

ARQUITECTURA QUE RESPIRA. PROGRAMACIÓN VISUAL BÁSICA APLICADA A PROYECTOS Cartagena, del 22 al 25 de Julio de 2019

25 horas

Objetivos

- Aprendizaje del entorno de programación visual para la generación de prototipos dinámicos de proyectos completos. Plataformas de programación basadas en nodos (node-based) para su gestión.
- Diseño de algoritmos. Planificar y explicitar procesos. Traducción de procesos a lenguajes de programación. Sintaxis básicas comunes entre todos los lenguajes de programación.
- Explorar tanto derivas como objetivos concretos. Programar herramientas de proyecto como parte del proyecto mismo. Explorar el proceso como esencia del proyecto.
- Incorporar al diseño datos externos al mismo. Aprender a programar, automatizar y después matizar decisiones. Generar proyectos adaptativos y reactivos en continua reinformación.
- Explorar los límites de lo codificado: producción de códigos como asistentes y no como imposiciones.
- Interrelacionar decisiones en equipos. Generar marcos y rutinas para el diseño colaborativo.
- Explorar topologías y prototipos, entornos de incertidumbre y posibilidades. Manejo de bases de datos y flujos de herencia y transporte de datos.
- Generación paramétrica modificable, dinámica y evolutiva. Producción de herramientas.

Dirigido a

Estudiantes de arquitectura y arquitectos.

Cada alumno/a deberá asistir con su equipo informático personal y las aplicaciones Rhinoceros 5 o 6 y Grasshopper instaladas.

Programa

I. PRINCIPIOS DE APLICACIÓN PRÁCTICA DE LA PROGRAMACIÓN VISUAL.

Objetivos: Dominio del entorno de programación visual. Exploración de los principios esenciales de la codificación de algoritmos.

PV.1- INTERFACES BASADAS EN NODOS Y USOS DE BASES DE DATOS MEDIANTE ALGORITMOS GENERATIVOS DE LAS CONDICIONES DEL PROYECTO. ASISTENTES Y ANALIZADORES.

Programación de estrategias de modelado. Introducción al algoritmo: recetas y rutinas. Traducción de algoritmos a lenguajes de programación visual. Rutinas y funciones. Elección de variables y objetivos a programar. Diseño mediante Tipos y Prototipos: diferencias entre modelos estáticos y dinámicos. Esquemas en diagramas de flujos. Código visual, selección de inputs y de outputs de la programación. Decisión de fases de diseño y de fases de acción y reacción del modelo dinámico. Fases que dependen de otras fases: vinculación de datos entre fases del diseño. Objetos de programación visual: Parámetros (gestores y almacenes de datos), Componentes (generadores de datos) y cables (transportadores de datos); receptores y emisores. Comandos en la plataforma 3D (Rhinoceros) y Componentes en Grasshopper, identificación. Relaciones entre datos, objetos y geometría: datos en los que consta cada objeto. Clases de objetos de programación, naturalezas y comportamiento. Reconversión o casting entre variables y entre clases de objetos de programación. Lectura de datos de archivos y lectura de datos entre plataformas: parsing o análisis pormenorizado de caracteres Comportamiento de los Objetos de programación visual: comportamiento de las cápsulas de parámetros y cápsulas de componentes (introducción directa, sobreescritura, herencia). Almacenar, fabricar, grabar, limpiar, emitir, recibir e internalizar datos Instancias de objetos de programación vs objetos físicos: modos de selección de objetos de programación en contraposición con los modos de selección de objetos físicos. Gestión de datos persistentes, formulación y expresiones, datos por defecto Herencia de datos: volátiles y persistentes. Sobreescritura y lectura. Gestión de múltiples cables y listas de datos. Almacenaje y apilamiento (orden) en listas. Coincidencia y cruce de flujos de datos. Casting múltiple o conversión de naturalezas de dato. Esquema jerárquico entre todas las clases posibles de datos en la interfaz: specials (timer,etc), brep, surface, mesh, curve, point, string(text), number(floating point), integer,colour,shader... Lógica de confrontación de listas y árboles de

Información útil

Código del curso

C1907

Fecha

El curso se celebrará del 22 al 25 de julio de 2019.

Horario

De 9:00 a 14:00h.

Lugar

Sede Corporativa Cámara de Comercio de Cartagena.

Edif. Cámara de Comercio de Cartagena, Calle Conducto, 5.

Matrícula

180 €.

Documentación a presentar:

- Impreso de [matrícula](#) debidamente cumplimentado.
- Fotocopia de DNI, o en caso de ser extranjero, NIE o Pasaporte.
- En su caso, fotocopia del documento acreditativo de ser estudiante o titulado universitario.

Dirección

Dr. D. Pedro García Martínez.

Profesorado

D. Alberto Moreno Herrero, Arquitecto UPCT. Master in Architectural Management and Design (MAMD).

D. Sergio del Castillo Tello, Arquitecto UPM-ETSAM. CEO de MIAUHAUS.org, Grupo de Expertos en Diseño Computacional. Profesional e Instructor oficial de Diseño y Programación 3D. Cofundador e instructor de MIAU, Master In Advanced Infographics UPM. Profesor del IAM Instituto de Arquitectura de Madrid (Programación Visual, Proyecto Audiovisual, Prototipado) dentro del DPA de la ETSAM. Profesor de Master Product Design y Master European Labs en IED Instituto Europeo de Diseño (Digital Lab, 3D Prototyping) y Profesor de Master en CICE Escuela de Nuevas Tecnologías.

Dr. D. Pedro García Martínez, Doctor Arquitecto UPM-DPA-ETSAM. Profesor de Proyectos Arquitectónicos UPCT.

Colabora



datos según tipos de componente.

PV.2- GESTIÓN DE DATOS MEDIANTE PROTOCOLOS DE SELECCIÓN DE PARÁMETROS DE CONTROL, FLUJOS DE TRABAJO Y OPERADORES A LO LARGO DE LOS FLUJOS.

Listas y gestión de listas mediante componentes y mediante matices de parámetro. Tipos de componentes escalares. Operaciones matemáticas básicas. Constantes, intervalos, polinomios, trigonometría, utilidades de evaluación. Operadores y funciones algebraicas. Sentencias condicionales, booleanos. Disparadores de procesos, sentencias "if...Then" Condiciones que disparan argumentos y condiciones que disparan otras condiciones. Flujos de datos lineales o cíclicos. Bucles(loops): Recursividad (llamada a si mismo) e Iteración (repetición de instrucciones, condiciones de parada, While).

PV.3- GESTIÓN DE DATOS AVANZADA MEDIANTE CRUCES DE ÁRBOLES Y BASES DE DATOS (COMPLEX DATA TREE MATCHING).

Listas de listas: matrices de datos almacenados. Rutas de rama y nombres de ruta. Navegación entre tree-branch-item. Acceso a items a través de rutas, construcción de rutas. Uso mixto de listas y árboles de datos. Importar y exportar de bases de datos. Datos entrelazados: mezclar datos y sobreescritura de datos según clases. Parámetros y Variables almacenadas como árboles de datos. Expresiones que condicionan variables. Exportación e importación de datos. Flujos continuos entre plataformas. Ejemplos de usos concretos de almacenaje de la información del proyecto en bases de datos gestionadas mediante estructuras complejas de rutas (paths).

PV.4- MULTI-TRANSFORMACIONES E INTERACTUACION MASIVA ENTRE OBJETOS Y DATOS.

Fuerzas de transformación y deformación: vectores básicos, matrices, nubes de puntos Manipuladores (handlers): manipulación punto/vector, utilidades de vectores. Objetos manipuladores de otros objetos mediante leyes programadas. Modificaciones adaptativas mediante atractores: campos de interacción. Tipos de transformaciones y matrices: deformación del espacio.

PV.5- ESPACIO PARAMÉTRICO INICIAL Y LA GESTIÓN MATEMÁTICA DE OBJETOS Y SUBOBJETOS.

Espacio paramétrico unidimensional y bidimensional. Intervalos y parámetros t y uv. Análisis de objetos complejos y gestión de los subelementos manteniendo estructura de datos. Reconstrucción, subdivisión y conexión de objetos de los espacios. Operaciones sobre los objetos mediante gestión de sus parámetros internos: dominios, subdivisiones, segmentaciones. Módulos, medidas normalizadas y equilateralización.

PV.6- ESPACIO PARAMÉTRICO AVANZADO PARA EL PANELIZADO COMPLEJO Y MATRICES DE OPERACIONES SOBRE GEOMETRÍAS.

Herramientas de panelizado: panelizado uniforme vs no uniforme. Deformación por morphing entre cajas, superficies. Remapeo de objetos sobre otros objetos, y distribución a lo largo de una superficie o polisuperficie. Proyecciones y presionado de objetos. Panelizado deformado o normalizado. Operaciones de intersección booleanas complejas.

PV.7- ORGANIZACIÓN GLOBAL (HOLÍSTICA Y ESPECIALIZADA) DE LA PROGRAMACIÓN: ESTRUCTURA, ORDEN, OBJETIVO. MODELOS DINÁMICOS RESULTANTES.

Introducción a clusters o conjuntos acoplados de componentes. Interconexiones y ediciones. Componentes de gestión de la programación: interruptores, disparadores, pausadores, temporizadores y ciclos. Flujos mixtos continuos y lineales en el entorno de programación. Referenciación de clústeres. Reprogramación y lógica directa y difusa. Streaming de datos. Introducción a componentes de scripting o programado textual. Multithreading.

II. PROGRAMACIÓN VISUAL APLICADA SOBRE EL PROYECTO.

Objetivos: Aplicación de múltiples rutinas específicas sobre el proyecto concreto simultáneamente y en tiempo real. Proyectos "inteligentes": reactivos o adaptativos, sensibles a los cambios de las condiciones.

PV.8- ESTUDIOS DE CASOS DE PROYECTO SEGÚN MODOS DE GENERACIÓN DEL PROYECTO.

Geometrías complejas adaptativas mediante deformaciones, morphing y form finding. Selección de tipos de proyectos según modo de generación. Naturalezas Nurbs / Mesh. Explosión y reconstrucción de objetos para su deformación por datos externos. Conversiones entre tipos de objeto y tipos de datos. Diagramas como input y output. Topología: Conexiones, voronoi, proximidad; Nubes de Pts.

PV.9- PROGRAMACIÓN DE ELEMENTOS ARQUITECTÓNICOS.

Generación de terrenos según tipos, casos y necesidades: triangulaciones, contornos, alturas. Atractores, permeabilidad. Despieces. Subdivisiones y practicabilidad, cinemática, mecánica. Estructuras espaciales. Triangulación, rutinas de interconexión entre puntos. Estructuras de entramado reticular y poligonales n-gon (poligonos de n-lados) Programación de urbanismo paramétrico.

PV.10- DOCUMENTACIÓN DEL PROCESO Y DE SUS RESULTADOS.

Datos de salida, display, outputs. Materialización. Auto-referenciado. Renderizado. Animación de secuencias y procesos.