

ASIGNATURA: Programación Web  
PERÍODO ACADÉMICO: 2024-2  
FECHA: Semana 13  
TIEMPO: - minutos

NOTA

### ASIGNACIÓN DE PROGRAMACIÓN

CÓDIGO	APELLIDOS Y NOMBRES	SECCIÓN

### ANTES DE INICIAR LA EVALUACIÓN DEBE LEER LAS INSTRUCCIONES

Si la evaluación indica cargar algún archivo en la computadora, recuerde que es responsabilidad del estudiante hacerlo en el tiempo establecido y con las instrucciones dadas. Debe indicar el número de la misma en el recuadro siguiente:

#### INSTRUCCIONES GENERALES:

- El trabajo deberá de realizarse en grupos de 5 integrantes (1 grupo será de 4).
- Debe entregar su código fuente por la plataforma Blackboard.
- Durante clase, deberá de realizar la sustentación de lo realizado. Es obligatorio presentarse a la sustentación, si no el alumno tendrá nota 0.
- **Leer detenidamente las situaciones que ocasionarán la anulación de la evaluación, que se encuentran a continuación.**

#### SITUACIONES QUE OCASIONARÁN LA ANULACIÓN DE LA EVALUACIÓN:

- No presentarse a la sustentación del trabajo.
- No se autores del código realizado.
- Conversar durante el desarrollo de la evaluación.

*Los profesores de la asignatura*

<div></div> <div>FIRMA DEL ALUMNO (LEYÓ LAS INSTRUCCIONES)</div>
--

## **Algoritmos de Ruteo (Pathfinding)**

Uno de los campos de uso de la inteligencia artificial en los juegos es lo que se conoce como el pathfinding o la búsqueda del camino.

Este campo estudia los algoritmos que permiten que un objeto del juego (puede ser un personaje) se mueva hacia una posición objetivo, evadiendo lugares inaccesibles.

### **Instrucciones**

Se le pide implementar una aplicación en lenguajes C++ y en otro lenguaje, que permita probar los algoritmos de pathfinding. Para esto, la aplicación debe tener interfaz gráfica y debe implementar dos algoritmos de pathfinding (uno de cada lista). La elección se realizará al azar.

#### **Lista A**

- Breadth-First-Search
- Dijkstra
- A\* (A star)

#### **Lista B**

- D\*
- Focussed D\*
- D\* Lite

### **Requerimientos**

1. El aplicativo debe mostrar gráficamente una grilla en la pantalla. La cantidad de casillas de la grilla debe calcularse dependiendo del ancho y alto de la pantalla. Cada grilla debe ser de un color celeste claro (#b8edff).
2. Apretando la tecla CTRL, el aplicativo debe entrar en modo de edición de escenario. En este modo, el usuario podrá hacer tres cosas:
  - a. Al hacer click izquierdo en una casilla la bloqueará (no podrá ser parte de una ruta) y la casilla deberá de pintarse de color rojo (#ff0000).
  - b. Al hacer click derecho en una casilla, esta se desbloqueará (podrá ser parte de la ruta). En caso en que esté bloqueada (color rojo), deberá regresar al color celeste claro. En caso de que la casilla que se le haga click derecho ya esté desbloqueada, entonces no se hará nada.
  - c. Al volver a apretar la tecla CTRL se saldrá del modo de edición.
3. Fuera del modo de edición, el aplicativo debe permitir seleccionar con el mouse (click izquierdo) una casilla origen y una casilla destino (de manera consecutiva). La casilla origen debe de pintarse del color verde (#04ff00) y la casilla destino de color amarillo (#ffea00).
4. Al presionar la tecla ENTER y tener seleccionadas las casillas de origen y destino, la aplicación deberá de pintar la ruta entre el origen y destino. Las casillas pertenecientes a esta ruta deberán de pintarlas con el color azul (#0000ff). La ruta la debe realizar siguiendo el algoritmo de la lista 1.
5. Al presionar la tecla ESPACIO y tener seleccionadas las casillas de origen y destino, la aplicación deberá de pintar la ruta entre el origen y destino. Las casillas pertenecientes a esta ruta deberán de pintarlas con el color azul (#0000ff). La ruta la debe realizar siguiendo el algoritmo de la lista 2.
6. Al presionar la tecla ESC, la aplicación deberá de reiniciarse a su estado inicial (todas las casillas desbloqueadas y de celeste claro).

## Rúbrica

Concepto	Detalle
<b>Funcionalidad</b> <b>0 - 12 puntos</b>	Implementa las funcionalidades pedidas en las instrucciones. Se darán dos puntos por cada requerimiento (2 puntos: todo correcto, 1 punto: con errores no bloqueantes, 0 puntos: con errores bloqueantes o no hay implementación)
<b>Utilización de estructuras del lenguaje</b> <b>0 – 4 puntos</b>	Debe utilizar las estructuras y constructos que el lenguaje permite. Su utilización debe ser sustentado en base a los siguientes criterios: <ul style="list-style-type: none"><li>- Performance</li><li>- Legibilidad</li><li>- Confiabilidad (no propenso a errores)</li></ul>
<b>Sustentación</b> <b>0 – 4 puntos</b>	Durante la sustentación, no solo deberán exponer las funcionalidades, si no también las decisiones de diseño en cuanto el uso de las estructuras del lenguaje de programación.
<b>Contribución (individual)</b> <b>Puntos negativos (hasta - 4)</b>	Nivel de participación del alumno en el desarrollo del proyecto: <ul style="list-style-type: none"><li>- - 1 puntos: No ha tenido contribución en el código, pero si ha respondido correctamente las preguntas realizadas.</li><li>- - 2 puntos: Ha respondido correctamente solo una pregunta realizada.</li><li>- - 3 puntos: No ha respondido totalmente de forma correcta ninguna de las preguntas.</li><li>- - 4 puntos: No ha demostrado suficiencia en sus respuestas, lo que demuestra una baja contribución.</li></ul>