

**Tarea Programa II – Laberintos**

Tomas Apeltauer

Felipe Mora

2014

INDICE

|  |  |
| --- | --- |
| Resumen Ejecutivo | 3 |
| Introducción | 4 |
| Presentación y análisis del problema | 5 |
| 1. Que es lo que hay que resolver | 5 |
| 1. Como se va a resolver el problema | 6 |
| 1. Análisis crítico de la implementación. | 7 |
| Conclusiones |  |
| Recomendaciones |  |
| Referencias |  |

Resumen Ejecutivo

El proyecto nace con la finalidad de entretener, enseñar, y poner a prueba al usuario final de este. En momentos ajetreados del día muchas personas ocupadas necesitan en sus cinco minutos de descanso poder despejar sus mentes pero al mismo tiempo poniéndolas a trabajar de una forma divertida y aprendiendo métodos de resolución de problemas.

Para fines como este se utilizan juegos como sopas de letras, crucigramas, sudokus y el que se emplea en este proyecto, el laberinto. Con un simple termino de iniciar en un punto de una cuadricula (tablero), hasta llegar al final que es el punto de salida del laberinto. Un juego muy simple en cuanto a reglas se complica un poco en cuanto como jugarse ya que depende de las rutas muertas, el tamaño del tablero así como la capacidad del usuario de poder resolver este tipo de problemas pueden complicar este juego.

El fin de este documento es tratar y mostrar al usuario los puntos que analizo el equipo de trabajo que sería importante que tuvieran conocimiento, iniciando por un planteamiento detallado del problema que se presenta, sus antecedentes y la manera que se empleó para realizarse.

Introducción

En las empresas privadas o públicas muchos trabajadores se ven expuestos a un gran nivel de estrés producido ya sea por asuntos personales o las responsabilidades laborales, para esto se busca la intervención de juegos interactivos y de que de una forma u otra ayude a estas personas a liberar estrés mediante la liberación de la mente y no solo liberarla, también despejarla y abrirle un espacio de tranquilidad donde puedan jugar con su cerebro estimulándolo.

Pero este tipo de estímulo no solo es apto para las personas que trabajan todos los días en empresas, sino también para los educadores, y público en general, demostrando ser un juego muy incluyente. A tal punto que también ayuda a los niños en etapas de crecimiento donde se les hace ver las maneras de estimular su cerebro de una manera divertida para los niños según el diseño físico o de interfaz del laberinto. Estos aprenden a ver más allá de simplemente salir del laberinto a fuerza bruta sino buscando inteligentemente una salida lo que les desarrollara una habilidad de resolución de todo tipo de problemas en el futuro y su razonamiento lógico también de lo que se debe y puede hacer y lo que no sin necesidad de videojuegos violentos.

Estos son solo algunos casos donde se podría aplicar este software pero sin dejar por fuera a otros usuarios que lo hacen simplemente por entretenimiento, pasar el rato o inclusive enseñar a sus hijos. Viendo esto podemos apreciar la inclusión de todos los sectores de la sociedad en este proyecto.

Siendo un proyecto con una política de juego muy inclusiva se puede apreciar que la única finalidad de este es motivar al usuario a estimular su cerebro, a divertirse y por qué no a competir durante la resolución de estos con otros usuarios.

En este se incluyen una interfaz gráfica agradable para el usuario final, la creación algorítmica de estos, así como ayudas para el usuario.

Dentro de un punto de vista más técnico en el presente documento se detallar ampliamente conceptos utilizados para el manejo de estructuras de datos para la realización del presente, así como la técnica de abstracción de conceptos hasta el punto en que fue posible el desarrollo del mismo.

El fin de este documento es tratar los puntos más específicos de posible interés tales como el planteamiento detallado del problema que se presenta, sus antecedentes y una manera muy completa de solucionarla realizada por el equipo de trabajo.

El análisis de este problema, la abstracción de conceptos hasta llegar a cada parte del problema y solucionarlo por aparte para formar la solución completa que se presenta y se plantea únicamente con el fin de mejorar el sistema para que sea más flexible y la mayor comodidad para ambas partes (usuarios y funcionarios).

Presentación y análisis del problema

En este punto del documento se dividirá en tres partes en la primera donde mediante métodos de abstracción se identificaran pequeños problemas a resolver dentro del desarrollo del proyecto. En la segunda parte de esta sección se explica la forma de resolver estos pequeños problemas planteados en la parte anterior pero a un nivel más descriptivo desde el punto de vista del grupo de trabajo como diseñadores del software.

Finalmente en la tercera parte de esta sección se detalla que se logró implementar y que no se pudo completar así como la razón del porque no se logró. Igualmente existe un análisis donde se detalla que se puede mejorar en el software además de cómo podría hacerse para una futura segunda versión.

1. Que es lo que hay que resolver

El problema principal a resolver es el de diseñar un software capaz de implementar un laberinto gráficamente desde el ordenador. Este se debe implementar mediante el uso de grafos, la resolución mediante un árbol de expansión del grafo.

Además de esto se enumeró la siguiente lista de prioridades:

* Creación de una estructura grafo con todos los métodos que este conlleva.
* Creación de una estructura aristas con sus respectivos métodos para unir dos vértices.
* Creación de una estructura vértices con sus respectivos métodos para poder guardar los datos en ellas.
* Creación de un algoritmo genérico que llene el árbol según la conveniencia para el grafo que se representara como el laberinto.
* Creación de un subgrafo que donde sea difícil encontrar la ruta entre dos grafos por lo general inicio y fin.
* Mostrar una bienvenida agradable al usuario.
* Iniciar en modo consola y luego pasar a modo gráfico con la librería específica que mostrara el laberinto.
* Creación de una estructura para la búsqueda en profundidad modificada.
* Obtener la lista de vecinos no visitados de un nodo en específico para la resolución de un problema.
* Crear el laberinto a partir de las medidas ingresadas por un usuario.
* Presentar el laberinto en forma gráfica.
* Poder mover el jugador según las reglas dentro del laberinto.
* Habilitar la ayuda para guiar el jugador hasta el final del laberinto.
* Ejecutar el Dijkstra para ayudar al jugador.
* Desaparecer la ayuda cuando el jugador presiona una tecla.
* Establecer un número fijo de veces para desplegar la ayuda al usuario,

1. Como se va a resolver el problema

El equipo de trabajo identifico 5 pasos más además de los que se brindaron por el asesor del proyecto, estos son:

* **Grafos**: implementar los grafos completamente todas sus estructuras. Estos se abstrae a todo lo lógicamente referente a esto. Empleando una formula especial para unir los nodos dentro del grafo para poder conectarlo perfectamente y además accesar a ellos perfectamente.
* **Subgrafos:** Crear un subgrafo con especificaciones establecidas con el búsqueda en profundidad para dibujar el camino del laberinto.
* **GUI:** en este paso se implementa todo lo que tiene que ver con interfaz gráfica. Graficando los nodos y sus caminos correspondientes.
* **Jugador:** en este paso se verificara los movimientos del jugador ya sea validarlos y contarlos, como la graficacion de estos.
* **Dijkstra:** Utilizando posiciones de inicio y final donde inicio variara ya sea si se implementa desde el inicio del laberinto o desde la posición del jugador hasta el final donde este final si será siempre el mismo ya que será la salida del laberinto.

En la sección anterior se describieron varias tareas por realizarse y al inicio de esta se brinda la clasificación que busco el equipo de trabajo para realizarla. Ubicando estas tareas en estos puntos de clasificación.

Se trató de ubicar en orden de prioridad de asuntos según la complejidad y algunos aspectos en común. Bajo criterios de similitudes también desde lógica e iniciación de graficacion para tener una estructura básica de trabajo. Lo relacionado con el jugador se deja al cuarto paso para que cuando se trate de implementar el jugador sea más fácil la validación y los movimientos de este. Cosas como el Dijkstra y algunos detalles de interfaz se dejan de último en caso de falta de tiempo tener el trabajo lo último y menos necesario para poder jugar la aplicación.

1. Análisis crítico de la implementación

Conclusiones

* El equipo de trabajo concluyo que se realizó una buena labor en cuestiones de principios de programación la abstracción de las funciones. El software se trató de hacer con el código lo más claro posible y escalable en caso de una segunda versión para el mejoramiento de detalles que otra persona fuera del proyecto tal vez pueda apreciar mejor.
* Se puede resaltar el aprendizaje sobre el manejo de la librería de winbgim para manejo de interfaz gráfica siendo una interfaz manejable y sencilla.
* El aprendizaje sobre la implementación de grafos y sus estructuras asociadas ayudan a una mejora en el manejo de estructuras de datos, empleando puntos lógicos para resoluciones de asignaciones de datos y valores.
* La implementación del Dijkstra como método de resolución de algo no tan relacionado a la computación como un simple laberinto lo que deja ver el uso de estos algoritmos y estructuras para la vida real.
* La complejidad del algoritmo de búsqueda en profundidad y su relación en la formulación de cosas complejas.

Recomendaciones

* Buscar abstraer lo máximo posible los diferentes sub problemas que este proyecto presenta.
* El diseño en papel de las clases es muy importante ya que nos da un esqueleto para el posible programa.
* Emplear los algoritmos facilitados por el profesor para una mayor facilidad de uso de tiempo y abstracción de conceptos.
* Tratar de tener una buena comunicación de equipo con el compañero para entre los dos unir ideas y buscar lo más eficiente intercambiando conocimiento.

Referencias

* Maze generation algorithm. (2013, October 15). In Wikipedia, The Free Encyclopedia. Retrieved 20:18, October 29, 2013, from <http://en.wikipedia.org/w/index.php?title=Maze_generation_algorithm&oldid=577285847>.
* Tutoriales de programación; switch case in C and C++ tomado de: <http://www.cprogramming.com/tutorial/lesson5.html>
* Compilador de C++ online; <http://www.compileonline.com/compile_cpp11_online.php>
* <http://stackoverflow.com/questions/1074474/should-i-usedouble-or-float>
* Header files and pheriferal examples tomado de: <http://www.ti.com/tool/sprc832>
* Solving multiple definition tomado de: <http://www.cplusplus.com/forum/unices/5784/>
* Uso de Winbgim tomado de: <http://www.cplusplus.com/forum/beginner/7409/>
* Borlang BGI graphics emulation for the Mingw (GCC port) <http://winbgim.codecutter.org/>
* Manual de Winbgim <http://www.stefansundin.se/programmering/proga/WINBGIM-%20MANUAL.pdf>