

Interfaz Natural de Programación Unificadora y Tangible usando el lenguaje Simple

Unified and Tangible Natural Programming Interface using the Simple language

Resumen. Profundizaremos sobre el proceso de desarrollo de la instalación interactiva Simple Input System, usando el nuevo lenguaje de programación en español llamado Simpl3. Con las pruebas aplicadas a la fecha se ha logrado evidenciar que la tecnología desarrollada para hacer funcionar esta instalación interactiva puede llegar a convertirse en una Interfaz Natural de Programación (INP) que se configure como herramienta de programación visual interactiva y que facilite la tarea de escribir y leer código básico de programación. De tal manera, usuarios no expertos podrían crear y compartir códigos de programación para producir gráficos y contenidos hipermediales con computadoras, sin tener que saber de memoria su sintaxis. Adicionalmente, Simple Input se convierte en la perfecta excusa para explorar este lenguaje de programación en español creado para artistas y diseñadores, con el que pueden crear código creativo en el mismo idioma en el que piensan, lo que elimina la necesidad del doble proceso cognitivo que implica traducir todo el tiempo lo que piensan al inglés, como sucede con la mayoría de lenguajes de programación para arte generativo.

Palabras clave: Simpl3, programación creativa, Interfaz de Programación Natural, Low-Code No-Code, Software libre, diseño de interacción, diseño de experiencia

Abstract. We will delve into the development process of the interactive installation called Simple Input System, using the new Spanish programming language called Simpl3. Through the tests conducted to date, it has been demonstrated that the technology developed to operate this interactive installation has the potential to become a Natural Programming Interface (NPI) that functions as an interactive visual programming tool, facilitating the task of writing and reading basic programming code. Consequently, non-expert users could create and share programming code to generate graphics and hypermedia content on computers without needing to memorize its syntax. Additionally, Simple Input becomes the perfect opportunity to explore this Spanish programming language designed for artists and designers, enabling them to create creative code in the same language they think in. This eliminates the need for the cognitive process of constantly translating their thoughts into English, as is often the case with most programming languages for generative art.

Keywords: Simpl3, creative programming, Natural Programming Interface, Low-Code No-Code, open-source software, interaction design, experience design

Fecha de recepción: 18/03/2023

Fecha de aceptación: 30/05/2023

Cómo citar: Cuartas, J. (2023).

Interfaz Natural de Programación Unificadora y Tangible usando el lenguaje Simple..

RChD: creación y pensamiento, 8(14), 69-77

<https://doi.org/10.5354/0719-837X.2023.70061>

Revista Chilena de Diseño,

rchd: creación y pensamiento

Universidad de Chile

2023, 8(14).

<http://rchd.uchile.cl>

Introducción

El nombre de esta instalación interactiva se compone de dos palabras: Simple Input. La primera palabra hace referencia al nombre del lenguaje de programación usado para codificar las figuras creadas por los usuarios (Simple3) —la cual fue presentada en el marco del XXI Festival Internacional de la Imagen en la ciudad de Manizales, Colombia— y la segunda palabra, Input, es una sigla que significa Interfaz Natural de Programación Unificadora y Tangible.

La sigla Input se inspira en el concepto de una Interfaz de Programación Natural acuñado en el trabajo doctoral del autor de esta instalación, titulado Programar el mundo en el contexto de las tecnologías libres y las culturas Hacker-Maker. En dicho documento se plantea la necesidad de comenzar a utilizar las Interfaces Naturales de Usuario (NUI), como por ejemplo el sensor Microsoft Kinect, el cual funciona a modo de sistema de entrada para realizar tareas de programación (Cuartas, 2017, p. 285). Es un concepto nuevo que por el momento no ha sido descrito o definido con mayor detalle, siendo esta instalación una posibilidad para ofrecer insumos relevantes y así comenzar a elaborar una construcción teórica y conceptual sobre la necesidad de «Interfaces de Programación Natural», partiendo de los resultados obtenidos con esta instalación como un primer caso de estudio.

Dicha propuesta surge motivada por algunas de las reflexiones planteadas por Bret Victor sobre la importancia y la necesidad de crear interfaces de manipulación directa (Victor, 2013a) y la necesidad de desarrollar medios para pensar lo impensable (Victor, 2013b).

El proyecto de instalación también concuerda con la idea que propone el profesor Hiroshi Ishii, cuando manifiesta que «las interfaces gráficas no alcanzan a abarcar la riqueza de los sentidos humanos y las habilidades que las personas han desarrollado a lo largo de toda una vida de interacción con el mundo físico» (Ishii y Ullmer, 1997, p. 7).

De igual modo, cuando en su momento John Maeda crea el proyecto Design by Numbers (del cual se deriva el proyecto de Processing), con Simple Input se busca hacer «un intento de crear un lenguaje de programación que pueda enseñar los conceptos básicos de la computación a las personas sin inclinaciones matemáticas» (Maeda, 1999, p. 16).

Con esta instalación interactiva se busca hacer una demostración de una Interfaz Natural de Programación experimental con la que se puede generar códigos de programación simplemente moviendo el cuerpo y dando ciertas órdenes de voz. Tal instalación interactiva es el inicio de la segunda fase en el desarrollo del lenguaje de meta programación, llamado «Simple». Este es un lenguaje desarrollado para no programadores que permite escribir códigos de programación usando instrucciones en español y no en inglés. Está licenciado como software libre y viene siendo desarrollado desde el año 2021 al interior del laboratorio Inmedia, en la Universidad del Valle, Cali, Colombia.

En la instalación interactiva Simple Input se busca aplicar los conceptos de manipulación directa e interfaces tangibles, de manera que las personas sin ningún conocimiento de programación puedan crear códigos para generar gráficas computacionales. El objetivo es que se pueda ver al tiempo tanto las formas visuales como el código de computadora correspondiente a cada forma.

Se puede considerar la propuesta de Simple Input como una experiencia de programación en vivo (live coding) para no programadores, que ofrece una experiencia inmersiva de arte tipo Fluxus, muy en la línea de lo que en los años 70 estuvo elaborando Phil Morton y Daniel Sandin en su proyecto General Motos (Morton y Sandin, 1976, pp. 15-20), usando el lenguaje GRASS (DeFanti, 1973) creado por Tomas DeFanti en el año 1973. Esta instalación soporta comandos de voz en los idiomas: español, inglés, portugués y francés.

Metodología y desarrollo

La propuesta de instalación interactiva surge de un proceso empírico analítico, basado en un prototipado incremental, a raíz de experimentos con funcionalidades básicas por separado y que se van integrando paulatinamente, hasta lograr ofrecer al usuario una experiencia completa.

Esta instalación no emerge de una hipótesis que se quisiera comprobar, sino que surge de un interés por explorar usos alternativos a las tecnologías existentes para otorgar al usuario una experiencia de programación creativa en la que no requiera teclear las instrucciones, como ocurre en la mayoría de los lenguajes de programación tradicionales.

Para dicha instalación primero se experimentó por separado con el sistema de captura de movimiento. Las pruebas iniciales se hicieron usando un sistema de captura de movimiento vía webcam basado en una librería de aprendizaje profundo llamado Mediapipe, aunque la latencia y el mayor consumo de recursos hizo descartar este acercamiento. Así, se optó por usar un sensor Kinect 360, con el cual ya se contaba con experiencia previa en otros proyectos, y además garantizaba un desempeño notoriamente más estable; sobre todo, porque para hacer el montaje contábamos apenas con máquinas viejas que no eran muy potentes, factor que también influyó para no elegir el sensor Kinect One, puesto que la máquina que teníamos a disposición no tenía puerto USB 3, el cual es requerido para usar este sensor, mientras que el sensor Kinect 360 funciona con puertos USB 2.0, y por eso resultó perfecto para adaptarse a nuestras limitaciones.

Una vez que funcionó el sistema de captura de movimiento para dibujar figuras y generar el código de las mismas dentro de la proyección, se pasó a trabajar por separado en el sistema de captura de comandos de voz, que simplemente mostraba en la consola de comando el texto de lo que se decía. La librería utilizada fue SpeechRecognition, la cual permite realizar consultas al sistema de reconocimiento de voz de Google, lo cual implica

que es necesario contar con conexión a internet para enviar el archivo de audio a Google y luego recibir de vuelta la cadena de caracteres que Google reconoce en las palabras dichas por el usuario. Una vez que el sistema funcionó de la forma deseada, se pasó a unir el sistema de captura de movimiento con el sistema de captura de comando de voz.

La última etapa fue experimentar con un sintetizador de voz que pudiera pronunciar las instrucciones de las líneas de código en diferentes idiomas, para lo cual se prefirió la librería eSpeak NG. Una vez logrado un uso de forma estable, se agregó al sistema que ya integraba la captura de movimiento y de comandos de voz.

Para esta instalación interactiva utilizamos la siguiente configuración de hardware y software:

Requerimientos de hardware

- Computador con procesador i3, 4 gigas de memoria RAM.
- Sensor Microsoft Kinect versión 360 con
- Convertidor a puerto USB y alimentación externa, para Kinect 360.
- Televisor de 50 pulgadas.

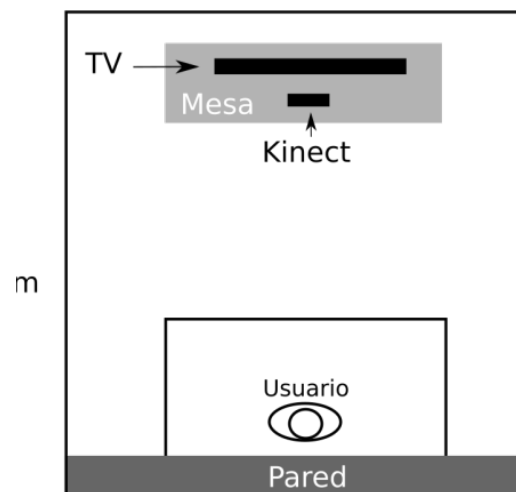
Requerimientos de software

- Ms Windows 10
- Software KinectSDK-v1.8-Setup
- Software Kinect for Windows Developer Toolkit
- Python versión 2.7
- Librería Pykinect para Python 2.7
- Librería SpeechRecognition para Python 2.7
- Librería eSpeak-ng para Python 2.7.

Requerimientos del espacio

Un lugar de tres metros cuadrados con una pared para que el usuario se ubique en frente (Figura 1).

Figura 1. Plano del montaje (vista superior).
Fuente: elaboración propia.



La instalación Simple Input usa un sensor Kinect 360 para reconocer la posición de las manos del usuario. Con estos datos se controla la interacción con la interfaz de la instalación. Adicionalmente la instalación incorpora un sistema de reconocimiento de voz, que se activa cuando el usuario toca el botón gris de la interface.

Para programar esta instalación se usó un computador con Windows 10 y Python 2.7 instalado. Para la captura de movimiento es necesario instalar primero los drivers del sensor Kinect Software KinectSDK-v1.8-Setup y Software Kinect for Windows Developer Toolkit. Ambos se descargan directamente de la página oficial de Microsoft.

Una vez instalados, se debe proceder a instalar Python versión 2.7 para luego, con la herramienta «pip», instalar las librerías Pykinect y SpeechRecognition. Gracias a Pykinect es posible acceder al sensor Kinect desde Python, y con SpeechRecognition se pueden hacer consultas a la API de Google para realizar el proceso de reconocimiento de voz. Ello significa que, para que la instalación funcione correctamente, se requiere contar con una buena conexión a internet, ya que cada comando de voz que utilice el usuario debe ser enviado vía web a los servidores de Google y así como respuesta se reciba una cadena de texto con las palabras reconocidas.

Usando Pykinect se puede detectar la posición en los ejes (x, y, z) de cada articulación principal del esqueleto humano. De estos datos se extrae la posición de cada una de las muñecas del usuario. Los datos de la mano izquierda se usan para determinar en qué parte de la pantalla se debe mostrar la figura geométrica y según el valor en el eje z se puede calcular que tan grande o pequeña se debe mostrar. De esta forma el usuario controla el tamaño de la figura.

Con la mano derecha se tiene control de dos botones: uno de ellos es color verde y sirve para fijar la figura que tiene en la mano izquierda, y el botón gris se usa para activar el micrófono para escuchar lo que dice el usuario y determinar si pronuncia alguna palabra que sea un comando de Simple Input. En el caso que la cadena que devuelve Google corresponda a alguno de los comandos, entonces se ejecuta las instrucciones de dicho comando, ya sea cambiando la forma, el color o ejecutando la animación (Figura 2).

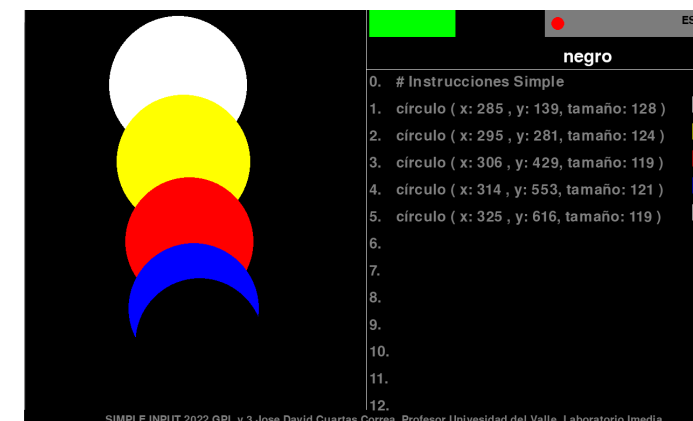


Figura 2. Imagen de animación Simple Input.
Fuente: elaboración propia

Resultados

Cuando el usuario ingresa a la instalación, debe ubicarse cerca de la pared que está en frente al sensor Kinect, que se encuentra justo debajo del televisor.

Con la mano derecha mueve las formas y con la mano izquierda activa los comandos de voz y fija las formas. Si acerca o aleja la mano izquierda del sensor, entonces la forma aumenta o disminuye de tamaño. Cuando con la mano derecha se toca el botón verde, se fija la forma en la ubicación donde se encuentre apuntando la mano izquierda.

Para facilitar la interacción, un círculo blanco representa la mano izquierda y un círculo verde pequeño representa la mano derecha. Una vez se elija otra forma u otro color entonces la mano izquierda controlará la forma que esté activa.

Cada vez que se fija una figura aparece en la mitad derecha de la interface la instrucción que corresponde a esa figura en esa posición y tamaño con el color activo.

Para usar los comandos de voz es preciso tocar con la mano derecha el botón gris que se encuentra en la parte superior derecha, y en el momento que suene la campana se debe decir en voz alta el comando a usar. Para cambiar formas se hace uso de los comandos de voz: cuadrado, círculo, triángulo. Para cambiar el color de las formas se hace uso de los comandos de voz: amarillo, azul, rojo, blanco y negro. Para cambiar el idioma de reconocimiento se hace uso de los comandos de voz: inglés, español, francés y portugués.

Una vez se logra el dibujo terminado se verá que cada figura corresponde a cada línea de código generada automáticamente en la mitad derecha de la pantalla.

Para animar la secuencia en el mismo orden que se generó el dibujo (línea por línea), se debe tocar el botón gris en la parte superior derecha y en el momento que suene la campana, hay que decir en voz alta el comando «anima» (Cuartas, 2022).

Hecho esto entonces la imagen se pone en negro y se va pronunciando automáticamente el nombre de la figura, sobreponiéndose una a una las figuras hasta completar la imagen final. El listado completo de los comandos de voz son los siguientes (Tabla 1):

COMANDO PARA FORMAS	
Comando	Función
Cuadrado	Muestra un cuadrado
Círculo	Muestra un círculo
Triángulo	Muestra un triángulo
COMANDO PARA COLORES	
Comando	Función
Amarillo	Cambia el color de la forma a amarillo
Azul	Cambia el color de la forma a azul
Rojo	Cambia el color de la forma a rojo
Blanco	Cambia el color de la forma a blanco
Negro	Cambia el color de la forma a negro
COMANDO DE CONTROL	
Comando	Función
Borra	Elimina la última forma que se acaba de agregar
Nuevo	Crea un nuevo proyecto y borra las líneas actuales
Anima	Genera una animación de las líneas actuales
Salir	Cierra el software de la instalación Simple Input
Inglés	Cambia el idioma de reconocimiento a inglés
Español	Cambia el idioma de reconocimiento a español
Francés	Cambia el idioma de reconocimiento a francés
Portugués	Cambia el idioma de reconocimiento a portugués

Tabla 1. Listado completo de comandos de voz.

Fuente: elaboración propia.

Discusión y conclusiones

Evidenciamos que con la tecnología desarrollada para hacer funcionar esta instalación interactiva puede llegar a convertirse en una Interfaz Natural de Programación (INP) que se configure como herramienta de programación visual interactiva, que facilita la tarea de escribir y leer códigos básicos de programación. De tal manera, usuarios no expertos podrían crear y compartir códigos de programación para producir gráficos y contenidos hipermediales con computadoras, sin tener que saber de memoria su sintaxis. Se ofrece así una experiencia interactiva que suaviza la inclinada curva de aprendizaje que implican los lenguajes de programación, reduciendo especialmente la posibilidad de cometer los típicos errores de sintaxis al programar con los populares lenguajes basados en texto. Gracias que esta instalación combina dos tecnologías, una para el reconocimiento de voz con el que se identifican los comandos de voz y otra para la captura de movimiento sensor Microsoft Kinect, se le puede

ofrecer al usuario una experiencia interactiva con una Interfaz Natural de Programación, donde con el movimiento de sus manos y con comando de voz puede crear códigos de programación sin ser programador. Ello responde a uno de los objetivos de la agenda investigativa en la que vengo trabajando desde hace más de una década, que es la de poder programar el mundo usando los objeto del mundo (Cuartas, 2017), con proyectos como el software Atomic Authoring Tool (Cuartas, 2009), para que artistas y diseñadores pudieran crear aplicaciones de realidad aumentada en una época en la que la única forma de hacerlo era sabiendo programar en lenguaje C; o el proyecto ShapeCodeWriter del año 2016, que puede considerarse la semilla de esta instalación, en la que se usaba un control de la consola Wii para dibujar rectángulos en el aire y generar su respectivo código en lenguaje Processing; o como mis proyectos Meta_Processing y Meta_Javascript del año 2020, con los cuales comencé a explorar las posibilidades para crear un lenguaje de programación creativa que permitiera usar instrucciones en español.

Los aprendizajes logrados con la construcción de esta instalación podrían adaptarse para facilitar la accesibilidad a la programación a personas con necesidades especiales. Adicionalmente, podría usarse como interfaz de programación en tiempo real dentro de experiencias inmersivas de realidad virtual, evitando la necesidad de levantar el visor de RV para escribir en una pantalla tradicional el nuevo código y probarlo nuevamente en el mundo virtual.

Finalmente, identificamos que el principal propósito de esta instalación fue la de hacer una invitación o provocación a mal utilizar la tecnología, ya que como decía el artista finlandés Erkki Kurenniemi, mantendremos el control sobre las tecnologías mientras podamos mal utilizarlas (Erkki Kurenniemi en Taanila, 2003).

Agradecimientos

Agradezco a la Universidad del Valle, Colombia, y al laboratorio INMEDIA de la Facultad de Ciencias Integradas, donde se desarrolló el prototipado de la instalación interactiva. Además, al Festival Internacional de la Imagen, por la oportunidad ofrecida para presentar esta instalación a pesar de las dificultades técnicas que implicaba su montaje.

Referencias

- Cuartas, J. D. (2009). *ATOMIC Authoring Tool*. SourceForge. Recuperado el 19 de junio de 2023 de <https://sourceforge.net/projects/atomic-project/>
- Cuartas, J. D. (2017). *Programar el mundo en el contexto de las tecnologías libres y las culturas Hacker-Maker*. Caso de estudio: Hitec Lab. [Tesis doctoral]. Universidad de Caldas.
- Cuartas, J. D. (2022). *Instalación Interactiva: Simple Input*. Recuperado el 19 de junio de 2023 de <https://www.youtube.com/watch?v=C14wrl-QaMo>
- DeFanti, T. (1973). *The Graphics Symbiosis System (GRASS)*. Recuperado el 19 de junio de 2023 de <https://www.evl.uic.edu/dan/GRASS.html>
- Ishii, H. y Ullmer, B. (1997). *Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms*. En CHI '17: Proceedings of ACM SIGCHI Conference on Human Factors in Computing Systems (pp. 234-241). Association for Computing Machinery.
- Maeda, J. (1999). *Design by Numbers*. MIT Press.
- Morton, P. y Sandin, D. J. (1976). *General Motors (Part 2) - Phil Morton (1976)* [video]. Vimeo. Recuperado el 19 de junio de 2023 de <https://vimeo.com/59157626>
- Victor, B. (2013a). Bret Victor - *Stop Drawing Dead Fish* [video]. Recuperado de <https://www.youtube.com/watch?v=ZfythvgHybA>
- Victor, B. (2013b). Bret Victor - *Media for Thinking the Unthinkable* [video]. Vimeo. Recuperado el 19 de junio de 2023 de <https://vimeo.com/67076984>
- Taanila, M. (2003). *The Dawn of Dimi* [documental]. Kiasma y Kinotar.