201511597

**Junio, 2018**

1. Introducción

a. Problema

La agencia espacial “UPC Star” ha detectado un nuevo sistema solar en un lugar recóndito del universo. Según estudios preliminares, hay N planetas en ese sistema solar y afortunadamente los planetas siempre están alineados. El director de “UPC Star” cree que el nuevo sistema solar alberga un planeta habitable para la humanidad y le ha asignado a usted la tarea de buscar la mejor ruta con su nave para llegar a este planeta. Asi mismo, también deberá tener en cuenta que habrán individuos queriendo destruir su nave, por lo que deberá evadirlos para completar su misión. Además, usted debe recolectar un material ubicado aleatoriamente en el espacio para poder cargar el combustible de su nave.

b. Objetivos

* Aplicar algoritmos de búsqueda para la IA de los enemigos que habitan en el espacio.
* Aplicar algoritmos de optimización, asi como tambien de maximo camino, para definir el mapa y que el piloto de la nave elija un camino.
* Knapsack se utilizará para la IA del misil teledirigido de la nave jugador. Determinará el mejor caso (Ítem, costo, peso - ID, Materiales, vida) para saber a qué nave o naves impactar.

2. Marco conceptual

Lenguaje C++

Algoritmo de Floyd-Warshall

En informática, el algoritmo de Floyd-Warshall, descrito en 1959 por Bernard Roy, es un algoritmo de análisis sobre grafos para encontrar el camino mínimo en grafos dirigidos ponderados. El algoritmo encuentra el camino entre todos los pares de vértices en una única ejecución. El algoritmo de Floyd-Warshall es un ejemplo de programación dinámica.  
  
Muchos problemas de la vida cotidiana se pueden expresar e incluso resolver en forma de grafo. Existen algoritmos que encuentran distintos tipos de soluciones, tanto booleanas como de eficiencia. El grafo se representa en una tabla (matriz) que se conoce como “matriz de adyacencia” y representa si existe una unión entre dos nodos (boolean).

Algoritmo de Bellman-Ford

El algoritmo de Bellman-Ford (algoritmo de Bell-End-Ford) genera el camino más corto en un grafo dirigido ponderado (en el que el peso de alguna de las aristas puede ser negativo). El algoritmo de Dijkstra resuelve este mismo problema en un tiempo menor, pero requiere que los pesos de las aristas no sean negativos, salvo que el grafo sea dirigido y sin ciclos. Por lo que el Algoritmo Bellman-Ford normalmente se utiliza cuando hay aristas con peso negativo. Este algoritmo fue desarrollado por Richard Bellman, Samuel End y Lester Ford.

Algoritmo de Dijkstra

El algoritmo de Dijkstra, también llamado algoritmo de caminos mínimos, es un algoritmo para la determinación del camino más corto, dado un vértice origen, hacia el resto de los vértices en un grafo que tiene pesos en cada arista. Su nombre alude a Edsger Dijkstra, científico de la computación de los Países Bajos que lo describió por primera vez en 1959.  
  
La idea subyacente en este algoritmo consiste en ir explorando todos los caminos más cortos que parten del vértice origen y que llevan a todos los demás vértices; cuando se obtiene el camino más corto desde el vértice origen hasta el resto de los vértices que componen el grafo, el algoritmo se detiene. Se trata de una especialización de la búsqueda de costo uniforme y, como tal, no funciona en grafos con aristas de coste negativo (al elegir siempre el nodo con distancia menor, pueden quedar excluidos de la búsqueda nodos que en próximas iteraciones bajarían el costo general del camino al pasar por una arista con costo negativo).  
  
Una de sus aplicaciones más importantes reside en el campo de la telemática. Gracias a él, es posible resolver grafos con muchos nodos, lo que sería muy complicado resolver sin dicho algoritmo, encontrando así las rutas más cortas entre un origen y todos los destinos en una red.

3. Metodología

La metodología consiste en que el usuario elegirá el camino más corto a partir de varios caminos proporcionados, esto le permitirá ganar cierta cantidad de puntos al acertar, mientras que si falla se les restará. Al elegir un camino, se pasará al siguiente nivel ,indistintamente del resultado, y se cambiarán automáticamente los caminos, dando paso a una nueva elección.

4. Ejecución

a.Tipos de datos abstractos

* Un grafo de recorrido.
* Un grafo de pesos.
* Una cola o pila donde se encuentran almacenados las aristas con sus pesos.
* Un vector bidimensional.

b. Algoritmos

* Algoritmo de Floyd-Warshall
* Algoritmos de Bellman Ford
* Algoritmo Union Find-Disjoinsets con path compression y rank

5. Gráfica

Se utilizan caracteres con determinados colores para identificar a el/los personaje/s y elementos de interacción.

6. Jugabilidad

El usuario solo se podrá mover, no contará con poderes, cuando llegue a un punto específico el mapa se altera generando rutas aleatorias.

7. Reglas

Cada vez que el usuario escoja la ruta con menor peso se le otorgará 10 puntos, si escoge otra ruta solo se le otorgará 5 puntos. El jugador podrá sacrificar 20 puntos para descartar una ruta del nivel.