

INSTRUCTOR



Estaba pensando flowgorithm para los primeros diagramas de flujo



Flowgorithm es, a día de hoy, la herramienta “low-floor, high-ceiling” que más rápido pasa del pseudocódigo gráfico a líneas de C# reales sin instalar VS ni escribir ; .
Para un curso “arte → VS → C#” encaja perfectamente: en 2-3 clases el alumno ya ve su diagrama convertido a código limpio y entiende variables, if, while y funciones antes de tocar Unity.

Ventajas concretas que he visto en aulas de animación

1. Exportación 1-Click:

File ➤ Export ➤ C# → código listo para pegar en Unity (clase MonoBehaviour incluida).

2. Colores “artísticos”: los bloques son grandes y se pueden cambiar de color; ayuda a que los alumnos visuales recuerden la lógica por asociación cromática.

3. Depurador visual: ejecuta paso a paso y muestra valores en tiempo real → mismo mindset que los “watch pins” de Blueprint.

4. No requiere instalación de frameworks: portable, < 15 MB.

5. Multilenguaje: si más adelante quieres mostrar diferencia sintáctica (C# vs Python vs Java) lo haces en 5 s.

Secuencia mínima que ya he testado (3 h teóricas + 2 h práctica)

Paso	Objetivo	Flowgorithm	Link con Unity/VS
1	Variables y entrada	Leer alturaPersona je	Mismo float en VS On Start
2	Decisión	if altura < 0 → “Game Over”	Nodo Branch en BP/VS
3	Bucle	while contador < 5 spawn moneda	For Loop (VS) / ForEach Loop (BP)
4	Función	spawnMoneda()	Custom Event en BP / SuperUnit en VS
5	Export C#	File ► Export ► C#	Copiar → crear script Monedas.cs en Unity → ya compila

Trucos para no atascar a los artistas

- Desactiva la opción “Mostrar tipos de datos” la primera clase → evita `double` vs `float`.
- Plantilla pre-hecha: un `.fprg` con las variables `playerX`, `playerY`, `isAlive` → sólo tienen que arrastrar bloques.
- Color-code fijo:
 - Entrada/Salida → azul
 - Proceso → verde
 - Decisión → naranja
(mismo esquema que VS y BP; refuerza transferencia).
- Límite de bloques: máx. 15 por diagrama la primera semana → reduce ansiedad.

De Flowgorithm a Visual Scripting sin grietas

1. Misma nomenclatura:

- Flowgorithm: `alturaPersonaje`
- VS: `Altitude (Float)`
- BP: `Altitude (float)`
→ los nombres se mantienen; copiar-pegar literal.

2. Conversión de tipos: Flowgorithm usa `Real` → enseñar a cambiar a `Float` en Unity (único punto que hay que explicar).

3. Mostrar el código autogenerated junto al diagrama: los alumnos ven que NO hay magia y pierden miedo a escribir `if` más tarde.

Tarea final (1 semana)

Entregable:

- a) Diagrama `.fprg` que haga spawn de 5 cubos si `puntuación > 10`.
- b) C# exportado pegado en Unity → debe compilar y ejecutarse.
- c) Video 15 s del resultado en Game View.

Rúbrica rápida

- Flujo correcto (40 %)
 - Uso de función/procedimiento (20 %)
 - C# compila sin errores (20 %)
 - Video claro (20 %)
-

INSTRUCTOR



Kimi se adorna un poco y no es así de fácil como en sus conclusiones, pero en resumen ese es el camino.

Conclusión

Flowgorithm acorta la curva de aprendizaje de pensamiento algorítmico y, al exportar C# limpio, actúa como puente natural hacia Visual Scripting (VS) y Blueprint.

En 4-5 horas de clase el alumno de arte ya tiene:

- Un diagrama que entiende,
- Un código que compila,
- Y la confianza de que "programar" no es escribir en blanco sobre negro, sino **conectar ideas que ya tiene dibujadas.**