Proyecto Algoritmos y estructuras de datos

Betancourt Ernesto(A00049172), Vera Juan David(A00293816)

**Introducción:**

Se va a realizar un juego cuyo objetivo es conseguir 100 monedas atreves de diferentes pantallas con minijuegos de diferentes dificultades. Para moverse por el mapa hacia las pantallas hay un costo de monedas que varía según el camino que quiera tomar el jugador. El juego va a recomendar la pantalla que para llegar a ella tenga el menor costo desde donde se encuentra el jugador, además se mostraran caminos hacia pantallas con una dificultad menor a la actual.

**Definición del problema:**

Generar un mapa basado en un grafo en donde el jugador con una cantidad de monedas pueda moverse por los diferentes caminos de este y llegar a los diferentes nodos los cuales contienen juegos para que el jugador gane más monedas.

**Requerimientos:**

|  |  |
| --- | --- |
| Requerimiento funcional 1 | |
| Nombre: | calcular niveles |
| Descripción: | Se calcularán los niveles del grafo desde donde haya iniciado el jugador |
| Entrada: | Índice del nodo |
| Salida: | Una lista de los niveles de todos los nodos |

|  |  |
| --- | --- |
| Requerimiento funcional 2 | |
| Nombre: | Asignar niveles |
| Descripción: | Asigna un juego con la dificultad asociada al nodo |
| Entrada: | Una lista de los niveles de los nodos |
| Salida: |  |

|  |  |
| --- | --- |
| Requerimiento funcional 3 | |
| Nombre: | menos costoso |
| Descripción: | Busca el camino menos costoso desde donde está parado el jugador |
| Entrada: | Posición del jugador |
| Salida: | Posición del nodo menos costoso |

|  |  |
| --- | --- |
| Requerimiento funcional 4 | |
| Nombre: | Menor dificultad |
| Descripción: | Encuentra todos los nodos con dificultad menor al nodo actual |
| Entrada: | Posición del jugador |
| Salida: | Lista de nodos |

Recopilación de información necesaria:

* Los niveles por lo general se presentan como 2D continuo o espacio 3D, pero en juegos como *Super Mario Bros. 3*, los espacios pueden estar separados, con alguna forma de tele transportación entre ellos. (Fundación Wikimedia, Inc., 2018)
* Un área se usa para definir un nivel que, literalmente, coexiste físicamente entre múltiples niveles, en el que el jugador puede progresar de un “nivel” a otro simplemente usando la física del juego. A pesar de que coexisten espacialmente, cada área presenta sus propios temas, recompensas y desafíos. (Fundación Wikimedia, Inc., 2018)
* El jugador se desplaza durante el juego a través de dos pantallas: un mapa que muestra el mundo y sus respectivos niveles, y el campo de juego del nivel. El mapa del mundo muestra una representación aérea y tiene varios caminos que llevan de la entrada del mundo a un castillo. Las rutas conectan los niveles, fortalezas y otros íconos del mapa y permitirá a los jugadores a tomar rutas diferentes para llegar a la meta de cada mundo. (Fundación Wikimedia, Inc., 2018)
* En matemáticas y ciencias de la computación, un grafo es un conjunto de objetos llamados vértices o nodos unidos por enlaces llamados aristas o arcos, que permiten representar relaciones binarias entre elementos de un conjunto. (Fundación Wikimedia, Inc., 2018)
* En Ciencias de la Computación, Búsqueda en anchura (en inglés *BFS* - *Breadth First Search*) es un algoritmo de búsqueda no informada utilizado para recorrer o buscar elementos en un grafo (usado frecuentemente sobre árboles). Intuitivamente, se comienza en la raíz (eligiendo algún nodo como elemento raíz en el caso de un grafo) y se exploran todos los vecinos de este nodo. A continuación para cada uno de los vecinos se exploran sus respectivos vecinos adyacentes, y así hasta que se recorra todo el árbol. (Fundación Wikimedia, Inc., 2018)
* 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. (Fundación Wikimedia, Inc., 2018)
* 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. (Fundación Wikimedia, Inc., 2018)

TAD del grafo:

|  |
| --- |
| TAD Map |
| Resultado de imagen para weighted graph in data structureGrafo no direccionado, ponderado, no cíclico, conexo o no conexo |
| inv: no es un bosque. |
| Operaciones Primitivas:  getGame: Game X Map: Game setGame: Game X Map:  getPosition: Map X Map: Integer  getSize: Map X Map: Integer  getNodesPositions: Map X Map: lista de nodos |
| |  | | --- | | getGame()  "retorna el juego que el jugador tiene que superar"  {pre: los juegos están cargados} {post: se tendrán las monedas que haya conseguido} | | setGame(entero}  "asigna un juego al nodo"  {pre: hay juegos predefinidos para selección} {post: el nodo tendrá un juego asignado} | | getPosition()  “devuelve la position del nodo actual”  {pre: el jugador ha terminado la partida}  {pos: da la posición} | | getNodesPositions:  “Devuelve una lista con los nodos creados”  {pre: hay un grafo creado}  {pos: se ha retornado la lista} | |

<https://es.wikipedia.org/wiki/Algoritmo_de_Dijkstra>

<https://es.wikipedia.org/wiki/Búsqueda_en_anchura>

<https://es.wikipedia.org/wiki/Nivel_(videojuegos)>

<https://es.wikipedia.org/wiki/Super_Mario_Bros._3>

<https://es.wikipedia.org/wiki/Grafo>

<https://es.wikipedia.org/wiki/Algoritmo_de_Floyd-Warshall>