



Construcción de un modelo de simulación determinista con JaamSim

Apellidos, nombre	Cardós Carboneras, Manuel (mcardos@doe.upv.es) Guijarro Tarradellas, Ester (esguitar@upvnet.upv.es)
Departamento	Organización de Empresas
Centro	Universitat Politècnica de València



1 Resumen

A continuación se presentan los elementos que deben conocerse del software JaamSim para construir un primer modelo de simulación determinista dinámico y discreto y hacerlo funcionar. Este será nuestro primer contacto con un software de simulación, así que realizaremos varias tareas:

1. Instalaremos JaamSim en nuestro ordenador.
2. Identificaremos las diferentes ventanas de JaamSim y su propósito.
3. Construiremos un modelo de simulación.
4. Haremos funcionar el modelo de simulación.

2 Introducción

Los modelos estocásticos dinámicos y discretos son los modelos de simulación más habituales en Dirección de Operaciones. Entonces, ¿por qué construir un modelo determinista dinámico y discreto? La respuesta está relacionada con la estrategia más recomendable para construir un modelo de simulación:

- Empezar por la versión más simple posible del sistema real que deseamos modelar.
- Validar su funcionamiento, es decir, comprobar que funciona como pretendíamos.
- Ampliar el modelo incluyendo elementos y características cada vez más realistas.
- Validar de nuevo el modelo.
- Repetir los pasos anteriores hasta que el modelo incluya todas las características deseadas.

En consecuencia, aprenderemos primero a construir un modelo determinístico y sólo después nos plantearemos un modelo estocástico.

3 Objetivos

El propósito de este documento es que, al completar su lectura habiendo realizado las tareas propuestas, el lector sea capaz de:

- Instalar el software JaamSim.
- Identificar las ventanas de la aplicación y sus usos principales.
- Construir y hacer funcionar un modelo básico de simulación.

4 Desarrollo

4.1 Instalación de JaamSim

Los requisitos para su instalación son:

- El ordenador debe tener instalada una máquina virtual JAVA 7 o superior. Estas máquinas están disponible, y con frecuencia instaladas por defecto, en Windows, Linux y OS X. Sin embargo no existen ni para Android ni para IOS.
- La tarjeta gráfica debe cumplir con la norma OpenGL 3.0 o superior para que el renderizado de los modelos en 2 y 3 dimensiones funciones con fluidez. Puede funcionar también con OpenGL 2.1 aunque su rendimiento se degrada.

Los pasos que hay que seguir son:

- Descargar de <http://jaamsim.com/downloads.html> el software de simulación y el manual de usuario de la versión disponible.
- Ejecutar el programa, si muestra las ventanas de la Imagen 1 ya podemos empezar a usarlo.

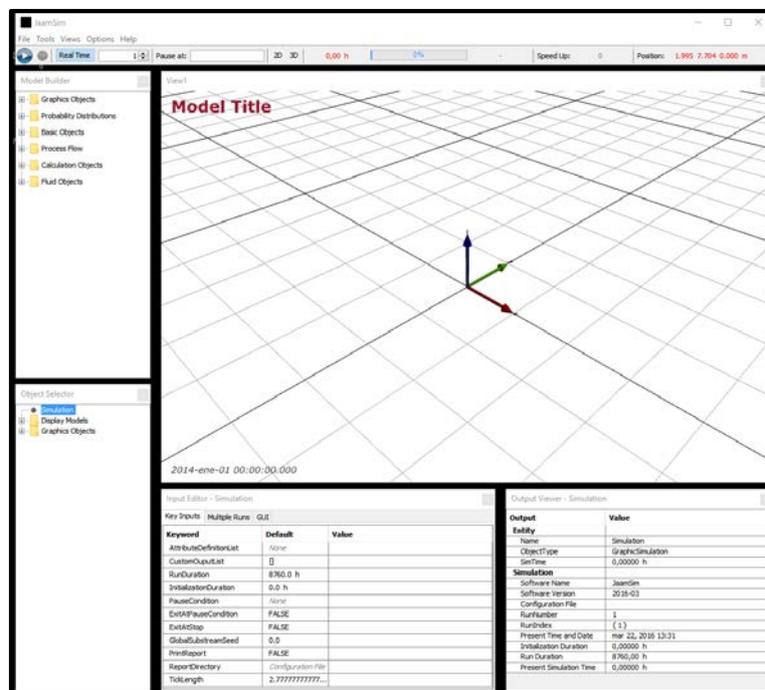


Imagen 1. Inicio de JaamSim

4.2 Problemas de instalación

Sin embargo, en ocasiones surgen problemas relacionados con los requisitos anteriores:

- Al ejecutar JaamSim no sucede nada y no aparece el icono de JAVA junto al nombre del programa como en la Imagen 2. La causa es que falta instalar la máquina virtual en el ordenador para lo que se puede seguir las instrucciones de <https://www.java.com/es/download/>. También se puede utilizar OpenJDK en Linux.

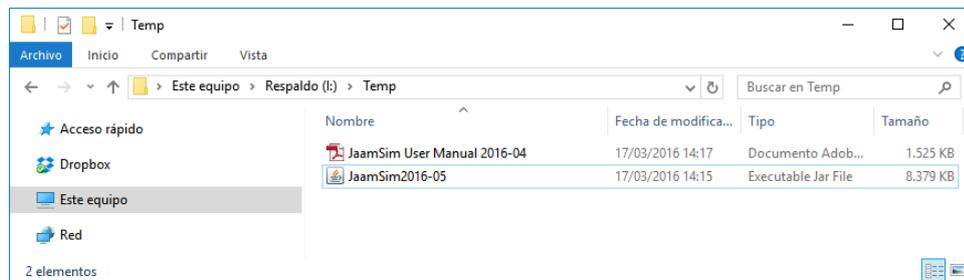


Imagen 2. La taza de café como icono de JaamSim

- Al ejecutar JaamSim aparecen las ventanas de la Imagen 1 excepto la ventana grande con los ejes de referencia. En este caso, la tarjeta gráfica no cumple al menos con OpenGL 2.1 o al menos no lo hace con los drivers que hay instalados. En ocasiones se soluciona el problema actualizando los drivers de la tarjeta de video con los últimos proporcionados por el fabricante. Una vez actualizados, se puede comprobar el resultado mediante GL-Z, un sencillo programa <http://www.geeks3d.com/glz/> que nos muestra la versión disponible.



```

GLSL Hacker 0.8.4.0 | GL-Z 0.3.0 | 40FPS, 31,540ms | Intel(R) HD Grap...
GL-Z simple OpenGL information utility
(C) 2014-2015 Geeks3D

- Platform: Windows 64-bit
- GL_VENDOR: Intel
- GL_RENDERER: Intel(R) HD Graphics 4000
- GL_VERSION: 4.0.0 - Build 10.18.10.4358
- GL_SHADING_LANGUAGE_VERSION: 4.00 - Build 10.18.10.4358
- # OpenGL extensions: 188
001> GL_EXT_blend_minmax
002> GL_EXT_blend_subtract
003> GL_EXT_blend_color
004> GL_EXT_abgr
005> GL_EXT_texture3D
006> GL_EXT_clip_volume_hint
007> GL_EXT_compiled_vertex_array
008> GL_SGIS_texture_edge_clamp
009> GL_SGIS_generate_mipmap
010> GL_EXT_draw_range_elements
011> GL_SGIS_texture_lod
012> GL_EXT_rescale_normal
013> GL_EXT_packed_pixels
014> GL_EXT_texture_edge_clamp
015> GL_EXT_separate_specular_color

OpenGL hardware limits
> GL_MAX_VERTEX_ATTRIBS: 16
> GL_MAX_TEXTURE_SIZE: 16384
> GL_MAX_TEXTURE_IMAGE_UNITS: 16
> GL_MAX_COLOR_ATTACHMENTS: 8
> GL_MAX_VIEWPORT_DIMS: 16384 16384
> GL_MAX_SAMPLES: 8
> GL_MAX_TESS_GEN_LEVEL: 64
> GL_MAX_PATCH_VERTICES: 32
> GL_MAX_COMPUTE_WORK_GROUP_SIZE: -1

Extension list navigation: PAGEDOWN/PAGEUP, UP/DOWN and HOME/END
  
```

Imagen 3. GL-Z muestra que se dispone de OpenGL 4.0.0

4.3 Primer contacto con JaamSim

Al iniciar JaamSim aparecen las ventanas que se muestran en la Imagen 4:

- Panel de control: permite cargar o guardar modelos, cambiar el tipo de vista del modelo, iniciar una simulación, cambiar su velocidad o pararla.
- Vista: muestra el estado del sistema a lo largo del tiempo.
- Constructor del modelo: proporciona los elementos que usamos para construir un modelo mediante arrastrar y soltar en la vista.
- Selector de objetos: incluye todos los objetos que forman el modelo que construimos.
- Editor de entrada: cuando seleccionamos un objeto (en la ventana o en el selector de objetos) muestra los parámetros de entrada en la pestaña *Key Inputs* y los de su representación gráfica en la pestaña *Graphics*.
- Visor de salida: muestra en cada momento el valor que tienen las variables del objeto seleccionado; su análisis permite conocer el comportamiento del modelo.

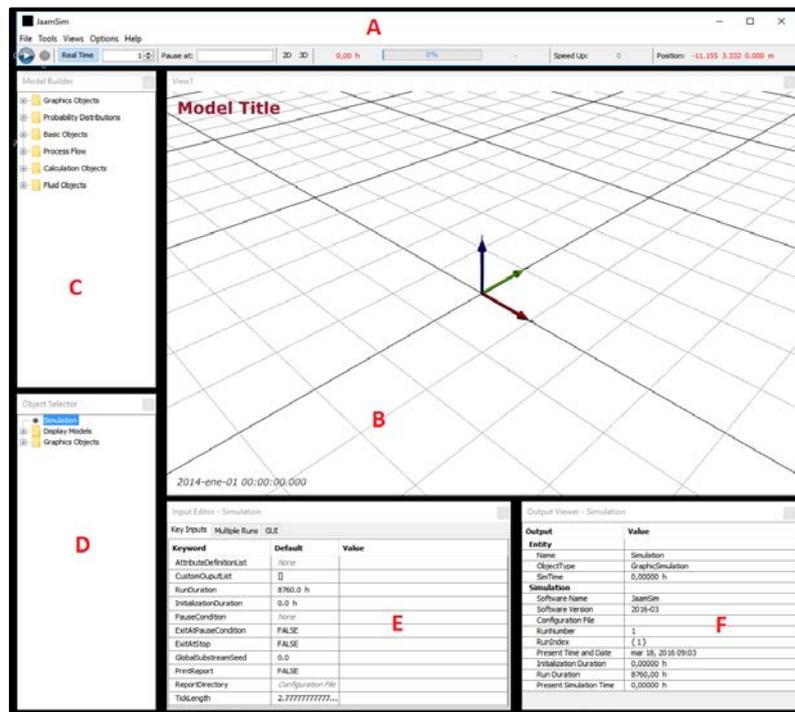


Imagen 4. Ventanas iniciales

Vamos a empezar a configurar un modelo determinista de una máquina con su correspondiente zona de espera para las piezas que deban esperar:

- Modelo plano
Panel de control / botón 2D
- Eliminamos la visualización de ejes y parrilla
Panel de control / *Options* / desmarcar *Show Axes* y *Show Grid*
- Cambiamos el título del modelo



Object Selector / Graphics Objects / OverlayText / Title / Key Inputs / Format / 'Una Máquina determinista'

- Ajustar el reloj de la simulación

Object Selector / Graphics Objects / OverlayClock / Clock / StartingYear / 2000

4.4 Seleccionar y configurar objetos

Es recomendable empezar por colocar en la Vista los objetos que necesitamos y en su posición, aunque luego se puedan cambiar. Los objetos que necesitamos los obtenemos de *Model Builder / Process Flow* y los arrastramos y soltamos en la ventana.

Una norma de estilo recomendable es colocar junto a cada objeto una etiqueta con su nombre y cambiarlo por un nombre sencillo, para lo cual:

- Uno a uno, seleccionar cada objeto con el ratón
Ventana / seleccionar objeto / botón derecha ratón / Add Label
- Con el objeto seleccionado, en *Object Selector* aparece su nombre, seleccionar el nombre actual con el ratón y escribir encima el nuevo nombre, con lo que se actualiza la etiqueta.

Posteriormente configuramos cada objeto:

- *SimEntity*: entidad que se utiliza para la simulación, en nuestro caso piezas. Reducimos su tamaño con:
Ventana / Seleccionar objeto / Input editor / pestaña Graphics / Size / 0.2 0.2 0.2 m
- *EntityGenerator*: entrada de las piezas en el sistema cada 50 segundos
Ventana / seleccionar Entrada / Input Editor / Key Inputs / PrototypeEntity / Pieza en el desplegable
InterArrivalTime / 50 s
NextComponent / Transp1 en el desplegable
- *EntityConveyor*: transportador por el que circulan las piezas durante 500 segundos
Ventana / seleccionar Transp1 / Input Editor / Key Inputs /
TravelTime / 500 s
NextComponent / Maquina en el desplegable
- *Queue*: cola de espera de las piezas que llegan y se encuentran la máquina ocupada, no hace falta más configuración.
- *Server*: características del proceso en la máquina
Ventana / seleccionar Maquina / Input Editor / Key Inputs /
NextComponent / Transp2 en el desplegable
WaitQueue / Cola
ServiceTime / 50 s

- *EntityConveyor*: transportador por el que circulan las piezas durante 500 segundos aunque con otro nombre
Ventana / seleccionar Transp2 / Input Editor / Key Inputs /
TravelTime / 500 s
NextComponent / Salida en el desplegable
- *EntitySink*: salida de las piezas, no necesita parametrización.

4.5 Probar el modelo de simulación

Antes de comprobar el funcionamiento del modelo, conviene guardarlo:

Panel de Control / File / Save as

Aunque la extensión de este fichero es *cfg* se trata de un fichero de texto que se puede leer con cualquier editor, incluso *Bloc de Notas*.

Finalmente ya podemos hacer funcionar el modelo durante 10 horas con

Panel de control / Pause at / 10 h / botón Play

El modelo evoluciona despacio porque funciona a la velocidad real, para aumentar la velocidad hay que aumentar el multiplicador de velocidad situado a la derecha del botón *Real Time* que por defecto está pulsado. Si se libera, la simulación se ejecuta tan rápido como puede. El resultado de esta simulación se encuentra en la Imagen 5.

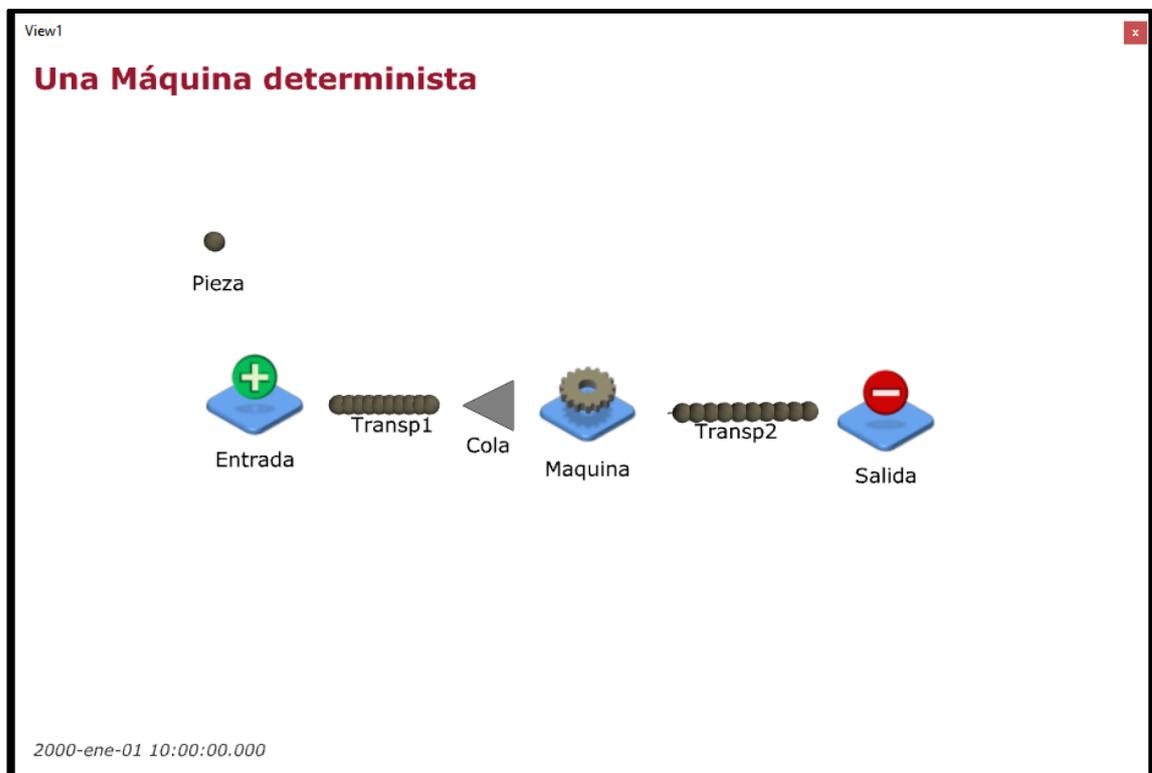


Imagen 5. Resultado de la simulación

Las unidades que han salido del sistema en estas 10 horas las podemos ver en el *Visor de Salida* del objeto *Salida*, Imagen 6. Se aprecia que la simulación ha durado 10 horas y han salido 700 unidades, lo que supone una tasa de 0,0194444 unidades/segundo. Puesto que entra una pieza cada 50 segundos, podríamos esperar que saliese también con esa tasa; sin embargo no es posible porque la primera pieza ha necesitado un tiempo para pasar por los transportadores y ser procesada por la máquina.

Output Viewer - Salida	
Output	Value
Entity	
Name	Salida
ObjectType	EntitySink
SimTime	10,0000 h
DisplayEntity	
Position	4.304132 0.4016 0.0 m
Size	1.0 1.0 1.0 m
Orientation	0.0 0.0 0.0 rad
Alignment	0.0 0.0 0.0
StateEntity	
State	None
WorkingState	false
WorkingTime	0,00000 h
StateTimes	{None=10,0000} h
LinkedComponent	
obj	Entrada_700
NumberAdded	700
NumberProcessed	700
NumberInProgress	0
ProcessingRate	0,0194444 /s
ReleaseTime	9,98611 h

Imagen 6. Visor de salida

5 Cierre

Este ha sido nuestro primer contacto con JaamSim, así que hemos tenido que empezar por instalarlo e identificar sus ventanas principales. Después hemos construido un modelo determinista sencillo, lo hemos hecho funcionar y finalmente hemos obtenido los resultados básicos de la simulación.

6 Bibliografía

<http://jaamsim.com/downloads.html>

<https://www.java.com/es/download/>