



Facultad de  
Ingeniería  
UNAM



# Universidad Nacional Autónoma de México

Torres de Hanoy

---

**Manual de Configuración**

RAMÍREZ GARCÍA DIEGO ANDRÉS



Universidad  
Nacional  
Autónoma de  
México

Fecha de entrega: 07/12/2023

# Índice

<b>TORRES DE HANOY .....</b>	<b>3</b>
PROPÓSITO.....	3
DESCRIPCIÓN DEL PROGRAMA .....	3
MANUAL DE USO.....	3
REQUISITOS PARA LA EJECUCIÓN DEL EJECUTABLE .....	4
<i>Requisitos de sistema.....</i>	<i>4</i>
<i>Requisitos de software.....</i>	<i>4</i>
<i>Bibliotecas y Dependencias.....</i>	<i>4</i>
INSTRUCCIONES DE EJECUCIÓN .....	4
REQUISITOS PARA EL ENTORNO DE DESARROLLO .....	5
<i>Requisitos de sistema.....</i>	<i>5</i>
<i>Requisitos de software.....</i>	<i>5</i>
<i>Bibliotecas y Dependencias.....</i>	<i>5</i>
CONFIGURACIÓN DEL ENTORNO DE DESARROLLO.....	6

# Torres de Hanoy

## Propósito

---

Este manual tiene como finalidad ofrecer una guía exhaustiva que simplifique la comprensión, instalación y uso eficaz del simulador de las Torres de Hanoy. Con el propósito de facilitar la consulta, se ha estructurado el manual en secciones temáticas y subsecciones. Además, se ha incorporado instrucciones detalladas sobre la configuración del entorno de desarrollo Visual Studio, permitiéndote explorar, ejecutar y realizar modificaciones en el simulador de manera eficiente.

## Descripción del programa

---

Este simulador se compone de tres torres y tres toroides (discos). Inicialmente, los tres toroides se encuentran colocados en la primera torre, ordenados de mayor a menor, de abajo hacia arriba.

Al presionar la tecla *SPACE* (o espacio), se inicia una animación que traslada los toroides uno a uno hacia la torre opuesta (torre 3). Al concluir la animación, los tres toroides quedan posicionados en la torre número 3. Esta animación se presenta de manera paso a paso y a una velocidad constante, permitiendo una apreciación óptima.

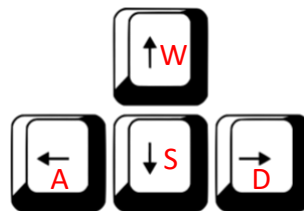
La secuencia de traslado y ordenamiento se repite al volver a presionar la tecla *SPACE*, es decir, los tres discos regresan a la torre 1, dando inicio a una nueva simulación.

## Manual de uso

---

La solución documentada implementa la simulación de traslado y ordenamiento de la torre 1 (izquierda) a la torre 3(derecha) al presionarse la tecla *SPACE*. Esta simulación abarca la implementación de un *reset* para volver a iniciar la simulación; esta opción está habilitada al volverse a presionar la tecla *SPACE*.

De forma adicional y con fines recreativos, mediante las teclas A,D,W y S se le permite al usuario desplazarse por el entorno. Las teclas están mapeadas de la siguiente forma:



## Requisitos para la ejecución del ejecutable

---

Para ejecutar el simulador de las Torres de Hanoy desarrollado en C++ mediante *Visual Studio*, es necesario asegurarse de que el sistema cumpla con los siguientes requisitos:

### Requisitos de sistema

Sistema Operativo:

- *Windows 7* o posterior

Hardware:

- Procesador compatible con arquitectura x86 o x64.
- Se recomienda un mínimo de 2 GB de RAM.
- Tarjeta gráfica con soporte para OpenGL 3.3 o superior.

### Requisitos de software

Entorno de Ejecución:

- Visual C++ Redistributable en Windows, es decir, archivos y bibliotecas necesarias para ejecutar programas desarrollados en C++.

### Bibliotecas y Dependencias

- glew32.dll
- assimp-vc140-mt.dll
- Carpeta con los shader utilizados
- Carpeta con los modelos (obj)

## Instrucciones de Ejecución

---

Una vez verificado el sistema, donde se quiere correr el ejecutable, cumple con los requerimientos del apartado anterior, se deben verificar tanto el archivo ejecutable como los recursos necesarios estén en el mismo directorio. Finalmente, para ejecutar el .exe basta con dirigirnos a la carpeta *Realease*, y dar clic sobre el ejecutable.

## Requisitos para el entorno de desarrollo

---

### Requisitos de sistema

#### Sistema Operativo:

- *Windows 7* o posterior

#### Hardware:

- Procesador compatible con arquitectura x86 o x64.
- Se recomienda un mínimo de 2 GB de RAM.
- Tarjeta gráfica con soporte para OpenGL 3.3 o superior.

### Requisitos de software

Entorno de desarrollo: *Visual Studio 2022* o posterior.

### Bibliotecas y Dependencias

Las bibliotecas enlistadas entre corchetes (“ ”) forman parte del código fuente pues son propias del proyecto, para hacer uso de ellas se debe importar, a la carpeta de la solución del proyecto, la carpeta *Headers* y *SOIL2*.

#### *Bibliotecas*

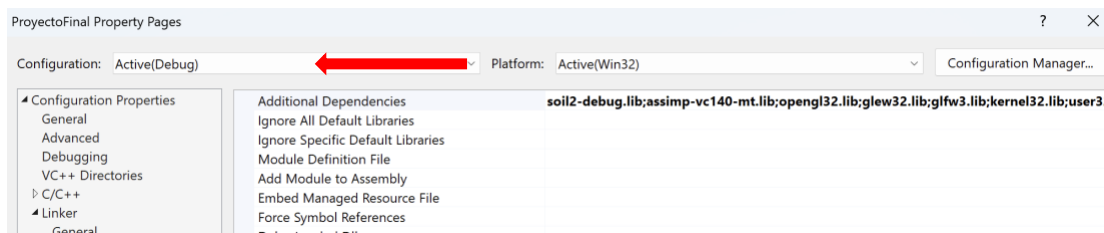
- `iostream`
- `cmath`
- `GL/glew.h`
- `GLFW/glfw3.h`
- `glm/glm.hpp`, `glm/gtc/matrix_transform.hpp`, `glm/gtc/type_ptr.hpp`
- `"stb_image.h"`
- `"SOIL2/SOIL2.h"`
- `" Shader.h"`
- `" Camera.h"`
- `" Model.h"`

#### *Dependencias*

- `glew32.dll`
- `assimp-vc140-mt.dll`

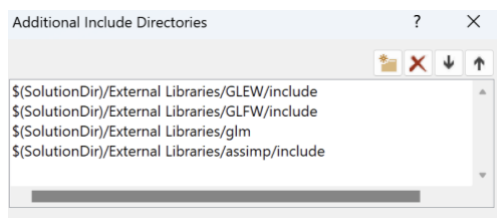
## Configuración del Entorno de Desarrollo

Para la configuración siguiente, debemos colocar en “*Solution Platform*” x86, luego nos dirigimos a las propiedades de la solución e indicamos en la esquina superior izquierda en “*Configuration*” la opción *All Configurations*.



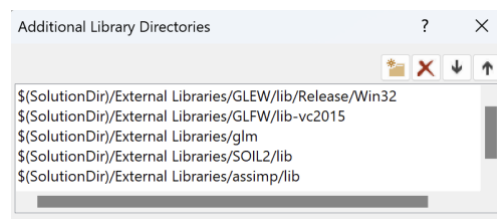
Ahora bien, estando en el apartado de propiedades, *nos dirigimos a C/C++ > General > Additional Include Directories*. En esta opción añadimos las direcciones donde están nuestras bibliotecas y dependencias adicionales.

**\$(SolutionDir)/External Libraries/GLEW/include**  
**\$(SolutionDir)/External Libraries/GLFW/include**  
**\$(SolutionDir)/External Libraries/glm**  
**\$(SolutionDir)/External Libraries/assimp/include**



Luego nos dirigimos a *Linker > General > Additional Library Directories*. En esta opción añadimos las direcciones donde están nuestras bibliotecas.

**\$(SolutionDir)/External Libraries/GLEW/lib/Release/Win32**  
**\$(SolutionDir)/External Libraries/GLFW/lib-vc2015**  
**\$(SolutionDir)/External Libraries/glm**  
**\$(SolutionDir)/External Libraries/SOIL2/lib**  
**\$(SolutionDir)/External Libraries/assimp/lib**



Y finalmente en *Properties > Linker > Input > Additional Dependencies* colocamos las últimas dependencias. Si en el apartado ya tenemos algunas dependencias, adicionamos estas antes o después pero separadas por punto y coma (;).

**soil2-debug.lib;assimp-vc140 mt.lib;opengl32.lib;glew32.lib;glfw3.lib**