

Introducción

El avance de las tecnologías digitales y su impacto en el desarrollo de la vida personal también ha alcanzado a los profesionales dedicados al arte. Los diseñadores, artistas e ilustradores utilizan cada vez más herramientas sobre los cuales plasmar su creatividad.

Dado que el profesional que diseña videojuegos debe integrar la programación con el arte es una buena idea aprender programación usando herramientas creadas por diseñadores.

Esto resulta interesante, por cuanto se puede aprender la base de los lenguajes de programación dibujando objetos y la interacción entre ellos visualizándolos en un lienzo.

En esta sección brindaremos los conceptos asociados a la programación creativa y nos centraremos en su desarrollo aplicado a la herramienta Processing.

La programación creativa

La programación creativa es un movimiento que establece que los lenguajes de programación se conciben como medios creativos, a la altura de la música, la danza o la pintura. Muchas universidades destacadas a nivel internacional como el MIT adoptaron esta concepción desde hace muchos años, y como resultado crearon entornos de programación que reflejan esta filosofía. No es de extrañar entonces que justamente el MIT, sea la creadora del entorno Processing, el cual usaremos en la asignatura. En este enfoque los lenguajes de programación son considerados un nuevo estándar internacional para crear aplicaciones en cualquier parte del mundo y para cualquier parte del mundo. Esto se torna mucho más visible en el mundo del desarrollo de videojuegos, donde se mezcla el arte y la programación para crear un producto de expresión de sensaciones. Los videojuegos son un medio para transmitir sensaciones por medio de la inmersión (la experiencia cierta de poder expresarse dentro de un mundo ficticio).

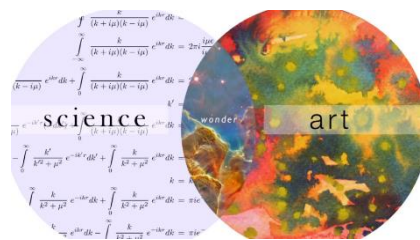
Por este motivo, desde el año 2011 el Ministerio de Cultura de la Nación de la República Argentina, a través de la Dirección Nacional de Industrias Creativas reconoce a la industria del videojuego como una industria creativa.

Fiel a este concepto, se plantea la necesidad de poder aprender programación de manera creativa.

El centro de la programación creativa es el código creativo, el cual se define como

$$\text{código creativo} = \text{arte} + \text{ciencia}$$

Así, el código de la computadora es un medio creativo natural y debe tratarse de la misma manera en que un pintor desarrolla sus pinturas. Entonces se vuelve más relevante la forma e COMO se resuelve y la búsqueda de resolverlo de la manera más interesante y novedosa; para lo cual usa las bases de las matemáticas, la física, la computación y la teoría y práctica artística.



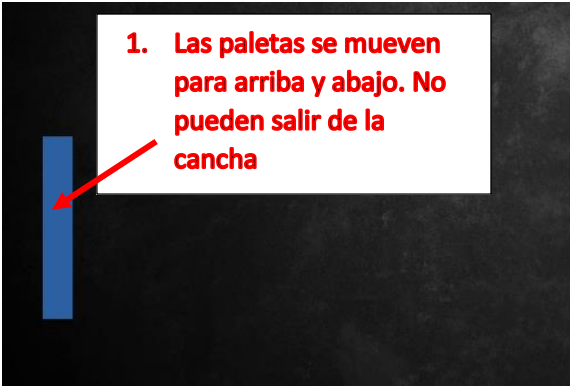
Uno de los objetivos de la programación creativa es combatir las ideas sociales prejuiciosas sobre las ciencias de la computación que sostienen que las mismas son difíciles, aburridas y oscuras; o que solo están dirigidas a un tipo de estudiantes y profesionales específicos.

Para lograrlo, promueve el “aprendizaje haciendo” mediante la construcción de bocetos de código creativo que rigen los principios detrás de la programación en el contexto de la creación y el descubrimiento; donde la teoría se volverá relevante.


Por ejemplo, suponga que deseamos aprender a programar el juego Pong¹, el cual es el primer juego exitoso, creado por Atari en el año 1972 y que está basado en el juego de mesa ping pong. En la programación creativa, se procede de la siguiente manera:

¹ Atari registro los derechos de la palabra. Además, estableció una nomenclatura por la cual los juegos tipo Pong hacen referencia a un género de juegos que consisten en el uso de bate y bola.

1. Primero se aprenderá como pegarle a bola (descubrir las primitivas. En la jerga de los videojuegos se les conoce como mecánicas del juego.
2. Luego se aprenderá la física que funciona detrás del hecho de pegarle a la bola, para saber con qué fuerza y en que dirección debe salir la bola.
3. Aprenderás a codificar una aplicación en frío (esto es entender el entorno de programación)
4. Se aprende luego los principios fundamentales detrás de ellos



1. Las paletas se mueven para arriba y abajo. No pueden salir de la cancha



**2. ¿Ángulo de la pelota?
¿Dirección de la pelota?
¿Velocidad de la pelota?
¿Para dónde sale la pelota luego del rebote?**

LEYES DE NEWTON

```
void setup(){
size(300, 300);
background(255, 0, 0);
noLoop();
}

void draw(){
ellipse(width/2, height/2, 150, 150);
ellipse(50, 50, 25, 25);
}
```

3. Un posible código para dibujar la cancha y la pelota

4. Concepto de clases – variables que intervienen – las operaciones de un programa. Colisiones. Prueba de escritorio. Estructuras de control

Esta manera de desarrollar es más divertida y fácil de aprender y provocará que la parte teórica sea más relevante y fascinante. Por otro lado, las investigaciones sostienen que se desarrolla significativamente el interés personal por la programación.

Qué es Processing

Es un entorno de programación open source² (código abierto) basado en el lenguaje de programación Java para gente creativa. La premisa es que no es necesario que estas personas sean programadores informáticos, por lo cual quienes se inician en la programación lo valoran de manera especial.



² Originalmente hacía referencia al software open source (OSS): código diseñado de manera que sea accesible al público: todos pueden ver, modificar y distribuir el código de la forma que consideren conveniente. El OSS se desarrolla de manera descentralizada y colaborativa, así que depende de la revisión

Su principal fortaleza o ventaja es su facilidad de uso, permitiendo a los desarrolladores crear rápidamente sofisticadas aplicaciones gráficas e interactivas, a la vez que se minimiza la dificultad asociada a la compilación³ y generación de código

Por medio de esta herramienta es posible introducirse en los principios de la programación utilizando formas visuales, movimiento e interacción tal como lo refleja la figura 1.



Figura 1. Objetos visuales en movimiento creados en Processing

El enfoque pedagógico de Processing

Processing fue creado para enseñar los fundamentos de la programación computacional dentro de un contexto visual para servir como software de bocetos y ser usado como herramientas de producción. El modelo de bocetos de la herramienta es totalmente compatible con el modelo de conceptos que se utiliza para diseñar videojuegos; por lo cual resulta apropiado para la enseñanza de los principios básicos de la programación. Y es que resulta que el documento de conceptos es la primera referencia del juego para el equipo de desarrollo. En él detallamos la visión general del juego y nos permitirá tener una idea clara de los objetivos que queremos alcanzar. Una de las partes de este documento hace referencia al “Apartado Gráfico”: cómo visualizamos nuestro juego, el tono, la atmósfera, etc. En este punto contar con una herramienta que permita crear un prototipo de este concepto resulta ideal. Ahí es donde entra Processing. Mediante un conjunto de primitivas, se pueden dibujar diferentes figuras de manera muy sencilla e integrarlas, tal como se observa de manera esquemática en la figura 2.

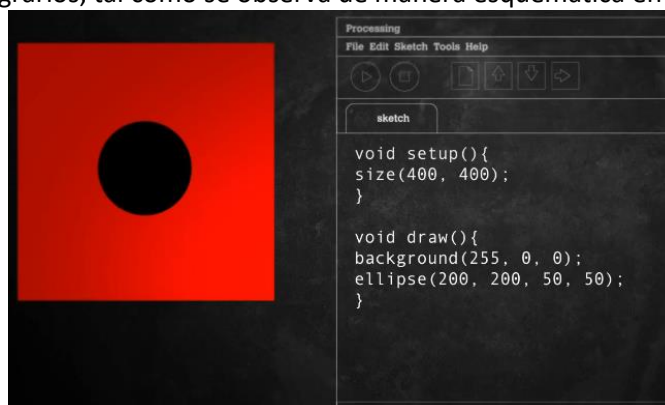
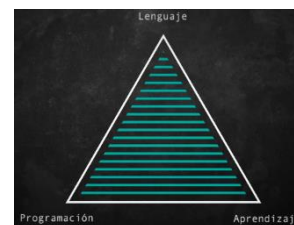
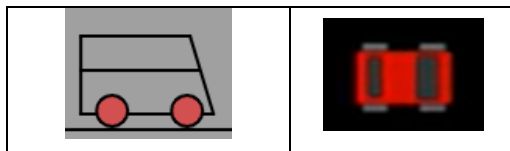


Figura 2. A la derecha las sentencias en Processing para crear un lienzo con un círculo negro en su centro.




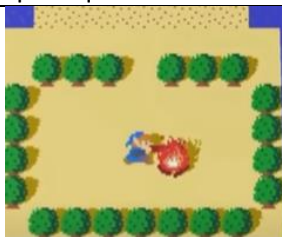




entre compañeros y la producción de la comunidad. Además, suele ser más económico, flexible y duradero que sus alternativas propietarias, ya que las encargadas de su desarrollo son las comunidades y no un solo autor o una sola empresa. El open source se convirtió en un movimiento y una forma de trabajo que trasciende la producción del software con el objetivo de hallar nuevas maneras de solucionar problemas en las comunidades y los sectores.

³ Proceso por el cual el código programado en un lenguaje de programación se traduce al lenguaje que interpretan los ordenadores.

Por ejemplo, suponga que debe crear un juego de autos en 3D, donde quiere mostrar el concepto con un prototipo. Ud podría modelar de diferentes maneras un auto rudimentario al integrar las diversas primitivas disponibles de Processing.



Esto es muy usual e importante. Por ejemplo, hace un par de años Nintendo reveló que para probar el nuevo concepto de mundo abierto para la saga Legend of the Zelda, realizó un prototipo en 2D del mismo, ya que armar el mismo en 3D seguramente hubiera sido bastante costoso en tiempo y recursos humanos, tal como se puede observar a continuación:

Prototipo 2D	Juego Real
	
<p>Prototipo de un concepto mediante programación creativa. The Legend of Zelda: Breath of the wild. A la izquierda el prototipo en 2D. Link dispara una flecha, que pasa por una fogata y se prende, esta flecha llega a los árboles y los quema. A la derecha: el concepto llevado al juego final en 3D (momento en el que Link está por disparar la fecha. En el fondo se ven las cajas que se desean quemar)</p>	
<p>Otra mecánica en prototipo 2D: Link encuentra una rama, la enciende en una fogata y la usa para quemar arbustos</p>	
	
	
	

Las capacidades de Processing no se quedan únicamente en el gráfico de primitivas en 2D, también brinda capacidades para trabajo en 3D, así como la gestión de recursos multimediales que incluyen sonido, vídeo e imágenes. Además, integra diferentes tipos de herramientas para extender las capacidades del entorno tales como:

- Color Selector: Una interface sencilla para seleccionar colores RGB, HSB y valores hexadecimales.

- Movie Maker: Crea películas Quick Time a partir de una secuencia de imágenes. Se puede configurar el tamaño de la imagen, la frecuencia de frames, la compresión de archivos y también incluir un archivo de sonido.
- Archive Sketch: Permite realizar una copia del Sketch (boceto) en la forma de un archivo comprimido con extensión .zip.
- Create Font: Convierte las fuentes dentro de un formato de fuentes interno en Processing para agregarlo a los bocetos.

Otras herramientas realizadas por terceros como contribuciones se deben descargar y agregarlas como funciones adicionales.

Por otro lado, cuenta con bibliotecas específicas para brindar una funcionalidad en particular tales como exportar a archivos PDF y otros. Entre una de ellas realizadas por terceros se halla la biblioteca Animación de Sprites de Peter Larger para realizar el control y animación para juegos y otras aplicaciones gráficas. También existe la biblioteca Green de Zacchary Dempsey-Plante que facilita la creación de juegos 2D dentro de Processing. Además, brinda varias bibliotecas para la gestión del sonido.

Finalmente, si bien no serán objeto de estudio en la materia, existen bibliotecas de Física que podría usarse para simular el motor de física de un juego.

En todos los casos se recomienda realizar una investigación personal para conocer y aplicar las bondades que ofrece la herramienta.