deepseek

Programa Académico: "Lógica de Programación para Animación y Arte Digital"

Duración:16 semanas
Metodología: Flowgorithm → Pseudocódigo → C# Unity
Enfoque: Desarrollo progresivo de habilidades desde algoritmos básicos hasta implementación en motor profesional

---

Desglose Detallado por Semanas

Módulo 1: Fundamentos con Flowgorithm (Semanas 1-4)

Semana 1: Introducción a Algoritmos Visuales
- Teoría: ¿Qué es un algoritmo? Aplicación en procesos creativos
- Flowgorithm:
    - Interfaz y herramientas básicas
    - Formas geométricas (Input, Output, Process, Decision)
    - Creación de flujos simples
- Proceso Triple:
    - (a) Diagrama: "Hola Mundo Artístico" con formas básicas
    - (b) Pseudocódigo: Generación automática desde Flowgorithm
    - (c) C# Unity: Traducción manual a Debug.Log en Unity
- Ejercicio: Algoritmo que "dibuja" un patrón con caracteres en consola

Semana 2: Variables y Operaciones para Propiedades Visuales
- Conceptos: Tipos de datos (int, float, string, bool)
- Flowgorithm:**
    - Declaración y asignación de variables
    - Operaciones matemáticas básicas
- Proceso Triple:
    - (a) Diagrama: Calculadora de proporción áurea
    - (b) Pseudocódigo: Variables y operaciones aritméticas
    - (c) C# Unity: Variables públicas en Inspector de Unity
- Práctica: Sistema que calcula dimensiones para composiciones visuales

Semana 3: Control de Flujo y Decisiones Creativas
- Teoría: Condicionales en procesos artísticos
- Flowgorithm:
    - Estructuras If-Else-If anidadas
    - Operadores lógicos (AND, OR, NOT)
- Proceso Triple:
    - (a) Diagrama: Selector de paletas de color por estilo
    - (b) Pseudocódigo: Estructuras condicionales complejas
    - (c) C# Unity: Implementación con Switch-Case
    - Proyecto 1: Asistente de decisiones creativas

Semana 4: Bucles para Patrones y Repetición
- Conceptos: Iteración en arte generativo
- Flowgorithm:
    - Loops While y For
    - Contadores y acumuladores
- Proceso Triple:
    - (a) Diagrama: Generador de secuencia Fibonacci visual
    - (b) Pseudocódigo: Estructuras de repetición
    - (c) C# Unity: Corrutinas y Update() para animación
- Ejercicio: Patrón geométrico progresivo en consola Unity

Módulo 2: Transición a Unity con C# (Semanas 5-8)

Semana 5: Fundamentos de Unity y Primer Script
- Unity Básico:
    - Interface: Scene, Hierarchy, Inspector
    - GameObjects y Components
- Proceso Triple Mejorado:
    - (a) Diagrama: Movimiento básico de un objeto
    - (b) Pseudocódigo: Update() y transformaciones
    - (c) C# Unity: Script de movimiento con Transform.Translate
- Práctica: Cubo que se mueve automáticamente en escena

Semana 6: Variables y Animación en Unity
- C# en Unity:**
    - Variables públicas vs privadas
    - Tipos Vector3 y Color
    - Frame rate y Time.deltaTime
- Proceso Triple:**
    - (a) Diagrama: Animación de rotación y cambio de color
    - (b) Pseudocódigo: Lógica de actualización continua
    - (c) C# Unity: Implementación con Update()
- Ejercicio: Esfera que cambia color gradualmente mientras rota

Semana 7: Interactividad con Input
- Sistema de Input de Unity:
    - Input.GetKey, Input.GetMouseButton
    - Input.GetAxis para movimiento suave
- Proceso Triple:
    - (a) Diagrama: Control de objeto con teclado
    - (b) Pseudocódigo: Detección de input y respuesta
    - (c) C# Unity: Script de control de jugador
- Proyecto 2: Galería interactiva simple con esferas coloreadas

Semana 8: Detección de Colisiones y Triggers
- Física en Unity:
    - Colliders y Rigidbody
    - OnCollisionEnter vs OnTriggerEnter
- Proceso Triple:
    - (a) Diagrama: Sistema de recolección de objetos
    - (b) Pseudocódigo: Lógica de detección y respuesta
    - (c) C# Unity: Sistema de puntuación y destrucción
- Práctica: Juego simple de recolectar cubos con contador

Módulo 3: Arte Generativo en Unity (Semanas 9-12)

Semana 9: Instanciación y Arrays
- Creación procedural:
    - GameObject.Instantiate()
    - Arrays y Listas de GameObjects
- Proceso Triple:
    - (a) Diagrama: Generador de grid de objetos
    - (b) Pseudocódigo: Bucles anidados para creación
    - (c) C# Unity: Instanciación masiva con patrones
- Ejercicio: Crear un campo de cubos con variaciones de color

Semana 10: Programación Orientada a Objetos para Sistemas Visuales
- POO en Unity:
    - Clases y objetos
    - Herencia para comportamientos
- Proceso Triple:
    - (a) Diagrama: Sistema de partículas básico
    - (b) Pseudocódigo: Clase Partícula con propiedades
    - (c) C# Unity: Implementación con clases y listas
- Proyecto 3: Sistema de partículas personalizado

Semana 11: Manipulación de Materiales y Shaders
- Componentes visuales:
    - Renderer y Material
    - Cambio dinámico de propiedades
- Proceso Triple:
    - (a) Diagrama: Animador de propiedades de material
    - (b) Pseudocódigo: Interpolación de valores
    - (c) C# Unity: Lerp para transiciones suaves
- Práctica: Objeto que cicla entre colores y emisión

Semana 12: Efectos de Post-procesamiento y Cámara
- Componentes de cámara:
    - Post-processing stack
    - Animación de propiedades de cámara
- Proceso Triple:
    - (a) Diagrama: Sistema de transiciones de cámara
    - (b) Pseudocódigo: Gestión de estados visuales
    - (c) C# Unity: Script de control de efectos
- Ejercicio: Galería con diferentes filtros de post-procesamiento

---

Módulo 4: Proyecto Final Integrador (Semanas 13-16)

Semana 13: Planificación del Proyecto Final
- Metodología:
    - (a) Diagrama de flujo completo del sistema
    - (b) Pseudocódigo modular por componentes
    - (c) Estructura de scripts en Unity
- Temas sugeridos:
    - Instalación interactiva
    - Animación procedural compleja
    - Herramienta de creación artística
- Entrega: Documentación completa con diagramas

Semana 14: Desarrollo - Módulo Principal
- Implementación:
    - Creación de escena base en Unity
    - Desarrollo del script principal
    - Integración de sistemas core
- Sesiones de tutoría: Resolución de problemas específicos

Semana 15: Desarrollo - Subsistemas y Pulido
- Refinamiento:
    - Implementación de subsistemas secundarios
    - Optimización y debugging
    - Ajustes estéticos y de UX
- Pruebas: Testeo integral y ajustes finales

Semana 16: Presentaciones y Evaluación
- Exhibición:
    - Demostración en vivo de proyectos
    - Explicación del proceso (diagramas → código)
    - Critique técnica y artística
- Evaluación final: Rúbrica integral del proceso completo

Recursos y Herramientas Específicos

Flowgorithm Templates:
- Diagramas base para algoritmos comunes en arte
- Plantillas para sistemas de partículas
- Flujos para gestión de estado visual

C# Unity Script Templates:
- Clase base para objetos visuales
- Manager para sistemas generativos
- Controlador de efectos y transiciones

Flujo de Trabajo Estándar por Ejercicio:
1. Análisis: Definir problema creativo
2. Diagramación: Flowgorithm para lógica
3. Pseudocódigo: Refinamiento conceptual
4. Implementación: C# en Unity
5. Pruebas: Verificación y ajustes
6. Documentación: Comentarios y memoria

Criterios de Evaluación:
- Claridad algorítmica (25%): Diagramas Flowgorithm
- Estructura lógica (25%): Calidad del pseudocódigo
- Implementación técnica (30%): Código C# funcional
- Resultado estético (20%):** Calidad visual final

Este programa asegura que los estudiantes comprendan profundamente la relación entre pensamiento algorítmico y creación visual, con una progresión natural desde conceptos abstractos hasta implementación concreta en herramientas profesionales.