

Especificación de requerimientos

| | |
|-------------------------------|---|
| Institución | Universidad Icesi |
| Nombre | Nicholas Solis |
| Contexto | <p>Usted debe desarrollar (analizar, diseñar e implementar) un juego de uno o más jugadores, el cual pueda ser modelado utilizando grafos, con un mínimo de 50 vértices y 50 aristas, y tal que para su solución sea posible aplicar al menos dos (2) de los algoritmos de grafos que se estudiarán durante el curso: Recorridos sobre Grafos, Caminos de Peso Mínimo, Árbol de Recubrimiento Mínimo.</p> |
| Requerimientos Funcionales | <p>RF1- El sistema debe permitir usar dos implementaciones diferentes del TAD Grafo.</p> <p>RF2- El sistema debe permitir cambiar entre las implementaciones del grafo en cualquier momento.</p> <p>RF3- El sistema debe manejar un grafo con mínimo 50 vértices y 50 aristas.</p> <p>RF4- El sistema debe ejecutar al menos dos algoritmos de grafos (BFS, DFS, Dijkstra, Floyd-Warshall, Prim o Kruskal).</p> <p>RF5- El juego debe permitir la participación de uno o más jugadores.</p> <p>RF6- El sistema debe contar con una interfaz gráfica para interactuar con el grafo y el juego.</p> |
| Requerimientos no funcionales | <p>RnF1- La interfaz gráfica debe ser intuitiva y fácil de usar.</p> <p>RnF2- Los algoritmos deben ejecutarse eficientemente para grafos de mínimo 50 nodos.</p> <p>RnF3- El sistema debe ser modular para facilitar mantenimiento y ampliaciones.</p> <p>RnF4- Las pruebas unitarias deben garantizar confiabilidad en las implementaciones del TAD Grafo.</p> <p>RnF5- El sistema debe ser portable según el lenguaje y herramientas usadas.</p> <p>RnF6- El código debe seguir buenas prácticas de programación y estar adecuadamente documentado.</p> |

| | | | |
|----------------------|--|-------------|----------------------|
| Requerimiento | RF1- El sistema debe permitir usar dos implementaciones diferentes del TAD Grafo. | | |
| Descripcion | El sistema debe utilizar estructuras genéricas para construcción, manejo y modificación de grafos. | | |
| Precondicion | | | |
| Entradas | Entrada | Tipo | Restricciones |
| | | | |
| Postcondicion | | | |
| Salidas | Salida | Tipo | Salida |
| | | | |

| | | | |
|----------------------|--|-------------|----------------------|
| Requerimiento | RF2- El sistema debe permitir cambiar entre las implementaciones del grafo en cualquier momento. | | |
| Descripcion | El sistema debe permitir que el jugador cambie entre los 2 tipos de grafos al momento de pasar o morir en el nivel | | |
| Precondicion | | | |
| Entradas | Entrada | Tipo | Restricciones |
| nivel1 | Button | | |
| nivel2 | Button | | |
| volverMenu | Button | | |
| Postcondicion | Dependiendo del botón se debe cargar la vista para el escenario | | |
| Salidas | Salida | Tipo | Salida |
| mensaje1 | String | | “Nivel 1 cargado” |
| mensaje2 | String | | “Nivel 2 cargado” |
| menu | String | | “Menu cargado” |

| | | | |
|----------------------|--|-------------|----------------------|
| Requerimiento | RF3- El sistema debe modelar y manejar 2 grafos con un mínimo 50 vértices y 50 aristas. | | |
| Descripcion | El sistema debe crear 2 grafos diferentes con un mínimo de 50 nodos y 50 vertices conectados entre ellos | | |
| Precondicion | Se deben crear los nodos y pasarlos al grafo para conectarlos | | |
| Entradas | Entrada | Tipo | Restricciones |
| | | | |
| | | | |
| | | | |
| Postcondicion | Se crean los dos grafos distintos | | |
| Salidas | Salida | Tipo | Salida |
| | | | |
| | | | |
| | | | |

| | | | |
|----------------------|--|-------------|----------------------|
| Requerimiento | RF4- El sistema debe ejecutar al menos dos algoritmos de grafos (BFS, DFS, Dijkstra, Floyd-Warshall, Prim o Kruskal). | | |
| Descripcion | El sistema debe crear dos enemigos por nivel que recorran los grafos en dos diferentes tipos de recorridos, de BFS y Dijkstra. | | |
| Precondicion | Los niveles y los grafos deben estar creados para poder crear a los enemigos y que puedan recorrerlos | | |
| Entradas | Entrada | Tipo | Restricciones |
| | | | |
| | | | |
| | | | |
| Postcondicion | Se crean los dos grafos distintos | | |
| Salidas | Salida | Tipo | Salida |
| | | | |
| | | | |
| | | | |

| | | | |
|----------------------|---|-------------|--|
| Requerimiento | RF5- El juego debe permitir la participación de uno o más jugadores. | | |
| Descripcion | El sistema debe permitir que un jugador recorra y juegue cada nivel, evadiendo a los enemigos y con el objetivo final de llegar a la meta | | |
| Precondicion | Los niveles y los grafos deben estar creados para poder moverse en ellos | | |
| Entradas | Entrada | Tipo | Restricciones |
| | arriba | KeyEvent | |
| | derecha | KeyEvent | |
| | izquierda | KeyEvent | |
| | abajo | KeyEvent | |
| Postcondicion | Por cada tecla que se presiona, el jugador se mueve según la dirección a la que se refiere | | |
| Salidas | Salida | Tipo | Salida |
| | movimiento | Imagen | Se ve la animación de movimiento con dirección |
| | | | |
| | | | |

| | | | |
|----------------------|---|-------------|---|
| Requerimiento | RF6- El sistema debe contar con una interfaz gráfica para interactuar con el grafo y el juego. | | |
| Descripcion | El sistema debe cargar visualmente los dos grafos en una interfaz, así como también los enemigos y el jugador en cada nivel para permitir entender y disfrutar la experiencia | | |
| Precondicion | Los niveles, grafos, jugador y enemigos deben estar cargados para poderlos ver en el mapa | | |
| Entradas | Entrada | Tipo | Restricciones |
| | | | |
| | | | |
| | | | |
| | | | |
| Postcondicion | Se deben poder ver en el mapa a todos los personajes | | |
| Salidas | Salida | Tipo | Salida |
| | Mapa | Imagen | Se ve la imagen del mapa del laberinto y sus conexiones |
| | Entidades | Imagen | Se ven las entidades moviéndose por el mapa |
| | | | |